

Analysis of Convergence Properties of a Stochastic Evolution Algorithm

Chi-Yu Mao and Yu Hen Hu

Abstract—In this paper, the convergence properties of a stochastic optimization algorithm called the stochastic evolution (SE) algorithm is analyzed. We show that a generic formulation of the SE algorithm can be modeled by an ergodic Markov chain. As such, the global convergence of the SE algorithm is established as the state transition from any initial state to the globally optimal states. We propose a new criterion called the mean first visit time (MFVT) to characterize the convergence rate of the SE algorithm. With MFVT, we are able to show analytically that on average, the SE algorithm converges faster than the random search method to the globally optimal states. This result is further confirmed using the Monte Carlo simulation.

I. INTRODUCTION

Stochastic evolution (SE) is a stochastic combinatorial optimization method which pursues optimization via a two-phase *combinatorial evolution process*. In the generalization phase, the solution of the present iteration is partially dismantled so that it is *generalized* to a skeleton solution. In the specialization phase, a new specific solution is generated from the skeleton solution.

The SE algorithm has found various applications in the area of computer-aided design tools for VLSI physical design [1]–[12]. For example, in applying the SE method to the partitioning problems, Saab and Rao [1], [2] suggested several implementation strategies to deal with a number of constraints. In [3], Saab used SE to solve problems such as Network Bisection, Vertex Cover, Set partition, Hamiltonian circuit, Traveling Salesman, and so on. In a package called ESP, the SE method has been applied to the standard cell placement problem [4], [5]. Lin *et al.* [6] proposed a SE router SILK to solve switchbox routing problems. Ly and Mowchenko [7] applied SE to high level synthesis. Mao and Hu [8] proposed a SE based algorithm for CMOS gate matrix layout generation. SE has also been used in the design of finite-state machine (FSM) [10], system partitioning for high-level synthesis of multichip module (MCM) [11], and reordering of binary decision diagrams (BDD's) [12]. In [3], Saab also described a number of general combinatorial optimization applications.

The SE algorithm distinguishes itself from the *simulated annealing (SA)* algorithm and the *genetic algorithm (GA)* [13] in several aspects. Compared to the SA algorithm, the SE algorithm uses a two-phase search procedure, and has no cooling schedule.¹ Compared to the GA algorithm, the SE algorithm keeps only one solution in each iteration (generation) and has a different (two phases) evolution mechanism. The GA algorithm usually keeps a pool of potential solutions (*genes*) concurrently, and uses mating and mutation to generate new solutions.

Several early works in the SE algorithm reported [1]–[6] that empirically, the SE algorithm converges faster than the SA algorithm

on specific physical design problems. In [5], it has been shown that a special case of the SE algorithm formulation will converge in probability. It is unclear whether more general SE algorithm formulations will converge, or how fast they will converge. To fill this gap, in this paper, we offer a convergence proof (in probability) of a generic SE algorithm formulation that is potentially applicable to many existing SE based applications. Our proof is based on the argument that the SE algorithm formulation constitutes an ergodic Markov chain. Moreover, we propose to use the *mean first visit time (MFVT)* of the corresponding Markov chain to quantify the time constant of the exponential asymptotic convergence rate of stochastic optimization algorithms. The MFVT proves to be a useful tool to compare the average convergence speed of two stochastic optimization algorithms analytically. To illustrate, we show that, in solving a one-dimensional CMOS gate matrix layout problem, the SE algorithm, on average, converges to the globally optimal solution faster than the random search method. This analysis is further verified with a Monte Carlo simulation that yields remarkably consistent results with those predicted analytically.

The rest of this paper is organized as follows. In Section II, the formulation of a generic SE algorithm is presented. In Section III, it will be shown that this generic SE algorithm corresponds to an ergodic Markov chain, and hence, converges to a global optimal solution in probability. Moreover, we develop the convergence rate analysis using a notion of MFVT, and derive the expression of the time constant which dictates the exponential convergence rate. In Section IV, an illustrative CMOS gate matrix layout example is given. Based on results developed in Section III, we derived the state transition matrices and the corresponding MFVT's of both the generic SE algorithm and the random search algorithm. A Monte Carlo simulation has been carried out that yielded very consistent results compared to those predicted analytically.

II. A GENERIC STOCHASTIC EVOLUTION ALGORITHM FORMULATION

There are several different versions of the SE algorithm formulation. The one presented in this paper is similar to that presented in [8]. However, we purposely keep the formulation generic so that the convergence analysis performed in this paper can be readily applied to other variations of the SE algorithm. To facilitate a fairly detailed description of the generic SE algorithm formulation, let us consider the solution of a linear assignment problem in which distinct elements in a set $E = \{e_1, e_2, \dots, e_{|E|}\}$ are to be assigned to a set of slots $L = \{l_1, l_2, \dots, l_{|L|}\}$.² A state $s = \{s(e_1), s(e_2), \dots, s(e_{|E|})\}$, with $s(e_i) \in L, 1 \leq i \leq |E|$, to be a *complete assignment*³ of all the elements in E to slots in L . Define⁴ a heuristic cost function $f_c(s(e))$ to be the cost incurred when the element e is assigned to a slot $s(e)$. The actual definition of $f_c(s(e))$ depends on the context of the application. In the SE formulation, it is assumed that the overall cost function $f_o(s)$ associated with a state is the average of the individual heuristic assignment cost. That is

$$f_o(s) = \frac{1}{|E|} \sum_{e \in E} f_c(s(e)). \quad (1)$$

² $|E|$ denotes the number of elements in the set E . $|L|$ denotes the number of slots in the set L .

³Depending on whether it is a placement problem, permutation problem, or a partitioning problem, one or more elements may be assigned to one slot.

⁴This cost function is similar to the gain function of a compound move of the SE algorithm formulation described in [1].

Manuscript received May 10, 1994; revised February 17, 1995 and March 19, 1996. This paper was recommended by Associate Editor C.-K. Cheng.

C.-Y. Mao is with Cadence Design Systems, Inc., San Jose, CA 95134 USA.

Y. H. Hu is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA.

Publisher Item Identifier S 0278-0070(96)04780-X.

¹The SE algorithm as described in [1] uses a parameter p which plays a role similar to the role of cooling temperature in the SA algorithm. But this parameter is not updated according to a predefined cooling schedule.

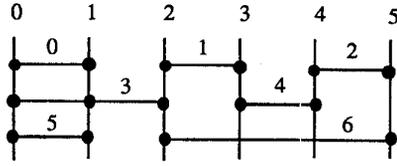


Fig. 1. The circuit "x2."

The goal of the linear assignment problem is to find an optimal state s^* which minimizes $f_c(s)$. For example, in [8], this linear assignment formulation is applied to solve a CMOS gate matrix layout problem [9], [16]. As depicted in Fig. 1, a gate matrix layout consists of vertical and horizontal wires corresponding to the gate of a MOS transistor as well as interconnecting wires, respectively. Dark circles represent contacts between vertical and horizontal wires. The objective is to minimize the maximum number of horizontal tracks needed to place those horizontal wires by searching for the optimal linear ordering of the vertical wires. The heuristic cost function f_c is defined as

$$f_c(s, g_i, l_j) = \frac{1}{|N(g_i)|} \sum_{n_i \in N(g_i)} (1 - \text{Com}(n_i; s)) \quad (2)$$

where $|N(g_i)|$ is the total number of contacts needed on gate g_i according to the schematics of the circuit, and

$$\begin{aligned} \text{Com}(n_i; s) &= \frac{O(n_i)}{\text{Len}(n_i; s)} \\ &= \frac{(\# \text{of contacts to a horizontal wire } n_i) - 1}{\text{the length of the wire } n_i} \end{aligned} \quad (3)$$

measures the *compactness* of a horizontal wire under the present ordering of vertical wires. In other applications, different heuristic cost functions can be defined based on the domain knowledge of the CAD tool designers.

The search steps in the SE algorithm are conducted in two phases: a generalization phase and a specialization phase. This two-phase evolution procedure is illustrated in Fig. 2. In the generalization phase, a portion of the assigned elements will be evicted from their present slots and placed into a queue $Q = \{q_1, q_2, \dots, q_{|Q|}\}$. The total number of evicted elements $|Q|$, sometimes also known as the *window size*, is usually chosen such that $|E|/2 \geq |Q| \geq 2$. In this paper, we will assume $|Q|$ is prespecified and will remain unchanged. Those elements to be placed into the queue are selected according to a *randomized* heuristic cost function defined on each assignment

$$\eta_i = \hat{f}_c(s(e_i), \alpha_i) = f_c(s(e_i)) + \alpha_i, \quad 1 \leq i \leq |E| \quad (4)$$

where $\{\alpha_i; 1 \leq i \leq |E|\}$ is a set of independent, identically distributed (i.i.d.) random variables with zero mean. After the randomized cost for all elements are evaluated, the $|Q|$ elements with largest randomized cost will be evicted from their current slots and placed into the queue in a descending order according to the value of η_i . The randomization procedure described here guarantees that every element in E will have a nonzero probability to be selected into Q .⁵ To be more specific,

Lemma 1: Denote i_* and i^* , respectively, to be the index of elements in E such that

$$f_c(s(e_{i_*})) \leq f_c(s(e_i)) \leq f_c(s(e_{i^*})) \quad 1 \leq i \leq |E|. \quad (5)$$

Let $\{\alpha_i\}_{i=1}^{|E|}$ be a set of uniformly distributed i.i.d. random variables over the interval $[-\frac{\lambda_2 - \lambda_1}{2}, \frac{\lambda_2 - \lambda_1}{2}]$ where λ_1 and λ_2 are such that $\lambda_1 < f_c(s(e_i)) < \lambda_2$. Then

$$\text{Pr}\{\eta_{i_*} > \eta_{i^*}\} > 0. \quad (6)$$

⁵There are alternatives ways to select elements to Q in [1].

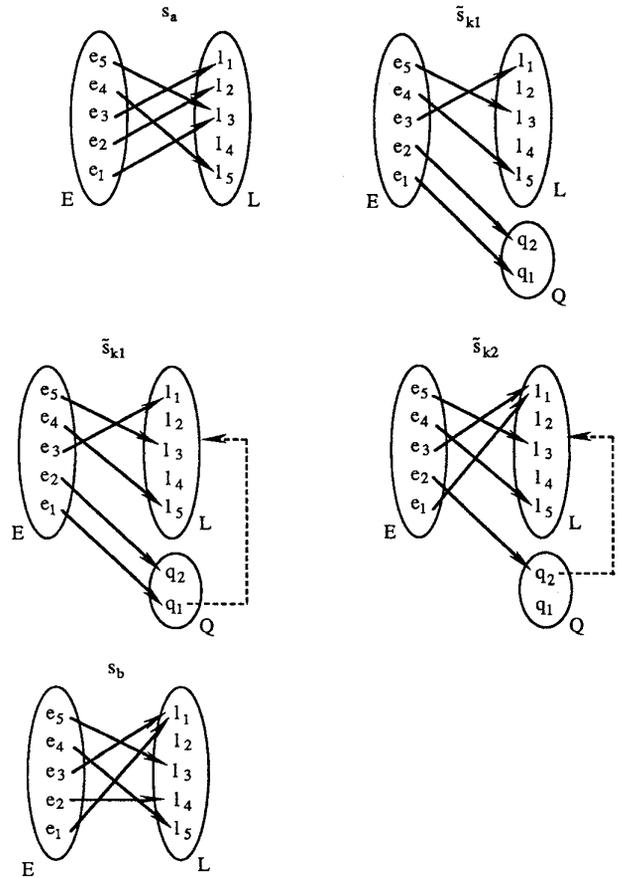


Fig. 2. State transition process.

Proof: Since α_i is uniformly distributed over $[-\frac{\lambda_2 - \lambda_1}{2}, \frac{\lambda_2 - \lambda_1}{2}]$, its mean value $E\{\alpha_i\} = 0$. Moreover, $\eta_i = \alpha_i + f_c(s(e_i))$ will be a uniformly distributed random variable over the interval $[f_c(s(e_i)) - \frac{\lambda_2 - \lambda_1}{2}, f_c(s(e_i)) + \frac{\lambda_2 - \lambda_1}{2}]$ with mean value $E\{\eta_i\} = f_c(s(e_i))$. Since α_{i_*} and α_{i^*} are independent, so are the random variables η_{i_*} and η_{i^*} . Therefore

$$\begin{aligned} \text{Pr}\{\eta_{i_*} > \eta_{i^*}\} &= \int_a^b h\eta_{i_*}(z) \int_a^z h\eta_{i^*}(x) dx dz \\ &= \frac{1}{2} \left(1 - \frac{f_c(s(e_{i^*})) - f_c(s(e_{i_*}))}{\lambda_2 - \lambda_1} \right)^2 > 0 \end{aligned} \quad (7)$$

where $a = f_c(s(e_{i^*})) - \frac{\lambda_2 - \lambda_1}{2}$, $b = f_c(s(e_{i_*})) + \frac{\lambda_2 - \lambda_1}{2}$, and h is probability density function. \square

The randomization of the heuristic cost function is crucial to maintain the stochastic nature of the SE algorithm. Otherwise, this search strategy degenerates to a deterministic heuristic search algorithm. Note that other families of symmetrical probability distributions can also lead to the same conclusion as stated in Lemma 1.

Once the $|Q|$ elements are evicted from their current slots and placed into the queue, the remaining partial assignment, together with the Q unassigned elements in the queue will constitute a skeleton state \bar{s} . This skeleton state is significant to the SE method because a set of *neighboring states* can be reached by assigning elements in Q into emptied slots.

During the specialization phase, elements in Q will be reassigned to empty slots one by one. Priority will be given to elements in Q with larger randomized heuristic cost calculated during the generalization phase. Once the reassignment starts, the solution evolves from the

skeleton state \bar{s} to a sequence of *intermediate states* until the queue is emptied, and a new solution (state) is arrived. To facilitate the reassignment of element e_j in Q , a heuristic cost function $f_c(\bar{s}(e_j), k)$ is defined when element e_j in the queue is assigned to an emptied slot k . This heuristic cost function then is randomized by adding it to a zero-mean, i.i.d. random variable β_k , yielding a randomized cost function

$$\gamma_{jk} = f_c(\bar{s}(e_j), k) + \beta_k. \quad (8)$$

When the element e_j in the queue is to be reassigned, we choose the slot k with the minimum randomized cost. In a manner similar to Lemma 1, we can prove the following lemma.

Lemma 2: Denote k to be the index of emptied slots in L . Let k_* and k^* , respectively, be such that

$$f_c(\bar{s}(e_j), k_*) \leq f_c(\bar{s}(e_j), k) \leq f_c(\bar{s}(e_j), k^*) \quad e_j \in Q \quad (9)$$

Let $\{\beta_i\}_{i=1}^{|E|}$ be a set of uniformly distributed i.i.d. random variables over the interval $[-\frac{\lambda_2 - \lambda_1}{2}, \frac{\lambda_2 - \lambda_1}{2}]$ where λ_1 and λ_2 are such that $\lambda_1 < f_c(\bar{s}(e_j), k) < \lambda_2$. Then

$$\text{Pr.}\{\gamma_{jk_*} > \gamma_{jk^*}\} > 0. \quad (10)$$

The proof of Lemma 2 is very similar to that of Lemma 1 and hence will be omitted here. Lemma 2 states that even the least favorable (emptied) slot k_* still has a nonzero probability to be chosen to place e_j . Hence, the specialization steps are governed by probability.

Once a new solution is generated, its quality will be evaluated based on the objective function. If it is better than the currently best solution, it will become the newly found best solution. This iteration continues until certain termination criteria are satisfied. One often used criterion is the maximum number of iterations. The other is the maximum number of iterations since the last update of the best solution.

III. CONVERGENCE ANALYSIS

In this section, we will establish the global convergence of the generic SE algorithm presented above. For this purpose, we will first show that the SE algorithm corresponds to an ergodic Markov chain. The structure of the Markov chain is determined by the size of Q , and the state transition probability is determined by the randomized cost functions used during generalization and specialization phases.

First, we would like to argue that the transition from a state s_a to a new state s_b is entirely determined by probability $P_{ab} = \sum P_S P_A$ shown in Fig. 3. To be more exact, during the generalization phase, the selection of $|Q|$ elements from s_a into the queue is random. In particular, according to the descriptions in Section II, the probability of reaching a specific skeleton state \bar{s} from s_a , $P_S = \text{Prob.}\{\text{skeleton state} = \bar{s} \mid \text{current state} = s_a\}$ is

$$P_S = \text{Prob.}\left\{\min_{i \in Q} \eta_i > \max_{j \in E \setminus Q} \eta_j\right\}. \quad (11)$$

Similarly, the probability of reassigning the first element e_j in Q to an emptied slot k , denoted by P_{A1} is also dictated entirely by probability. Specifically, we have

$$P_{A1} = \text{Prob.}\{z > \gamma^*\} \quad (12)$$

where $\gamma^* = \min \gamma_{jk}$ is the randomized heuristic assignment cost evaluated on all emptied slots, $z = \min_{i \in Q \setminus k^*} \gamma_i$ is the minimum assignment cost of the elements remaining in Q . Such evaluation of P_S and P_A must be carried out for all possible selections of Q elements from the E elements (i.e., all possible skeleton states). This then will yield the state transition probability P_{ab} , and the

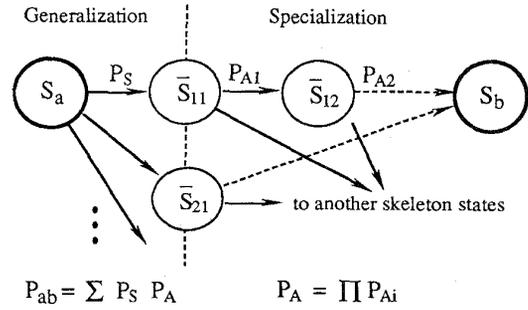


Fig. 3. The calculation of state transition probability P_{ab} .

entire probability transition matrix P . Detailed derivation of these probabilities can be found in [9], [17].

From above argument, we establish that the state transition probability P_{ab} is dependent only on the present state and independent of all prior states. Moreover, it is not a function of time (iteration indices), and the total number of states are finite. Hence by definition [14], we have

Lemma 3: The set of states and the state transition probabilities derived from the generic SE algorithm constitute a time-homogeneous Markov chain.

State s_j of a Markov chain is said to be *reachable* from state s_i if it is possible to make transitions from state s_i to state s_j using finite number of state transitions. If every state is reachable from every other state, this Markov chain is said to be *irreducible*. The following lemma proves that the generic SE algorithm formulation corresponds to an irreducible Markov chain.

Lemma 4: In the generic SE algorithm formulation, any state can be reached from any other states in $\lceil \frac{|E|}{|Q|} \rceil$ state transitions.

Proof: During a single state transition, a state s_a can reach any other states s_b that has at most $|Q|$ different assignment compared to s_a . Since there are at most $|E|$ different assignments between any pair of states, any state can be reached from any other state in $\lceil \frac{|E|}{|Q|} \rceil$ state transitions. \square

The state s_i is said to be *periodic* with period d if $P_{ii}^n > 0$ only for $n = d, 2d, 3d, \dots$, where d is the largest such integer and $d > 1$. If $d = 1$ then s_i is an aperiodic state. Because every state may return to itself in one state transition, i.e., $d = 1$, the Markov chain corresponding to the generic SE algorithm is aperiodic.

A time-homogeneous Markov chain which has finite number of states and is both irreducible and aperiodic, is said to be *ergodic*. The limiting state probability vector π of an ergodic process always exists and is independent of the initial state probability distribution $\pi^{(0)}$. The above discussions lead to the following observation.

Theorem 1: The Markov chain produced by the generic SE algorithm is an ergodic Markov chain.

Because the Markov chain associated with the SE algorithm is ergodic, there will be a nonzero probability that every state, including the globally optimal state will be visited eventually as $n \rightarrow \infty$ [14]. In other words,

Theorem 2: The generic SE algorithm converges to a globally optimal solution asymptotically.

To characterize the asymptotic convergence rate, we need to analyze how the SE algorithm converges to a globally optimal state. However, in many combinatorial problems, it may not be possible to know whether a globally optimal solution has been reached. Instead, a best solution observed so far is recorded and updated. One termination criterion which has been used by the SE algorithm, as well as other stochastic optimization algorithms, is to terminate iterations when the currently best solution remains unchanged for a fixed number of

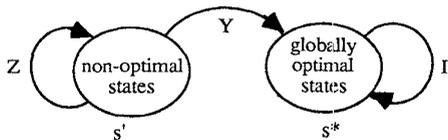


Fig. 4. Markov chain model of the current best solution.

iterations. Therefore, when the currently best solution becomes the globally optimal solution, the SE algorithm will be terminated after a few more iterations. Thus, the event that the currently best solution becomes the globally optimal solution guarantees the termination of the SE algorithm. Moreover, the state transition of the currently best solution can be modeled by an *absorbing Markov chain* in which the globally optimal states form the set of absorbing states. The MFVT of the Markov chain is equal to the absorbing time of the absorbing Markov chain [14]. When the currently best solution hits an absorbing state, it will never return to any nonoptimal states. If we lump all nonoptimal states into a super state, and all globally optimal states into another super state, the state transition diagram can be depicted in Fig. 4. The transition matrix of an absorbing Markov chain can be partitioned into a special structure

$$P = \begin{bmatrix} I & 0 \\ Y & Z \end{bmatrix} \quad (13)$$

where we assume all the absorbing states are reordered to be at the first part of the state vector, Z is the probability state transition matrix corresponding to the state transition among *transient states*, Y is the probability that a transient state will move to an absorbing state. The convergence of the SE algorithm thus can be interpreted as the eventual state transition of the currently best solution from the set of nonoptimal states to the set of absorbing globally optimal states. Suppose the set of transient states has an initial state vector π' , then the probability that at the k th steps, the current best solution will become a globally optimal solution can be found as

$$p(k) = \pi' Z^k \zeta \quad (14)$$

where ζ is a column vector consisting of ones.

Lemma 5:

$$\lim_{k \rightarrow \infty} p(k) = 0. \quad (15)$$

Proof: Since the Markov chain is irreducible, we must have $Y \neq 0$. In addition, Z is a submatrix of the state transition matrix P whose eigenvalues must have their magnitudes bounded by unity. Therefore, there exists a transient state i such that $\sum_j z_{ij} < 1$. Following Corollary 3 in [17, p. 31], we conclude that the Perron–Frobenius radius of the matrix Z , $r(Z) < 1$. Hence, this lemma is proved. \square

Lemma 5 states that the SE algorithm not only will converge asymptotically, but also converges exponentially. Next, we would like to characterize the average number of iterations until convergence using a notion called MFVT. The average number of steps taken for the SE algorithm to converge can be computed as [14], [15]

$$\tau_c = \pi' (I - Z)^{-1} \zeta = \pi' \left[\sum_{k=0}^{\infty} Z^k \right] \zeta = \sum_{k=0}^{\infty} p(k) \quad (16)$$

which is also known as the MFVT of the Markov chain. Usually, the smaller the value of τ_c , the faster an algorithm will converge to a globally optimal solution. The advantage of the SE algorithm over a purely random optimization algorithm is that domain knowledge is encapsulated in the heuristic cost functions to bias the search direction in order to reach a globally optimal solution faster. Unfortunately, it

is impossible to prove that this is always true due to the stochastic nature of the SE algorithm as well as the random search algorithm. Below, we will use an analytical example and the corresponding Monte Carlo simulation to verify these assertions.

IV. AN ILLUSTRATIVE EXAMPLE

We have developed a software program to compare the performance of SE with random search method on a small scale CMOS gate matrix layout problem. The purpose is to illustrate convergence analysis results derived in the previous section. In particular, with this package, we are able to explicitly derive the state transition matrix of the corresponding Markov chain of these two methods and to predict the convergence rates of them.

The circuit (x2) under analysis, shown in Fig. 1, consists of six transistor gates (vertical wires) and four nets (horizontal wires). Every permutation of the ordering of these six transistor gates corresponds to a state in the Markov chain. Thus, there are $6! = 720$ different states.⁶ Because $|E| (= 6)$ is small, we choose the minimum window size $|Q| = 2$. Using uniformly distributed random variables to randomize the cost functions, we are able to explicitly compute the state transition matrix of the corresponding Markov chain. To implement the random search method, we simply set the heuristic cost function $f_c \equiv 0$ during both the generalization and specialization phases. The remaining steps of the SE method and the random search methods are exactly the same. This allows a fair comparison on the effects of using a heuristic cost function to bias the search.

To simplify the presentation of the results, states having the same total net (horizontal wires) length and using the same number of tracks are grouped into the same class. For the circuit x2, as listed in Table I, there are 21 classes ranging from three to seven tracks. Among them, class 1, which contains 16 states, corresponds to the set of globally optimal solutions. Using formula presented in the previous section, we calculate the theoretical MFVT's for each class using both the random search method and the SE method. The results are listed in columns (e) and (f) in Table I. Also, in column (g), we compute the ratio of (e) to (f). Note that for the globally optimal solution (class 1), on average, the SE method requires approximately one third of steps to converge compared to the random search method. The significance of the heuristic cost function is thus quite clear. A three-dimensional (3-D) column graph shown in Fig. 5 depicts the ratios of the MFVT's of the two methods against the track number and the total net length. Again, it is evident that due to the heuristic cost function, the SE method tends to visit those states with smaller heuristic cost earlier than the random search method.

We have also conducted a Monte Carlo simulation to verify the theoretical results using experimentation. For each trial, the gate order of the circuit x2 is randomized. The evolution process starts from this initial gate order and ends when an optimal gate order with minimum net length is reached. The number of iterations is recorded as first visit time for that trial. After the trial is repeated for 120 000 times, we average the first visit times as MFVT. The empirically derived MFVT for both the SE method and the random search method, as well as the ratio of the two are listed in columns (h)–(j) in Table I. Note that with 120 000 trials, these imperial results are very close to what was predicted from theory.

V. CONCLUSION

In this paper, we analyze the convergence properties of a generic SE algorithm for solving a combinatorial optimization problem.

⁶The total number of states grows exponentially as a function of $|E|$. Thus due to hardware constraints, we are able to conduct a small scale experiment in this paper. However, the results are typical for real world problems.

TABLE I
MEAN FIRST VISIT TIME, NUMBER OF CLASSES, AND NUMBER OF STATES PER CLASS OF CIRCUIT x2

(a) class	(b) tracks	(c) total net length	(d) # states/ class	Mean First Visit Time (# steps)					
				Theory			Monte Carlo Simulation		
				(e) SE	(f) random	(g) =(e)/(f)	(h) SE	(i) random	(j) =(h)/(i)
1	3	10	16	40.26	116.13	0.35	41.60	115.45	0.36
2	4	11	16	51.37	116.13	0.44	51.98	115.70	0.45
3	4	12	24	37.16	79.16	0.47	37.51	79.12	0.47
4	4	13	16	43.82	118.54	0.37	43.73	118.04	0.37
5	5	13	8	168.31	220.00	0.77	173.06	219.21	0.79
6	5	14	72	23.79	25.65	0.93	24.15	25.57	0.94
7	5	15	56	30.71	30.34	1.01	30.76	30.32	1.01
8	5	16	56	41.25	32.03	1.29	41.15	32.16	1.28
9	5	17	32	69.45	54.29	1.28	69.08	54.20	1.27
10	5	18	16	139.33	106.58	1.31	138.50	106.61	1.30
11	5	19	8	402.21	219.63	1.83	400.57	219.77	1.82
12	6	16	16	289.55	117.52	2.46	297.94	118.08	2.52
13	6	17	64	57.15	27.59	2.07	57.64	27.59	2.09
14	6	18	32	120.78	54.86	2.2	120.10	54.96	2.19
15	6	19	32	120.75	55.66	2.17	120.66	55.43	2.18
16	6	20	64	49.30	29.81	1.65	49.98	29.68	1.68
17	6	21	48	36.71	43.74	0.84	36.04	43.77	0.82
18	7	20	16	777.03	117.52	6.61	790.30	117.24	6.74
19	7	21	48	297.44	36.73	8.10	300.17	36.65	8.19
20	7	22	48	166.33	36.83	4.52	165.99	36.79	4.51
21	7	23	32	175.16	61.27	2.86	172.86	61.17	2.83

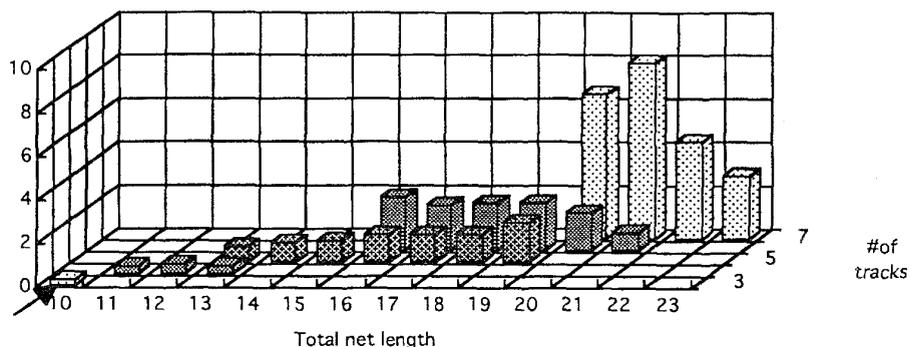


Fig. 5. Ratio of MFVT of the SE algorithm to the MFVT of random search algorithm. Smaller ratio to the globally optimal states (marked by an arrow) means faster convergence.

We first establish that the algorithm presented corresponds to an ergodic Markov chain. As such, the globally optimal state can be reached within finite number of steps. We further propose to use the MFVT to the class of globally optimal states as a criterion to compare convergence rate. We applied this technique to a CMOS gate matrix layout problem and validated the results using Monte Carlo simulation. The two approaches yield highly consistent results. We feel the major contribution of this paper is to analytically demonstrate

that the generic SE algorithm converges faster than a random search algorithm.

ACKNOWLEDGMENT

The constructive suggestions made by the anonymous reviewers and Associate Editors are deeply appreciated. The authors wish to thank R. Agrawal with the Department of Electrical and Computer

Engineering, University of Wisconsin, Madison, for helpful comments on this manuscript.

Functional Test Generation for Synchronous Sequential Circuits

M. K. Srinivas, James Jacob, and Vishwani D. Agrawal

REFERENCES

- [1] Y. G. Saab and V. B. Rao, "Combinatorial optimization by stochastic evolution," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 525–535, Apr. 1991.
- [2] ———, "An evolution-based approach to partitioning ASIC systems," in *Proc. 26th Design Automation Conf.*, 1989, pp. 767–770.
- [3] Y. G. Saab, "Combinatorial optimization by stochastic evolution with applications to the physical design of VLSI circuits," Ph.D. thesis, Univ. Illinois, Urbana, Aug. 1990.
- [4] R. M. Kling and P. Banerjee, "ESP: Placement by simulated evolution," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 245–256, Mar. 1989.
- [5] R. Kling and Banerjee, "Empirical and theoretical studies of the simulated evolution method applied to standard cell placement," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1303–1315, Oct. 1991.
- [6] Y. L. Lin, Y. C. Hsu, and F. H. S. Tsai, "SILK: A simulated evolution router," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1108–1114, Oct. 1989.
- [7] T. A. Ly and J. T. Mowchenko, "Applying simulated evolution to high level synthesis," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 389–409, Mar. 1993.
- [8] C. Y. Mao and Y. H. Hu, "SEGMA: A simulated evolution gate matrix layout algorithm," *VLSI Design*, vol. 2, no. 3, pp. 241–257, 1994.
- [9] C. Y. Mao, "Simulated evolution algorithms for gate matrix layouts," Ph.D. Thesis, Univ. Wisconsin, Madison, June 1994.
- [10] B. Mitra, P. R. Panda, and P. P. Chaudhun, "Estimating the complexity of synthesized designs from FSM specifications," *IEEE Design Test Comput.*, vol. 10, pp. 30–35, Mar. 1993.
- [11] R. V. Cherabuddi and M. A. Bayoumi, "Automated system partitioning for synthesis of multi-chip modules," in *Proc. Fourth Great Lakes Symp. VLSI, Design Automation High Perform. VLSI Syst.*, 1994, pp. 21–25.
- [12] N. Calazans, Q. Zhang, R. Jacobi, B. Yernaux, and A. Trullemans, "Advanced ordering and manipulation techniques for binary decision diagrams," in *Proc. Euro. Design Automation Conf.*, Brussels, Belgium, 1992, pp. 452–457.
- [13] J. H. Holland, *Adaptation in Neural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.
- [14] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. New York: D. Van Nostrand, 1969.
- [15] E. Seneta, *Non-Negative Matrices and Markov Chains*. New York: Springer-Verlag, 1973.
- [16] O. Wing, S. Huang, and R. Wang, "Gate matrix layout," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 220–231, July 1985.
- [17] A. Papoulis, *Probability, Random Variables, and Stochastic Process*. New York: McGraw-Hill, 1984, pp. 123–140.

Abstract—We present a novel, highly efficient functional test generation methodology for synchronous sequential circuits. We generate test vectors for the growth (G) and disappearance (D) faults using a cube description of the finite state machine (FSM). Theoretical results establish that these tests guarantee a complete coverage of stuck faults in combinational and sequential circuits, synthesized through algebraic transformations. The truth table of the combinational logic of the circuit is modeled in the form known as personality matrix (PM) and vectors are obtained using highly efficient cube-based test generation method of programmable logic arrays (PLA). Sequential circuits are modeled as arrays of time-frames and new algorithms for state justification and fault propagation through faulty PLA's are derived. We also give a fault simulation procedure for G and D faults. Experiments show that test generation can be orders of magnitude faster and achieves a coverage of gate-level stuck faults that is higher than a gate-level sequential-circuit test generator. Results on a broad class of small to large synthesis benchmark FSM's from MCNC support our claim that functional test generation based on G and D faults is a viable and economical alternative to gate level ATPG, especially in a logic synthesis environment. The generated test sequences are implementation-independent and can be obtained even when details of specific implementation are unavailable. For the ISCAS'89 benchmarks, available only in multilevel netlist form, we extract the PM and generate functional tests. Experimental results show that a proper resynthesis improves the stuck fault coverage of these tests.

I. INTRODUCTION

The growth (G) and disappearance (D) faults in the combinational function of a circuit are a subset of the faults normally modeled in the programmable logic array (PLA) implementation [1]. It is known that the tests for G and D faults cover all stuck faults in any two level implementation of the combinational logic [2]. For certain synthesis styles [3], [4], these tests will also cover all single stuck faults in the multilevel combinational circuit.

The main contribution of this paper is a sequential circuit test generation algorithm based on the G and D fault model and its implementation. Many sequential circuit test generators use the time-frame expansion method where the circuit is represented as an iterative array of its combinational logic [5]. At the core of such a method, there usually is a combinational test generation algorithm. In order to find a test sequence, the test generator repeatedly uses the combinational algorithm. Thus, the overall efficiency depends upon how well this algorithm performs. We model the combinational logic at the functional level by its personality matrix (PM) and develop an efficient cube-based test-generation algorithm to obtain test sequences for G and D faults in the finite state machine (FSM). Our recent research [2], [6] has shown the feasibility of this approach. In this paper, we give the theoretical validation of the fault model along with the algorithms and experimental results on a broad range of synthesized sequential circuits.

Manuscript received March 25, 1994; revised April 21, 1995 and March 27, 1996. This paper was recommended by Associate Editor W. K. Fuchs.

M. K. Srinivas was with the Indian Institute of Science, Bangalore 560 012, India. He is now with the CAIP Center, Rutgers University, Piscataway, NJ 08855 USA.

J. Jacob is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560 012, India.

V. D. Agrawal is with Bell Laboratories, Murray Hill, NJ 07974 USA.
Publisher Item Identifier S 0278-0070(96)05039-7.