Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

USING ACCESSIBILITY TO ASSESS THE PERFORMANCE OF GENERALIZED HILL CLIMBING ALGORITHMS

Sheldon H. Jacobson

Dept. of Industrial & Systems Engr.
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0118 , U.S.A.

Enver Yücesan

INSEAD
Technology Management Area
Boulevard de Constance
77305 Fontainebleau Cedex, FRANCE

ABSTRACT

The search problem, ACCESSIBILITY, asks whether a finite sequence of events can be found such that, starting with a specific initial event, a particular state can be reached. This problem is intractable, indicating the need for heuristics to address it. One difficulty when applying heuristics to ACCESSIBILITY is assessing a priori their effectiveness, and knowing how to best adjust them to improve performance. This paper introduces the false negative probability as a performance measure for generalized hill climbing algorithms applied to discrete optimization problems, using ACCESSIBILITY as the analysis framework. The false negative probability is also used to obtain necessary convergence conditions. The implications of these results on how GHC algorithms can be effectively applied are discussed.

1 INTRODUCTION

The discrete event simulation search problem, ACCESSIBILITY, asks whether a finite sequence of events can be found such that, starting with a specific initial event, a particular state can be reached. ACCESSIBILITY has implications for the design of discrete event dynamic systems such as telecommunication networks and manufacturing facilities. For example, message loss in a telecommunications network or machine blocking in a manufacturing process represent system states of particular interest. To avoid such states, one would be interested in studying the sequences of events that lead to these critical states. In other words, it is vital to assess high-risk scenarios to identify what must take place for such states to occur and how a system can be modified to avoid their occurrence. This issue can be addressed by solving particular instances of ACCESSIBILITY.

ACCESSIBILITY also has implications for building and analyzing discrete event simulations, which was the original motivation for studying the problem (Yücesan and Jacobson 1992). For example, identifying whether a discrete event simulation model is *logically connected* involves verifying whether a particular set of states is reachable from an initialization event (that establishes the initial state of the model) through a finite sequence of events, each of which may modify the state of the model. Moreover, a simulation model might be logically connected only for a specific set of model parameter values that defines a valid *experimental frame.* Verification of the logically connected property and determination of a valid experimental frame both require instances of ACCESSIBILITY to be solved.

Jacobson and Yücesan (1998a) prove ACCESSIBILITY to be NP-hard. A consequence of this result is that it is unlikely that a polynomial-time algorithm exists to address it. One difficulty when applying heuristics to ACCESSIBILITY is assessing *a priori* their effectiveness, and knowing how to best adjust them to improve their performance. This paper introduces the *false negative probability* for generalized hill climbing (GHC) algorithms (Johnson and Jacobson 1998) applied to ACCESSIBILITY. The false negative probability is the probability that a GHC algorithm will, in the limit, determine that a particular state can be reached, given that the algorithm could not find such a state in finite time. Therefore, the false negative probability provides a performance measure for GHC algorithms applied to ACCESSIBILITY, a means to establish the convergence of a GHC algorithm as well as a means to determine whether a GHC algorithm has been terminated prematurely.

There are several results in the literature concerning the asymptotic performance of simulated annealing algorithms. For simulated annealing algorithms with exponential acceptance probability functions, Mitra et al.

(1986) and Hajek (1988) develop conditions for three convergence properties: asymptotic independence of the starting conditions, convergence in distribution of the solutions generated, and convergence to a global optimum. They also characterize the convergence rate. Anily and Federgruen (1987) present convergence conditions for simulated annealing algorithms with general acceptance probabilities. In addition, they provide conditions for the reachability of the set of global optima. Yao and Li (1991) and Yao (1995) also discuss simulated annealing algorithms with general acceptance probabilities, though their primary contribution is with respect to general neighborhood generation distributions. More recently, Schuur (1997) provides a description of acceptance functions ensuring the convergence of the associated simulated annealing algorithm to the set of global optima.

This paper analyzes convergence properties of GHC algorithms, generalizing earlier studies on simulated annealing. The discrete event simulation search problem, ACCESSIBILITY, is used as the analysis framework, greatly simplifying the technical development. The paper is organized as follows: In Section 2, relevant concepts from simulation and discrete optimization are introduced. In Section 3, the false negative probability is formally defined. In Section 4, the false negative probability is used to obtain necessary convergence conditions for GHC algorithms, which are illustrated in Section 5 for Monte Carlo search and threshold accepting. Section 6 summarizes the results and their implications for the design of algorithms for discrete optimization problems.

2 BACKGROUND AND DEFINITIONS

To define the false negative probability, several discrete event simulation model and GHC algorithm concepts are needed.

2.1 Discrete Event Simulation Models

The following definitions are taken from Jacobson and Yücesan (1998a). Let *MS* represent a discrete event simulation *model specification*, a representation of the system under study, reflecting the objectives of the study and the assumptions of the analysis. Let *MI* represent a discrete event simulation *model implementation* of the *MS*, a translation of the model specification into a computer executable form (i.e., *MI* describes a procedure to mimic the system behavior). Therefore, a model specification defines what a model *does*, while a model implementation defines *how* the model behavior is achieved. A model specification can be in the form of a Generalized Semi-Markov Process (Shedler 1987), or an Event Graph Model (Schruben and Yücesan 1993); a model implementation can be in a high-level programming language such as Pascal or C, or in a simulation language such as SLAM (Pritsker 1995) or SIGMA (Schruben 1995).

The *state* of an *MS* is a collection of values that provide a complete description of the system. *Events* induce changes in the state of the system. We assume that there are a countable number of distinct events, D, associated with a given *MS*. For an *MS*, define E_0 to be an initial event establishing the initial state of the *MS*, S to be a particular state, and M to be a non-negative finite integer. The notation $E_0E_1...E_m \Rightarrow S$ denotes that the execution of the sequence of events leads to state S. The search problem ACCESSIBILITY is formally stated.

ACCESSIBILITY: (Jacobson and Yücesan 1998a)
INSTANCE: A discrete event simulation model specification, *MS,* with D events, and an associated discrete event simulation model implementation, *MI,* an initial event of this *MS*, E_0, a state, S, and a non-negative finite integer, M.
QUESTION: Find a sequence of events E_1, E_2,..., E_m, $m \leq M$, such that the execution of the sequence yields $E_0E_1E_2...E_m \Rightarrow S$.

EXAMPLE: Consider a single-server queueing system. The *MS* for this system can be represented using an event graph (Schruben 1995, p.24), with the *MI* of this *MS* in the simulation language SIGMA. A state of the *MS* can be defined as Q = the number of customers in the system. Define the initialization event E_0 to set $Q \leftarrow 0$ and schedule an initial arrival, where U denotes the arrival event, and C denotes the service completion event. Therefore, D=3. If the length of the event sequence is M=12, then state S=10 is accessible. On the other hand, state S≡Q=15 is valid, though not accessible for M=12 and E_0 as defined. However, if E_0 initializes the system with Q=5, then state Q=15 is accessible for M=12. Lastly, S=-2 is an invalid state, hence is not accessible. □

2.2 Generalized Hill Climbing Algorithms for Discrete Optimization Problems

Discrete optimization problems are characterized by a countably finite set of solutions and an objective function value associated with each such solution (see Garey and Johnson 1979, p.123, for a formal definition). The goal is to find solutions for which the objective function is optimized. Unless otherwise noted, assume that all discrete optimization problems are minimization problems.

For a discrete optimization problem, define the *solution space*, Ω, as the set of all possible solutions.

Define an *objective function* f: $\Omega \to [0,+\infty]$ that assigns a non-negative value to each element of the solution space. Define a *neighborhood structure* $\eta:\Omega \to 2^{\Omega}$, where $\eta(\omega) \subset \Omega$ for all $\omega \in \Omega$. The neighborhood structure establishes relationships between the solutions in the solution space, hence allows the solution space to be traversed or searched by moving between solutions. The goal is to identify a globally optimal solution ω^* (i.e., $f(\omega^*) \leq f(\omega)$ for all $\omega \in \Omega$).

GHC algorithms can be used to address intractable (NP-hard) discrete optimization problems. GHC algorithms allow inferior solutions to be visited *en route* to the optimal solution. In practice, the best solution obtained over the entire GHC algorithm run, not just the final solution, is reported. This allows the algorithm to aggressively traverse the solution space (i.e., visit many inferior solutions in search of a globally optimal solution), while retaining the best solution obtained throughout the entire GHC run. The GHC algorithm is described in pseudo-code form:

Define an objective function f: $\Omega \to [0,+\infty]$
Define a neighborhood structure $\eta:\Omega \to 2^{\Omega}$
Define the random variable $R_k:\Omega\times\Omega \to [-\infty,+\infty]$
Set the iteration indices i=k=n=1
Set the outer loop counter bound K
Set the inner loop counter bounds N(k), k=1,2,…,K
Select an initial solution $\omega(1) \in \Omega$
Repeat while $k \leq K$
Repeat while $n \leq N(k)$
Generate a solution $\omega' \in \eta(\omega(i))$
Calculate $\delta = f(\omega') - f(\omega(i))$
If $\delta<0$, then $\omega(i+1) \leftarrow \omega'$
If $\delta \geq 0$ and $R_k(\omega(i), \omega') \geq \delta$, then $\omega(i+1) \leftarrow \omega'$
If $\delta \geq 0$ and $R_k(\omega(i), \omega') < \delta$, then $\omega(i+1) \leftarrow \omega(i)$
$n \leftarrow n+1$
$i \leftarrow i+1$
Until n=N(k)
n=1
$k \leftarrow k+1$
Until k=K

Several general search strategies can be described using the GHC algorithm framework. For example, simulated annealing can be described as a GHC algorithm by setting $R_k(\omega(i),\omega') = -t_k \ln(u(i))$, $\omega(i) \in \Omega$, $\omega' \in \eta(\omega(i))$, where t_k is the temperature parameter (hence, defines a cooling schedule as $t_k \to 0$) and u(i) are independent and identically distributed U(0,1) random variables. Other search strategies that have been represented as GHC algorithms include threshold accepting (Dueck and Scheuer 1990), tabu search (Glover and Laguna 1997), Monte Carlo search, local search, and Weibull accepting (Johnson and Jacobson 1997, 1998).

2.3 Applying GHC Algorithms to Address ACCESSIBILITY

To apply GHC algorithms to ACCESSIBILITY, define the solution space, Ω, to be the set of all possible event sequences of length M, starting with event E_0. Note that some of these event sequences may be invalid, or valid only up to the k^{th} event, $k \in \{0,1,\ldots,M\}$. Define the neighbors of $\omega \in \Omega$, $\eta(\omega)$, by changing exactly one of the events in ω (i.e., a one-change rule). Define the objective function as $f(\omega) = \min_{k=0,1,\ldots,M} \| S_k - S \|$, where S_k is the state reached after executing the first k events of ω (Yücesan and Jacobson 1996), and $\| \bullet \|$ is a norm on the state space. Therefore, ω^* is a global minimum for this objective function (i.e., $f(\omega^*)=0$) if and only if $S_k=S$ for some k=0,1,…,M. Note that if executing the first k events of ω results in an invalid event sequence (not defined in *MS*), then $S_k=+\infty$. Moreover, the neighborhood structure and objective function defined here are not unique; different neighborhood structures and objective functions may result in distinct GHC algorithm implementations and performance results.

Define A to be a GHC algorithm that can be implemented to address an instance of ACCESSIBILITY. For a given *MS* with D events and its associated *MI*, where all event sequences are initialized with event E_0, with state S and $M \in Z^+$ given, define the solution space as the set of all event sequences of length at most M. Define two events

B(S,I,M)≡{algorithm A reports NO for state S after I iterations}, (1)

and

B(S,M)≡{algorithm A reports NO for state S} (2)

These two events are distinct in that B(S,I,M) is for algorithm A executed over a finite number of iterations I, while B(S,M) has no such limitation. The algorithm reporting YES (NO) means that algorithm A could (not) find a sequence of at most M events that reaches state S. Note that events B(S,I,M) and B(S,M) are not events of an *MS*, but rather, events as defined in probability theory (Jacobson and Yücesan 1998b). The complementary events, $B^c(S,I,M)$ and $B^c(S,M)$, are defined as

$B^c(S,I,M)$≡{algorithm A reports YES for state S after I or fewer iterations} (3)

and

$B^c(S,M)$≡{algorithm A reports YES for state S}. (4)

Also define the event $Q_S(I,M) \equiv B^c(S,M) \cap B(S,I,M)$, with $q_S(I,M) \equiv P\{Q_S(I,M)\}$. GHC algorithms can be viewed as sampling procedures over the solution space Ω. The key distinction between different GHC algorithms is in *how* the sampling is performed. For example, Monte Carlo search produces unbiased samples from the solution space, while simulated annealing produces biased samples, guided by the neighborhood structure, the objective function, and the temperature parameter.

2.4 The False Negative Probability

The definition of $B(S,I,M)$ in (1) implies that $B(S,I,M) \supseteq (B(S,I+1,M)$, for all $I \in Z^+$, hence $\{B(S,I,M)\}$ is a telescoping, non-increasing sequence of events. Therefore, since the probability function is a continuous set function, then, by the Monotone Convergence Theorem,

$$P\{B(S,I,M)\} \rightarrow P\{B(S,M)\} \text{ as } I \rightarrow \infty,$$

where

$$B(S,M) = \bigcap_{I=1}^{\infty} B(S,I,M).$$

Therefore, for a given (fixed) initial solution $\omega(1)$, if algorithm A is guaranteed to converge to state S, then $P\{B^c(S,M)\}=1$. Equivalently, if $P\{B^c(S,M)\}<1$, then algorithm A cannot be guaranteed to converge. Convergence results for the GHC algorithms are fully explored in Section 4.

In light of these observations, the *false negative* problem asks whether a GHC algorithm, in the limit, will establish that a state is accessible, given that the GHC algorithm, executing a finite number of iterations, is unable to show that it is accessible. This problem is quantified by considering the *false negative probability*, defined as $P\{B^c(S,M) \mid B(S,I,M)\}$.

3 FALSE NEGATIVE PROBABILITY RESULTS

3.1 Analysis Framework

Determining the false negative probability requires an understanding of the relationship between $B^c(S,M)$ and $B(S,I,M)$. To this end, since $Q_S(I,M)=B^c(S,M) \cap B(S,I,M)$, then

$$P\{B(S,I,M)\} = P\{B(S,M)\} + q_S(I,M) \qquad (5)$$

where $q_S(I,M)$ measures the probability gap between $B(S,M)$ and $B(S,I,M)$. Therefore, $P\{B^c(S,M) \mid B(S,I,M)\}$

$$= P\{B^c(S,M) \cap B(S,I,M)\}/P\{B(S,I,M)\}$$
$$= q_S(I,M) / P\{B(S,I,M)\}. \qquad (6)$$

If $P\{B^c(S,M)\}=0$ for state S, then it is not necessary to run algorithm A, since $B^c(S,M)$ and $B(S,I,M)$ are independent for all $I \in Z^+$. This means that $P\{B^c(S,M) \mid B(S,I,M)\} = P\{B^c(S,M)\} = 0$, which also follows from (6). If $P\{B^c(S,M)\}=1$, then (5) implies $P\{B^c(S,I,M)\} + q_S(I,M) = 1$ for all $I \in Z^+$. Therefore, $P\{B^c(S,M) \mid B(S,I,M)\}= 1$ for all $I \in Z^+$, hence only one iteration of algorithm A needs to be executed. In general, it is unlikely that $P\{B^c(S,M)\}$ would be known *a priori*. The relationship between $B(S,M)$, $B(S,I,M)$, and $Q_S(I,M)$ is depicted in Figure 1. This figure shows that $B(S,M) \subseteq B(S,I,M)$, and $P\{B(S,I,M)\} = P\{B(S,M)\}+q_S(I,M)$.

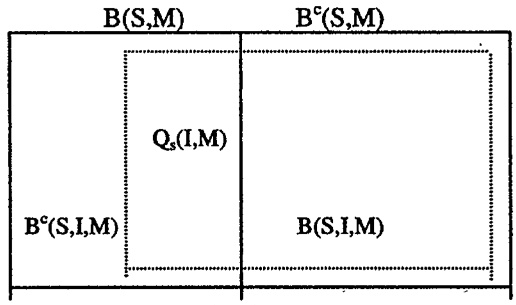

Figure 1. The relationship between $B(S,M)$, $B(S,I,M)$, and $Q_s(I,M)$

3.2 GHC Algorithm Scenarios

The false negative probability depends upon the effectiveness of a GHC algorithm to report YES for state S. In particular, four cases can be considered.

Case a: State S is not accessible. For this case, $P\{B^c(S,M)\} = 0$, hence, the false negative probability is equal to zero for all $I \in Z^+$.

Case b: State S is accessible but $P\{B^c(S,M)\}=0$. This is the case where the GHC algorithm is ineffective, irrespective of the run length. For this situation, the false negative probability is equal to zero for all $I \in Z^+$.

Case c: State S is accessible and $P\{B^c(S,M)\}=1$. This is the case where the GHC algorithm is guaranteed to find state S, provided the algorithm run length is sufficiently long. For this situation, the false negative probability is equal to one for all $I \in Z^+$.

Case d: State S is accessible and $0 < P\{B^c(S,M)\} < 1$. This is the case where the GHC algorithm may converge, though this is not guaranteed. For this situation, the false negative probability is between zero and one for all $I \in Z^+$.

Each of these four cases is encountered in practice. Case a occurs when no solution to ACCESSIBILITY exists; namely, either state S is invalid or there are no sequences of length M that reach state S. Case b occurs, for example, if the algorithm is a local search procedure and always gets trapped in a local (not global) optimum. Case c occurs, for example, if the algorithm is Monte Carlo search or simulated annealing with a convergent cooling schedule. Case d occurs, for example, if the algorithm is simulated annealing and a cooling schedule guaranteeing convergence is not used, though convergence is possible. This paper focuses on Case d: state S is accessible and the purpose of the GHC algorithm is to identify an event sequence that reaches state S.

EXAMPLE (continued): Suppose M=4. There are 2^4=16 possible event sequences of length four, with six of these sequences fully valid (i.e., the entire event sequence of length M=4 is valid), and the remaining ten only partially valid (i.e., the event sequence is valid only for the first k events, where k is either 0, 1, 2, or 3). Table 1 lists these 16 event sequences, where an asterisk indicates that the event sequence is fully valid.

If Monte Carlo search is implemented, then each of the 16 sequences of events has probability 1/16 of being selected. The probability that Monte Carlo search executes for I iterations with all NO responses, for state S, is $P\{B(S,I,M)\}=[P\{B(S,1,M)\}]^I$, where $P\{B(0,1,M)\}=0$, $P\{B(1,1,M)\}=1/2$, $P\{B(2,1,M)\}=11/16$, $P\{B(3,1,M)\}=7/8$, $P\{B(4,1,M)\}=15/16$, and $P\{B(S,1,M)\}=1$ for $S \notin \{0,1,2,3,4\}$. Moreover, for accessible states, namely 0, 1, 2, 3, and 4, $P\{B^c(S,M)\}=1$, hence $P\{B(S,I,M)\}=q_S(I,M)$.

When Monte Carlo search is applied, the false negative probability is one for all iterations and for all accessible states, as indicated by (6). Conversely, for all non-accessible states (i.e., $S\notin\{0,1,2,3,4\}$), the false negative probability is zero for all iterations. This behavior is a consequence of Monte Carlo search asymptotically guaranteeing to find state S (this corresponds to Case c). □

Table 1: Event Sequences for a Single-Server Queueing Simulation Model

#	Event Sequence	State(s) Reached
1	IUUUU*	0,1,2,3,4
2	IUUUC*	0,1,2,3
3	IUUCU*	0,1,2
4	IUCUU*	0,1,2
5	ICUUU	0
6	IUUCC*	0,1,2
7	IUCUC*	0,1
8	IUCCU	0,1
9	ICCUU	0
10	ICUCU	0
11	ICUUC	0
12	IUCCC	0,1
13	ICUCC	0
14	ICCUC	0
15	ICCCU	0
16	ICCCC	0

4 CONVERGENCE PROBABILITIES FOR GHC ALGORITHMS

In this section, necessary convergence conditions are derived for a GHC algorithm using the false negative probability. Without loss of generality, assume that $P\{B(S,1,M)\}=1$ and $P\{B^c(S,M)\}>0$. Furthermore, unless otherwise stated, assume that there exists $I'\in Z^+$ such that $0< P\{B^c(S,I,M)\}<1$ for all $I\geq I'$. Define the (probability) event

R(S,I,M) ≡ {GHC algorithm A reports YES for state S at iteration I, given that A reported NO for state S up to iteration I-1}, (7)

with

$$r(S,I,M) \equiv P\{R(S,I,M)\} = P\{B^c(S,I,M) \mid B(S,I-1,M)\}. \quad (8)$$

These probability measures can be used to quantify the false negative probability. To this end, Lemma 1 establishes a closed-form expression for (1) in terms of (8).

LEMMA 1:

(i) $P\{B(S,I,M)\} = \prod_{i=1}^{I}[1-r(S,i,M)]$ for all $I\in Z^+$.

(ii) $P\{B(S,M)\} = \prod_{i=1}^{\infty}[1-r(S,i,M)]$.

Proof: By the definition of $r(S,i,M)$, i=1,2,…,I, in (8),

$$\begin{aligned}1 - r(S,I,M) &= P\{B(S,I,M) \mid B(S,I-1,M)\} \\ &= P\{B(S,I,M)\cap B(S,I-1,M)\}/P\{B(S,I-1,M)\} \\ &= P\{B(S,I,M)\} / P\{B(S,I-1,M)\}.\end{aligned}$$

Therefore, since $P\{B(S,1,M)\}=1$, then

$$P\{B(S,I,M)\} = \prod_{i=1}^{I}[1-r(S,i,M)] .$$

Taking the limit as $I \to +\infty$ establishes (ii). □

Define the *finite false negative probability*, $P\{B^c(S,J,M) \mid B(S,I,M)\}$, $I,J\in Z^+$, $J>I$. Lemma 2 gives an expression for this probability in terms of (8).

LEMMA 2: For $J > I$, $I,J\in Z^+$,

$$P\{B^c(S,J,M) \mid B(S,I,M)\} = 1-\prod_{i=I+1}^{J}[1-r(S,i,M)] .$$

Proof: Since {B(S,I,M)} are telescoping non-increasing events, then using (8),

$$P\{B^c(S,J,M) \mid B(S,I,M)\} = [P\{B(S,I,M)\} - P\{B(S,J,M)\}]/P\{B(S,I,M)\}$$

$$= \frac{\prod_{i=1}^{I}[1-r(S,i,M)] - \prod_{i=1}^{J}[1-r(S,i,M)]}{\prod_{i=1}^{I}[1-r(S,i,M)]}$$

$$= 1 - \prod_{i=I+1}^{J}[1-r(S,i,M)].\ \square$$

Theorem 1 gives an explicit expression for the false negative probability.

THEOREM 1: For all $I \in Z^+$,

$$P\{B^c(S,M) \mid B(S,I,M)\} = 1 - \prod_{i=I+1}^{\infty}[1-r(S,i,M)].$$

Proof: Taking the limit as $J \to +\infty$ in Lemma 2 establishes the result. $\square$

COROLLARY 1: $P\{B^c(S,M) \mid B(S,I,M)\} \geq P\{B^c(S,M) \mid B(S,I+1,M)\}$ for all $I \in Z^+$.

Proof: Since

$$\prod_{i=I+1}^{\infty}(1-r(S,i,M)) \leq \prod_{i=I+2}^{\infty}(1-r(S,i,M)),$$

the result follows from Theorem 1. $\square$

Corollary 1 shows that the false negative probability is non-increasing in I; that is, the probability of finding a sequence of events reaching state S can never increase with each iteration that does not find such a sequence. This result can be used as a guideline for execution termination (Jacobson and Yücesan 1998b). The following results pertain to the finite-time performance of GHC algorithms.

THEOREM 2: For J>I, $I,J \in Z^+$, $P\{B(S,I,M) \mid B^c(S,J,M)\}$

$$= \frac{\prod_{i=1}^{I}[1-r(S,i,M)] - \prod_{i=1}^{J}[1-r(S,i,M)]}{1-\prod_{i=1}^{J}[1-r(S,i,M)]}.$$

Proof: Applying Bayes theorem to Lemma 2 yields the result. $\square$

COROLLARY 2: $P\{B(S,I,M) \mid B^c(S,M)\}$

$$= \frac{\prod_{i=1}^{I}[1-r(S,i,M)] - \prod_{i=1}^{\infty}[1-r(S,i,M)]}{1-\prod_{i=1}^{\infty}[1-r(S,i,M)]}$$

for all $I \in Z^+$.

Proof: Taking the limit as $J \to +\infty$ in Theorem 2 establishes the result. $\square$

Note that the probability in Corollary 2 is non-increasing in I. Therefore, for all $\varepsilon>0$ (close to 0), there exists $I_0 \in Z^+$, such that, for all $I \geq I_0$, $P\{B(S,I,M) \mid B^c(S,M)\} \leq \varepsilon$. This means that a GHC algorithm run can be terminated once $P\{B(S,I,M) \mid B^c(S,M)\}$ is sufficiently small, since the marginal value for each additional iteration is negligible.

To establish the relationship between the convergence in probability of a GHC algorithm and the false negative probability, the following definition is needed.

DEFINITION: GHC algorithm **A** *converges in probability* (to a sequence of at most M events that reaches state S) if $P\{C(S,I,M)\} \to 1$ as $I \to +\infty$, where

C(S,I,M) = {algorithm **A** reports YES for state S at iteration I}. $\square$

Theorem 3 establishes the relationship between the convergence in probability of a GHC algorithm and the false negative probability.

THEOREM 3: If GHC algorithm **A** converges in probability, then $P\{B^c(S,M) \mid B(S,I,M)\} = 1$ for all $I \in Z^+$.

Proof: By the definition of conditional probability, $P\{B(S,M) \mid B(S,I,M)\} = P\{B(S,M) \cap B(S,I,M)\} / P\{B(S,I,M)\} = P\{B(S,M)\}/P\{B(S,I,M)\}$ for all $I \in Z^+$. Note that $C(S,J,M) \subseteq B^c(S,J,M)$ for all $J \in Z^+$, hence, $B(S,J,M) \subseteq C^c(S,J,M)$ for all $J \in Z^+$. Therefore, $P\{B(S,M)\}/P\{B(S,I,M)\}$

$$\leq \lim_{J \to +\infty} P\{C^c(S,J,M)\}/P\{B(S,I,M)\} = 0$$

for all $I \in Z^+$. This implies that $P\{B^c(S,M) \mid B(S,I,M)\} = 1$ for all $I \in Z^+$. $\square$

From Theorem 3, if a GHC algorithm converges in probability, then the false negative probability is equal to one at all iterations. Theorem 4 gives necessary and sufficient conditions for the false negative probability to be one at all iterations.

THEOREM 4: For a GHC algorithm A, $P\{B^c(S,M) \mid B(S,I,M)\} = 1$ for all $I \in Z^+$ if and only if

$$\sum_{i=I+1}^{\infty} r(S,i,M) = +\infty \text{ for all } I \in Z^+.$$

Proof: Follows from Jacobson and Yücesan (1998b). □

Corollary 4 provides a necessary condition for the convergence in probability of GHC algorithms.

COROLLARY 4: If a GHC algorithm converges in probability, then

$$\sum_{i=I+1}^{\infty} r(S,i,M) = +\infty \text{ for all } I \in Z^+.$$

Proof: The result follows from Theorems 3 and 4. □

5 ILLUSTRATIONS

Theorem 4 can be used to show that the false negative probability is one (for all iterations) for Monte Carlo search, which can be described as a GHC algorithm by setting $\eta(\omega(i))=\Omega$ for all $\omega(i) \in \Omega$, and $R_k=+\infty$ for all $k \in Z^+$. Define $p(S)=|\Omega(S)|/|\Omega|$, where $\Omega(S)$ is the set of event sequences in Ω that solve ACCESSIBILITY. Then, $r(S,I,M)=p(S)$. Therefore, $P\{B(S,I,M)\}=[1-p(S)]^I$. From Lemma 2, for $J>I$, $I,J \in Z^+$, $P\{B^c(S,J,M) \mid B(S,I,M)\}=1-[1-p(S)]^{J-1}$. Hence, the false negative probability approaches one as J approaches infinity. Moreover, from Theorem 4, $\sum_{i=I+1}^{+\infty} r(S,i,M) = \sum_{i=I+1}^{+\infty} p(S) = +\infty$ for all $I \in Z^+$. This means that, with probability one, Monte Carlo search will find an event sequence that solves ACCESSIBILITY as k approaches infinity. However, $P\{C(S,I,M)\}=p(S)$ for all $I \in Z^+$, hence, Monte Carlo search does not converge in probability.

Corollary 4 can be used to show that particular forms of threshold accepting do not converge in probability. Threshold accepting is a particular GHC algorithm with $R_k(\omega(i),\omega') = T_k$, $\omega(i) \in \Omega$, $\omega' \in \eta(\omega(i))$, for all k, where T_k approaches zero as $k \to +\infty$. Note that as T_k becomes sufficiently close to zero, $P\{R_k(\omega(i),\omega') \geq \delta\} = 0$ for all $\omega(i) \in \Omega$, $\omega' \in \eta(\omega(i))$, with $\delta = f(\omega')-f(\omega(i))>0$. Therefore, there exists an I_0 such that $r(S,I,M)=0$ for all $I \geq I_0$ (Jacobson and Yücesan 1998b). Therefore, from Corollary 4, if T_k approaches zero as $k \to +\infty$, then threshold accepting does not converge in probability.

6 CONCLUDING COMMENTS

This paper defines the false negative probability as a performance measure for GHC algorithms applied to discrete optimization problems, using ACCESSIBILITY as the analysis framework. The relationship between convergence in probability and the false negative probability is also discussed. These results can be used to guide the design and execution of effective algorithms for discrete optimization problems (Jacobson and Yücesan 1998b).

The use of ACCESSIBILITY as the analysis framework greatly simplifies the development of our results. Such cross-fertilization of concepts between discrete event simulation and discrete optimization provides a rich avenue of investigation, to obtain useful tools that can benefit both fields. The work presented here is one of many directions that can provide new and useful insights into difficult unresolved issues in both domains.

ACKNOWLEDGEMENTS

The first author is supported in part by the National Science Foundation (DMI-9409266) and the Air Force Office of Scientific Research (F49620-95-1-0124, F49620-98-1-0111).

REFERENCES

Anily, S., and A. Federgruen. 1987. Simulated annealing methods with general acceptance probabilities. *Journal of Applied Probability.* 24: 657-667.

Dueck, G. and T. Scheuer. 1990. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *European Journal of Operational Research.* 46: 271-281.

Garey, M.R. and D.S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Fransisco: Freeman and Company.

Glover, F. and M. Laguna. 1997. *Tabu Search.* Norwell: Kluwer Academic Publishing.

Hajek, B. 1988. Cooling schedules for optimal annealing. *Mathematics of Operations Research.* 13: 311-329.

Jacobson, S.H. and E. Yücesan. 1998a. On the complexity of verifying structural properties of discrete event simulation models. Forthcoming in *Operations Research.*

Jacobson, S.H. and E. Yücesan. 1998b. On the effectiveness of discrete optimization heuristics," Working Paper. INSEAD. Fontainebleau, France.

Johnson, A.W. and S.H. Jacobson. 1997. A convergence result for a class of generalized hill climbing algorithms," Technical Report, Department of

Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA.

Johnson, A.W., and S.H. Jacobson. 1998. On the convergence of Generalized Hill Climbing Algorithms. Technical Report, Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA.

Mitra, D., F. Romeo, and A.L. Sangiovanni-Vincentelli. 1986. Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probability.* 18: 747-771.

Pritsker, A.A.B. 1995. *Introduction to Simulation and SLAM II.* 4th Edition. W. Lafayette, IN: Systems Publishing Company.

Schruben, L.W. 1995. *Graphical Simulation Modeling and Analysis: Using SIGMA for Windows.* Boyd and Fraser Publishing Company: The Scientific Press Series, Danvers, Massachusetts.

Schruben, L., and E. Yücesan. 1993. Modeling paradigms for discrete event simulation. *Operations Research Letters.* 13: 265-275.

Schuur, P.C. 1997. Classification of acceptance criteria for the simulated annealing algorithm. *Mathematics of Operations Research.* 22: 266-275.

Shedler, G.L. 1987. *Regeneration and Networks of Queues.* New York: Springer-Verlag.

Yao, X. 1995. A new simulated annealing algorithm. *International Journal of Computer Mathematics.* 56: 161-168.

Yao, X., and G. Li. 1991. General simulated annealing. *Journal of Computer Science and Technology.* 6: 329-338.

Yücesan, E. and S.H. Jacobson. 1992. Building correct simulation models is difficult. In *Proceedings of the 1992 Winter Simulation Conference.* 783-790.

Yücesan, E. and S.H. Jacobson. 1996. Intractable structural issues in discrete events simulation: special cases and heuristic approaches. *ACM Transactions on Modeling and Computer Simulation.* 6: 53-75.

AUTHOR BIOGRAPHIES

SHELDON H. JACOBSON is an Associate Professor and Director of the Simulation and Optimization Laboratory in the Department of Industrial and Systems Engineering at Virginia Polytechnic Institute and State University (Virginia Tech). He has a B.Sc. and M.Sc. (both in Mathematics) from McGill University, and a M.S. and Ph.D. (both in Operations Research and Industrial Engineering) from Cornell University. His theoretical research interests include discrete event simulation, applied probability, computational complexity and algorithm development for intractable discrete optimization problems. His applied research and consulting interests address problems in the manufacturing, service, and health-care industries. His research has been published in a wide spectrum of journals, including Operations Research, INFORMS Journal on Computing, Operations Research Letters, IIE Transactions, and the Journal of the Operational Research Society. He has received research funding from the National Science Foundation and the Air Force Office of Scientific Research. He has also collaborated on projects with the Federal Aviation Administration, through the FAA Center of Excellence in Operations Research at Virginia Tech, and the Center for Disease Control and Prevention.

ENVER YÜCESAN is an Associate Professor of Operations Research at the Technology Management Area at INSEAD in Fontainebleau, France. He holds a BSIE degree from Purdue University, and an MS and a Ph.D. both in OR, from Cornell University. The work described in this paper has been initiated while he was visiting the Department of Systems Engineering at the University of Pennsylvania.