Journal of Global Optimization **29:** 173–190, 2004.

Global Optimization Performance Measures for Generalized Hill Climbing Algorithms

SHELDON H. JACOBSON[1] and ENVER YÜCESAN[2]
[1]*Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street (MC-244), Urbana, IL 61801-2906, USA*
[2]*Technology Management Area, Boulevard de Constance, INSEAD 77305 Fontainebleau Cedex, France*

(Received 25 September; accepted in revised form 4 June 2003)

Abstract. Generalized hill climbing algorithms provide a framework for modeling several local search algorithms for hard discrete optimization problems. This paper introduces and analyzes generalized hill climbing algorithm performance measures that reflect how effectively an algorithm has performed to date in visiting a global optimum and how effectively an algorithm may perform in the future in visiting such a solution. These measures are also used to obtain a necessary asymptotic convergence (in probability) condition to a global optimum, which is then used to show that a common formulation of threshold accepting does not converge. These measures assume particularly simple forms when applied to specific search strategies such as Monte Carlo search and threshold accepting.

Mathematics Subject Classification. Analysis of algorithms, Suboptimal algorithms, Probability, Stochastic model applications, Simulation, Efficiency

Key words: Convergence, Finite-time performance, Hill climbing, Local search algorithms

1. Introduction

Discrete optimization problems are defined by a finite set of solutions and an objective function value associated with each solution (Garey and Johnson, 1979). The goal when addressing such problems is to determine the set of solutions for which the objective function is optimized (i.e., minimized or maximized). Heuristic procedures are typically formulated with the hope of finding good or near-optimal solutions for intractable (NP-hard) discrete optimization problems. Generalized Hill Climbing (GHC) algorithms (Johnson and Jacobson, 2002a, b), such as simulated annealing (Kirkpatrick et al., 1983) and threshold accepting (Dueck and Scheuer, 1990), are a class of general local search strategies, offering a means to find reasonable solutions to a wide variety of discrete optimization problems. The objective of these algorithms is to find the best

possible solution using a limited amount of computing resources. A further challenge is to construct algorithms that find near-optimal solutions for all instances of a particular problem, since the effectiveness of many algorithms tends to be problem-specific, exploiting particular characteristics of problem instances (e.g., Lin and Kernighan, 1973, for the traveling salesman problem). It is therefore important to assess the performance of algorithms and devise strategies to improve their effectiveness in solving hard discrete optimization problems.

There are several results in the literature concerning the asymptotic performance of simulated annealing algorithms. Mitra et al. (1986) and Hajek (1988) provide some of the earliest results, where they 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. For a complete review of both theoretical and practical results for simulated annealing, see Aarts and Korst (2002) or Henderson et al. (2003).

The current literature focuses mainly on asymptotic convergence properties. This paper considers asymptotic as well as finite-time performance measures of GHC algorithms where convergence cannot be guaranteed. These measures are used to develop necessary convergence conditions for GHC algorithms. Note that the results presented do not provide readily deployable tools for improving the performance of GHC algorithms, but rather, provide a mathematical framework under which the performance of such algorithms can be described (see Orosz and Jacobson, 2002a, b, for examples to illustrate the application of this framework). The most practical result arising from this framework is a necessary convergence condition that provides a proof of nonconvergence for common implementations of threshold accepting.

The paper is organized as follows: In Section 2, the GHC algorithm framework is described, including relevant concepts from discrete optimization. In Section 3, the false-negative probability performance measure for GHC algorithms is introduced and formally defined. In Section 4, this probability is used to obtain finite-time performance results for non-convergent GHC algorithms, as well as a necessary convergence condition for such algorithms. This necessary condition provides a proof of nonconvergence for common implementations of threshold accepting (as presented in Section 6). Upper and lower bounds for the false negative probability are also derived. Section 5 illustrates how these probability measures can be represented. Section 6 illustrates these expressions for particular GHC algorithms. Section 7 summarizes the results presented.

2. Background and Definitions

2.1. GENERALIZED HILL CLIMBING ALGORITHMS FOR DISCRETE OPTIMIZATION PROBLEMS

For a discrete (minimization) optimization problem, define the *solution space*, Ω, as a finite 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. Two important components of GHC algorithms are the *neighborhood function*, $\eta: \Omega \to 2^{\Omega}$, where $\eta(\omega) \subseteq \Omega$ for all $\omega \in \Omega$, and the *hill climbing random variables* $R_k: \Omega \times \Omega \to \Re$, $k = 1, 2, \ldots$ For each solution $\omega \in \Omega$, the neighborhood function $\eta(\omega)$ defines a set of solutions that are close to ω (Aarts and Korst, 2002). The neighborhood function is assumed to be symmetric (i.e., if $\omega' \in \eta(\omega'')$, then $\omega'' \in \eta(\omega')$ for all ω', $\omega'' \in \Omega$) and that $\omega \in \eta(\omega)$ for all $\omega \in \Omega$. Moreover, at each iteration of a GHC algorithm, a solution is randomly generated among all neighbors of the current solution by a neighborhood probability mass function, where the resulting random variables are independent (given the current solution). For example, neighbors are said to be generated uniformly at each iteration of a GHC algorithm execution if, for all $\omega \in \Omega$, with $\omega' \in \eta(\omega)$, $P\{\omega'$ is selected as the neighbor of ω at a given iteration of a GHC algorithm$\} \equiv h_{\omega}(\omega') = 1/|\eta(\omega)|$. Without loss of generality, assume that if $\omega' \in \eta(\omega)$, then $h_{\omega}(\omega') > 0$.

The GHC algorithm is described in pseudo-code form below. By definition, the hill climbing random variables, R_k, map points in $\Omega \times \Omega$ to distributions that determine whether a randomly generated neighboring solution is accepted during a particular inner loop iteration associated with outer loop iteration k. The hill climbing random variables are assumed to be independent. The stopping criterion for the inner loops, *STOP INNER*, determines when the hill climbing random variable index k increments by one, hence a new hill climbing random variable is used to accept or reject neighboring solutions. By setting the *STOP INNER* criterion to check whether the current solution is a local optimum, the hill climbing random variable changes only when a local optimum is visited; this will be further discussed in Section 2.2.

Although the range of the hill climbing random variables can be the set of real numbers, $\Re$, in practice they are typically restricted to the set of non-negative real numbers, $\Re^{+}$ (which is what will be assumed for the rest of the paper). Therefore, for minimization problems, when a randomly generated neighboring solution has objective function value greater than the current solution, then the neighboring solution is accepted (hence becomes the new current solution) if the difference between the objective function values is not too large (i.e., smaller than the value generated for

the hill climbing random variable). This concept of accepting an inferior solution is the origin for the name "hill climbing".

Define a neighborhood function η and a set of hill climbing random variables R_k
Set the iteration indices $i = 0, k = 1$ and select an initial solution $\omega(0) \in \Omega$
Repeat
  Repeat
    Generate a neighboring solution$\omega \in \eta(\omega(i))$ according to $h_{\omega(i)}(\omega)$
    Compute $\delta(\omega(i), \omega) = f(\omega) - f(\omega(i))$
    Generate an observation R from the random variable $R_k(\omega(i), \omega)$
    If $R \geqslant \delta(\omega(i), \omega)$, then $\omega(i+1) \leftarrow \omega$ (accept improving or hill climbing moves)
    If $R < \delta(\omega(i), \omega)$, then $\omega(i+1) \leftarrow \omega(i)$ (reject hill climbing moves)
    $i \leftarrow i + 1$
  Until *STOP INNER*
  $k \leftarrow k + 1$
Until *STOP OUTER*

Assume that the hill climbing random variables have finite means and finite variances (i.e., $E[|R_k(\omega(i), \omega)|] < +\infty$ and $\mathrm{Var}[R_k(\omega(i), \omega)] < +\infty$ for all $\omega(i) \in \Omega, \omega \in \eta(\omega(i)), k = 1, 2, \ldots, i = 1, 2, \ldots$).

The neighborhood function establishes relationships between the solutions in the solution space, hence allows the solution space to be traversed or searched by moving between solutions. To ensure that the solution space is not fragmented, assume that all the solutions in the solution space (with neighborhood function η and neighborhood probability mass function $h(\omega)$) are *reachable* (i.e., for all $\omega', \omega'' \in \Omega$, there exists a set of solutions $\omega_1, \omega_2, \ldots, \omega_m \in \Omega$ such that $\omega_r \in \eta(\omega_{r-1})$, $r = 1, 2, \ldots, m+1$, where $\omega' \equiv \omega_0$ and $\omega'' \equiv \omega_{m+1}$). If all solutions in the solution space are reachable, then the solution space (with neighborhood function η) is said to be reachable. Note that solution space fragmentation can be a problem, for example, in some implementations of tabu search with a deterministic tabu list. Fox (1993) describes a clever method on avoiding fragmentation altogether. The objective function, f, and the neighborhood function, η, allow the solution space, Ω, to be decomposed into three mutually exclusive and collectively exhaustive sets:

- a set of global optima, $G = \{\omega^* \in \Omega : f(\omega^*) \leqslant f(\omega) \text{ for all } \omega \in \Omega\}$;

- a set of local optima that are not global optima, $L \equiv L(\eta) = \{\omega \in \Omega \backslash G : f(\omega) \leqslant f(\omega') \text{ for all } \omega' \in \eta(\omega)\}$;
- a set of hill solutions, $H = \Omega \backslash (G \cup L)$.

Therefore $G \cup L$ are the set of local optima in Ω associated with neighborhood function η, where by definition, $\Omega = G \cup L \cup H$ with $G \cap L = \varnothing$, $G \cap H = \varnothing$, and $L \cap H = \varnothing$. Note also that for all $\omega \in G, \eta(\omega) \cap L = \varnothing$, and for all $\omega \in L$, $\eta(\omega) \cap G = \varnothing$ (i.e., a global optimum and a local optimum cannot be neighbors).

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 visiting many inferior solutions en route to a globally optimal solution, while retaining the best solution obtained through the entire GHC run. By design, GHC algorithms are sampling procedures over the solution space Ω. For example, Monte Carlo search generates independent samples (with replacement) from the solution space, while simulated annealing generates samples guided by the neighborhood function, the objective function, and the temperature parameter. More specifically, simulated annealing can be described as a GHC algorithm by setting $R_k(\omega(i), \omega) = -t(k) \ln(v_i), \omega(i) \in \Omega, \omega \in \eta(\omega(i)), k = 1, 2, \ldots$, where $t(k)$ is the temperature parameter (hence, defines a cooling schedule as $t(k) \to 0$) and v_i are independent and identically distributed $U(0, 1)$ random variables. Note that in the "accept improving or hill climbing moves" step of the GHC algorithm pseudo-code, for the simulated annealing hill climbing random variable, $R_k(\omega(i), \omega) \geqslant \delta(\omega(i), \omega)$ becomes $v_i \leqslant \exp -\delta(\omega(i), \omega)/t(k)$, which is the standard form in which the simulated annealing hill climbing acceptance probability is described (Aarts and Korst, 2002). Other algorithms that can be described using the GHC framework include threshold accepting (Dueck and Scheuer, 1990), some simple forms of tabu search (Glover and Laguna, 1997), Monte Carlo search, deterministic local search, the noising method (Charon and Hudry, 2001), and Weibull accepting (see Jacobson et al., 1998 and Johnson and Jacobson, 2002a, b, for a discussion on how these algorithms can be fit into the GHC algorithm framework).

2.2 CLASSIFYING THE ITERATIONS FOR GENERALIZED HILL CLIMBING ALGORITHMS

The iterations of a GHC algorithm can be classified using the concept of macro iterations. A *macro iteration* is a set of consecutive iterations that move the algorithm from any element of $G \cup L$ to any element of $G \cup L$ (including itself), where the solutions at any intervening iterations are (not necessarily distinct) elements of H. From the pseudo-code in Section 2.1,

by requiring that the *STOP INNER* criterion checks whether the current solution is a local optimum, then the outer loops correspond to macro iterations. If there are a polynomial number of neighboring solutions of the current solution or the neighborhood of the current solutions can be searched in polynomial time, then verifying that the current solution is a local optimum can be done in polynomial time. Assume that this is the case, hence local optimality can be verified in polynomial time.

Using this *STOP INNER* criterion, at macro iteration k fixed, the iterations can be modeled as a homogeneous discrete-time Markov chain, with $|\Omega| \times |\Omega|$ *transition matrix*

$$P^k = \begin{bmatrix} P^k_{GG} & P^k_{GL} & P^k_{GH} \\ P^k_{LG} & P^k_{LL} & P^k_{LH} \\ P^k_{HG} & P^k_{HL} & P^k_{HH} \end{bmatrix},$$

where the entries of P^k denote the single iteration transition probabilities between all elements of Ω. Without loss of generality, assume that the GHC algorithm run is initialized at a solution $\omega(0) \in L$, since local search can be applied from any element in Ω, and the solution space is reachable. This places a restriction on the classes of discrete optimization problems that can be studied, since if a local optimum cannot be obtained in polynomial time in the size of the problem instance, then initializing the GHC algorithm run in this way may not be feasible (see Johnson et al., 1988; Jacobson and Solow, 1993). In addition, if local search is applied and the local optimum obtained is a global optimum, then the problem is solved, though this may not be known until further iterations are executed.

In the pseudo-code presented in Section 2.1, for k fixed, the macro iterations can also be modeled as a homogeneous discrete-time Markov chain, with a $(|G| + |L|) \times (|G| + |L|)$ *macro iteration transition matrix*,

$$P^k_M = \begin{bmatrix} P^k_{GH}\left[\sum_{j=0}^{+\infty}(P^k_{HH})^j\right]P^k_{HG} + P^k_{GG} & P^k_{GH}\left[\sum_{j=0}^{+\infty}(P^k_{HH})^j\right]P^k_{HL} + P^k_{GL} \\ P^k_{LH}\left[\sum_{j=0}^{+\infty}(P^k_{HH})^j\right]P^k_{HG} + P^k_{LG} & P^k_{LH}\left[\sum_{j=0}^{+\infty}(P^k_{HH})^j\right]P^k_{HL} + P^k_{LL} \end{bmatrix},$$

where the entries represent the probability of a GHC algorithm moving from any element of $G \cup L$ to any element of $G \cup L$ (including itself), passing only through elements of H (Sullivan and Jacobson, 2001). If P^k_{HH} is the zero matrix, then set $(P^k_{HH})^0 \equiv \mathbf{I}$, the identity matrix. Matrix P^k_M can be simplified, since for all $\omega \in G$, $\eta(\omega) \cap L = \varnothing$, and for all $\omega \in L$, $\eta(\omega) \cap G = \varnothing$ (i.e., a global optimum and a local optimum cannot be neighbors), hence P^k_{GL} and P^k_{LG} are both zero matrices. Moreover, if a global optimum cannot be a neighbor of another global optimum, and a local

optimum cannot be a neighbor of another local optimum, then P_{GG}^k and P_{LL}^k are both diagonal square matrices.

3. The False Negative Probability for Generalized Hill Climbing Algorithms

Consider a GHC algorithm applied to an instance of a discrete optimization problem. Assume that $R_k(\omega(i), \omega) \geqslant 0$ for all $\omega(i) \in \Omega, \omega \in \eta(\omega(i))$, for all outer loop, macro iterations $k = 1, 2, \ldots$. At each macro iteration k, define the event

$$B(k) \equiv \{\text{The algorithm does not visit any element of } G \text{ over first } k \text{ macro iterations}\} \tag{1}$$

and its complementary event

$$B^c(k) \equiv \{\text{The algorithm visits } G \text{ over the first } k \text{ macro iterations}\}. \tag{2}$$

By definition, $B(k) \supseteq B(k+1)$ for all macro iterations k, hence $\{B(k)\}$ is a telescoping, non-increasing sequence of events in k. Therefore, by the Monotone Convergence Theorem (Billingsley, 1979),

$$P\{B(k)\} \to P\{B\} = P\left\{\bigcap_{k=1}^{+\infty} B(k)\right\} \text{as } k \to +\infty. \tag{3}$$

Over the first k macro iterations, the algorithm visits k solutions, $\{\omega_1, \omega_2, \ldots, \omega_k\} \subseteq G \cup L$. Define f^k to be the minimum objective function value among these k solutions and ω^k to be the associated solution (i.e., $f^k = f(\omega^k)$ with $\omega^k = \text{argmin}\{f(\omega_j),\ j = 1, 2, \ldots, k\}$). In practice, the best solution to date (i.e., ω^k) is reported. The key issue is whether $\omega^k \in G$. If $\omega^k \in G$, then the algorithm should be terminated no later than macro iteration k, while if $\omega^k \notin G$, then it would be desirable to determine whether the algorithm will at some future macro iteration visit a solution in G. Therefore, $P\{\omega^k \in G\} = P\{B^c(k)\}$ provides an algorithm performance measure for the solutions obtained within the first k macro iterations.

To establish the relationship between the convergence of a GHC algorithm and the event B, the following definition is needed.

DEFINITION 1. A GHC algorithm *converges in probability* to G if $P\{C(k)\} \to 1$ as $k \to +\infty$, where $C(k) \equiv \{\omega_k \in G\} = \{\text{The algorithm is at an element of } G \text{ at macro iteration } k\}$.

Therefore, given an initial solution $\omega(0) \in L$, if a GHC algorithm converges in probability to G (as $k \to +\infty$), then $P\{B^c\} = 1$. Equivalently, if $P\{B^c\} < 1$, then the algorithm does not converge in probability to G. The convergence behavior of GHC algorithms is further investigated in Section 4.

In light of these observations, the false-negative problem asks whether a GHC algorithm will eventually visit G, given that the algorithm, after executing a finite number of macro iterations, has yet to visit G. The false negative probability is formally defined.

DEFINITION 2. For a GHC algorithm, the *false-negative probability* at macro iteration k is $P\{B^c|B(k)\}$, provided $P\{B(k)\} > 0$.

The false-negative probability at macro iteration k provides a measure for the *effectiveness* of a GHC algorithm, namely the ability of an algorithm to visit G beyond macro iteration k. In particular, if $P\{B^c\}$ is small, then one can use the false-negative probability to assess whether a GHC algorithm will eventually visit G; if the false-negative probability at macro iteration k is sufficiently close to zero, then the algorithm may be terminated.

4. Asymptotic Properties of Generalized Hill Climbing Algorithms

This section derives a necessary convergence condition for GHC algorithms. Recall that $P\{B(0)\} = 1$ (i.e., all GHC algorithm runs are initialized at an element of L). Furthermore, unless otherwise stated, assume that $P\{B^c(k)\} < 1$ for all macro iterations $k = 1, 2, \ldots$.

For macro iteration k, define the conditional probability

$$r(k) \equiv P\{B^c(k)|B(k-1)\} = P\{C(k)|B(k-1)\}. \tag{4}$$

This probability can be used to quantify the false-negative probability. lemma 1. expresses the relationship between (4) and (1).

LEMMA 1. *Given a GHC algorithm initialized at solution* $\omega(0) \in L$,

(i) $P\{B(k)\} = \prod_{j=1}^{k}[1 - r(j)]$ *for all macro iterations* k.

(ii) $P\{B\} = \prod_{j=1}^{+\infty}[1 - r(j)]$.

Proof. By the definition of $r(j)$, $1 - r(j) = P\{B(j)|B(j-1)\} = P\{B(j) \cap B(j-1)\}/P\{B(j-1)\} = P\{B(j)\}/P\{B(j-1)\}$. Therefore, since $P\{B(0)\} = 1$, then (i) holds. Letting $k \to +\infty$ in (i) establishes (ii). □

Theorem 1 provides a closed form expression for the false-negative probability.

THEOREM 1. *Given a GHC algorithm initialized at solution* $\omega(0) \in L$, *for all macro iterations* k *with* $P\{B(k)\} > 0$,

$$P\{B^c|B(k)\} = 1 - \prod_{j=k+1}^{+\infty}[1 - r(j)]. \tag{5}$$

Proof. Since $\{B(k)\}$ are telescoping non-increasing events, then using (4), for all macro iterations m and k, $m > k$, with $P\{B(k)\} > 0$

$$P\{B^c(m)|B(k)\} = [P\{B(k)\} - P\{B(m)\}]/P\{B(k)\}$$
$$= \left[\prod_{j=1}^{k}[1 - r(j)] - \prod_{j=1}^{m}[1 - r(j)]\right] \Bigg/ \prod_{j=1}^{k}[1 - r(j)]$$
$$= 1 - \prod_{j=k+1}^{m}[1 - r(j)].$$

Taking the limit as $m \to +\infty$ establishes (5). □

Theorem 2 provides upper and lower bounds for the false-negative probability.

THEOREM 2. *Given a GHC algorithm initialized at initial solution* $\omega(0) \in L$, *then for all macro iterations* k *with* $P\{B(k)\} > 0$,

$$1 - \exp\left\{-\sum_{j=k+1}^{+\infty} r(j)\right\} \leqslant P\{B^c|B(k)\} \leqslant 1 - \exp\left\{-\sum_{j=k+1}^{+\infty} [r(j)]/[1 - r(j)]\right\}. \tag{6}$$

Proof. For all macro iterations k with $P\{B(k)\} > 0$, from Lemma 1, $r(j) < 1$ for all $j = 1, 2, \ldots, k$. Therefore, for all macro iterations m and k with $m > k$ and $P\{B(m)\} > 0$, hence $r(j) < 1$ for all $j = 1, 2, \ldots, m$,

$$1 - \exp\left\{-\sum_{j=k+1}^{m} r(j)\right\} \leqslant P\{B^c(m)|B(k)\}$$
$$\leqslant 1 - \exp\left\{-\sum_{j=k+1}^{m} [r(j)]/[1 - r(j)]\right\}.$$

To see this, from the proof of Theorem 1, for $m > k$, $P\{B^c(m)|B(k)\} = 1 - \prod_{j=k+1}^{m}[1 - r(j)]$. Therefore, $P\{B(m)|B(k)\} = \prod_{j=k+1}^{m}[1 - r(j)]$, which implies that

$$\ln[P\{B(m)|B(k)\}] = \sum_{j=k+1}^{m} \ln[1 - r(j)].$$

For $0 < r(j) < 1$, $j = k + 1, k + 2, \ldots, m$,
$-r(j)/[1 - r(j)] \leqslant \ln[1 - r(j)] \leqslant -r(j)$.
Therefore,

$$-\sum_{j=k+1}^{m} [r(j)]/[1 - r(j)] \leqslant \ln[P\{B(m)|B(k)\}] \leqslant -\sum_{j=k+1}^{m} r(j).$$

Taking the exponential function and the limit as $m \to +\infty$ establishes the result. □

To compute the false-negative probability for both convergent and non-convergent GHC algorithms, Proposition 1 establishes the relationship between convergence in probability to G and visits to G in probability.

PROPOSITION 1. *If a GHC algorithm converges in probability to G, then the GHC algorithm visits G in probability* (*i.e.*, $P\{B^c|B(k)\} = 1$ *for all macro iterations* $k = 1, 2, \ldots$ *with* $P\{B(k)\} > 0$).

Proof. By the definition of conditional probability,

$$P\{B|B(k)\} = P\{B \cap B(k)\}/P\{B(k)\}$$
$$= P\{B\}/P\{B(k)\} \text{ for all macro iterations } k = 1, 2, \ldots$$

Since $C(m) \subseteq B^c(m)$, then $B(m) \subseteq C^c(m)$ for all macro iterations $m = 1, 2, \ldots$ Therefore, for all macro iterations $k = 1, 2, \ldots$

$$P\{B\}/P\{B(k)\} \leqslant \lim_{m \to +\infty} P\{C^c(m)\}/P\{B(k)\} = 0$$

hence $P\{B^c|B(k)\} = 1$ for all macro iterations $k = 1, 2, \ldots$ □

Proposition 2 provides necessary and sufficient conditions for a GHC algorithm to visit G in probability (i.e., the false-negative probability is one for all macro iterations).

PROPOSITION 2. *A GHC algorithm visits G in probability if and only if* $\sum_{j=1}^{+\infty} r(j) = +\infty$.

Proof. ($\Leftarrow$) If $\sum_{j=1}^{+\infty} r(j) = +\infty$, then $\sum_{j=k+1}^{+\infty} r(j) = +\infty$ for all macro iterations k, hence from the lower bound in (6), $P\{B^c|B(k)\} = 1$ for all macro iterations k with $P\{B(k)\} > 0$. Therefore, the GHC algorithm visits G in probability. ($\Rightarrow$) If the GHC algorithm visits G in probability (i.e., $P\{B^c|B(k)\} = 1$ for all macro iterations k with $P\{B(k)\} > 0$), then from the upper bound in (6), $\sum_{j=k+1}^{+\infty} r(j)/[1 - r(j)] = +\infty$ for all macro iterations k. To complete the proof, it is necessary to show that $\sum_{j=k+1}^{+\infty} r(j)/[1 - r(j)] = +\infty$ implies that $\sum_{j=1}^{+\infty} r(j) = +\infty$. To see this, suppose that $\sum_{j=1}^{+\infty} r(j) < +\infty$. Then for all $\varepsilon > 0$, there exists a non-negative integer $j(\varepsilon)$ such that $r(j) \leqslant \varepsilon$ for all $j \geqslant j(\varepsilon)$. Therefore,

$$r(j)/(1 - r(j)) \leqslant r(j)/(1 - \varepsilon)$$

for all $j \geqslant j(\varepsilon)$, which implies that

$$\sum_{j=j(\varepsilon)}^{+\infty} r(j)/(1 - r(j)) \leqslant \sum_{j=j(\varepsilon)}^{+\infty} r(j)/(1 - \varepsilon). \tag{7}$$

However, if the right hand side of (7) is finite, then the left hand side must

also be finite, which contradicts that the left hand side diverges. Therefore, $\sum_{j=1}^{+\infty} r(j) = +\infty$. □

Proposition 3 establishes the relationship between a GHC algorithm visiting G in probability and $P\{B^c\}$.

PROPOSITION 3. *A GHC algorithm visits G in probability if and only if $P\{B^c\} = 1$.*

Proof. The result follows from the law of total probability, since
$$P\{B^c\} = P\{B^c|B(k)\}P\{B(k)\} + P\{B^c|B^c(k)\}P\{B^c(k)\}.$$ □

Theorem 3 summarizes the relationship between $P\{B^c\}$, the false negative probabilities, $r(k)$, visits G in probability, and convergence in probability to G.

THEOREM 3. *Given a GHC algorithm initialized at initial solution $\omega(0) \in L$, consider the expressions*

(*D*1) $P\{C(k)\} \to 1$ as $k \to +\infty$ (*converges in probability to G*).
(*D*2) $P\{B^c|B(k)\} = 1$ *for all macro iterations k* (*visits G in probability*).
(*D*3) $P\{B^c\} = 1$ (*visits G in probability*).
(*D*4) $\sum_{j=1}^{+\infty} r(j) = +\infty$ *for all macro iterations k.*

Then $(D1) \Rightarrow (D2) \Leftrightarrow (D3) \Leftrightarrow (D4)$.

Proof. Follows from Propositions 1, 2, and 3. □

Theorem 3 provides three necessary conditions for the convergence of a GHC algorithm. The only restriction on how the GHC algorithm traverses the solution space is that $P\{B(k)\} > 0$ for all macro iterations $k = 1, 2, \ldots$ This restriction means that there is no finite-time convergence to G with probability one. Note that from Lemma 1, if $P\{B\} = \prod_{j=1}^{+\infty}[1 - r(j)] > 0$, then from Theorem 3, the GHC algorithm does not converge in probability to G. Moreover, since $C(k) \subseteq B^c(k)$ for all macro iterations $k = 1, 2, \ldots$, then $P\{C(k)\} \leqslant 1 - \prod_{j=1}^{+\infty}[1 - r(j)]$ for all macro iterations k.

5. False-Negative Probability Representation

This section derives closed-form expressions for $r(k)$ such that the results in Section 4 can be applied to GHC algorithms, including the computation of the false-negative probability and condition (D4) in Theorem 3. To obtain such an expression, the following definition is needed to represent $r(k)$ as a function of the macro iteration transition matrix, P_M^k.

DEFINITION 3. For all $\omega \in L$, at macro iteration k, $Q(\omega, k) \equiv B(k) \cap \{\text{The algorithm is at solution } \omega \text{ at macro iteration } k\}$ with $q(\omega, k) = P\{Q(\omega, k)\}$. Theorem 4 provides a closed-form expression for $r(k)$ in terms of the macro iteration transition matrix. This expression is used in Section 6 to identify properties of non-convergent GHC algorithms.

THEOREM 4. *Given a GHC algorithm initialized at initial solution $\omega(0) \in L$, then for all macro iterations k,*

$$r(k) = \sum_{\omega_2 \in L} \sum_{\omega_1 \in G} q(\omega_2, k-1) P^k_{LH}(\omega_2, \bullet) \left[\sum_{j=0}^{+\infty} \left(P^k_{HH}\right)^j \right] P^k_{HG}(\bullet, \omega_1). \tag{8}$$

Proof. By the law of total probability,

$$\begin{aligned} r(k) &= \sum_{\omega_2 \in L} q(\omega_2, k-1) P\{B^c(k) | Q(\omega_2, k-1)\} \\ &= \sum_{\omega_2 \in L} \sum_{\omega_1 \in G} q(\omega_2, k-1) P^k_{LH}(\omega_2, \bullet) \left[\sum_{j=0}^{+\infty} (P^k_{HH})^j \right] P^k_{HG}(\bullet, \omega_1). \end{aligned}$$ □

The closed-form expression for $r(k)$ in Theorem 4 can be expressed in terms of the hill climbing random variable R_k. To see this, for all $\omega_1 \in G, \omega_2 \in L$, $\omega_3, \omega_4 \in H$,

$$P^k_{LH}(\omega_2, \omega) = \begin{cases} (1/|\eta(\omega_2)|) P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\}, & \omega \in \eta(\omega_2) \cap H \\ 0, & \omega \notin \eta(\omega_2) \cap H \end{cases} \tag{9}$$

and

$$P^k_{HH}(\omega_3, \omega_4) = \begin{cases} (1/|\eta(\omega_3)|) P\{R_k(\omega_3, \omega_4) \geqslant \delta(\omega_3, \omega_4)\}, & \begin{cases} \delta(\omega_3, \omega_4) > 0, \\ \omega_4 \in \eta(\omega_3) \cap H, \end{cases} \\ 1/|\eta(\omega_3)|, & \begin{cases} \delta(\omega_3, \omega_4) \leqslant 0, \\ \omega_4 \in \eta(\omega_3) \cap H, \end{cases} \\ 0, & \begin{cases} \omega_3 \in H, \\ \omega_4 \notin \eta(\omega_3) \cap H. \end{cases} \end{cases} \tag{10}$$

Moreover, for all $\omega_1 \in G$, $\omega \in H$,

$$P^k_{HG}(\omega, \omega_1) = \begin{cases} (1/|\eta(\omega)|), & \omega_1 \in \eta(\omega) \cap G, \\ 0, & \omega_1 \notin \eta(\omega) \cap G. \end{cases} \tag{11}$$

To determine whether $\sum_{k=1}^{+\infty} r(k)$ converges, only the most dominant terms in (8) need to be considered (i.e., the terms in (8) that approach zero the slowest as $k \to +\infty$). From (9)–(11), the most dominant terms in (8) are $O(P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\})$ for $\omega_2 \in L$, $\omega \in \eta(\omega_2) \cap H$ (as $R_k \to_P 0$ as $k \to +\infty$). Therefore, if the hill climbing random variables are defined such that the probability of moving from any local optimum in a *single iteration* converges to zero sufficiently fast (as the number of macro iterations

approaches infinity) so that the infinite sum (over k) of the $r(k)$ converges (i.e., (D4)), then from Theorem 3, the resulting GHC algorithm will not converge in probability to G. This necessary condition provides a simple feature to check for a given GHC algorithm, hence can be used to determine when a particular GHC does not converge.

To use this result in practice, $P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\}$, $\omega_2 \in L$, $\omega \in \eta(\omega_2) \cap H$, can be bounded above using the first and second moments of $R_k(\omega_2, \omega)$. For example, from Markov's inequality,

$$P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\} \leqslant E[R_k(\omega_2, \omega)]/\delta(\omega_2, \omega) \tag{12}$$

for $\omega_2 \in L$, $\omega \in \eta(\omega_2) \cap H$, (where $\delta(\omega_2, \omega) > 0$). Moreover, by the one-sided Chebyshev inequality,

$$P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\} \leqslant \mathrm{Var}[R_k(\omega_2, \omega)]/[\mathrm{Var}[R_k(\omega_2, \omega)] + (\delta(\omega_2, \omega) - E[R_k(\omega_2, \omega)])^2]. \tag{13}$$

If either of these upper bounds approaches zero sufficiently fast such that $\sum_{k=1}^{+\infty} r(k) < +\infty$, then the GHC algorithm does not converge in probability to G.

To illustrate the use of these bounds, consider a simulated annealing algorithm with temperature parameters $t(k) = 1/k^2$. Then $E[R_k(\omega_2, \omega)] = 1/k^2$ and from (12), $P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\} \leqslant 1/(k^2\delta(\omega_2, \omega))$, which implies that $r(k) < +\infty$. Therefore, from Theorem 3, this simulated annealing algorithm does not converge in probability to G. On the other hand, for a simulated annealing algorithm with temperature parameters $t(k) = 1/k$, $E[R_k(\omega_2, \omega)] = 1/k$ and from (12), $P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\} \leqslant 1/(k\delta(\omega_2, \omega))$, which is not sufficient (from Theorem 3) to show that this simulated annealing algorithm does not converge in probability to G. However, $\mathrm{Var}[R_k(\omega_2, \omega)] = 1/k^2$ and from (13), $P\{R_k(\omega_2, \omega) \geqslant \delta(\omega_2, \omega)\} \leqslant (1/k^2)/[1/k^2 + (\delta(\omega_2, \omega) - 1/k)^2] = O((1/k\delta(\omega_2, \omega))^2)$ for k large, which implies that $\sum_{k=1}^{+\infty} r(k) < +\infty$. Therefore, from Theorem 3, this simulated annealing algorithm does not converge in probability to G.

6. Illustrative Examples

The results presented in Sections 3–5 can be used to assess the performance of various GHC algorithms. In this section, the performance of four GHC algorithms, Monte Carlo search, random-restart local search, threshold accepting, and simulated annealing are evaluated.

6.1. MONTE CARLO SEARCH

Monte Carlo search is the process of randomly generating a large set of solutions in the solution space and taking the best solution among those

generated. Theorem 3 implies that the false-negative probability is one (for all macro iterations) for Monte Carlo search. To see this, Monte Carlo search can be described as a GHC algorithm by setting $\eta(\omega) = \Omega$ for all $\omega \in \Omega$, and $R_k = \max|f(\omega) - f(\omega')|$, $\omega \in \Omega$, $\omega' \in \eta(\omega)$ for all macro iterations $k = 1, 2, \ldots$. If $p(G) \equiv |G|/(|G| + |L|)$, then $r(k) = p(G)$. Therefore, $P\{B(k)\} = [1 - p(G)]^k$. For macro iterations j and k, with $m > k$,

$$P\{B^c(m)|B(k)\} = 1 - [1 - p(G)]^{m-k},$$

which approaches one as $m \to +\infty$. Moreover, from Theorem 3, $\sum_{j=1}^{+\infty} r(j) = \sum_{j=1}^{+\infty} p(G) = +\infty$. This means that Monte Carlo search visits G in probability as $k \to +\infty$. However, $P\{C(k)\} = p(G)$ for all macro iterations k, hence Monte Carlo search does not converge in probability to G (i.e., from Theorem 3, (D2), (D3), and (D4) all hold, but (D1) is not satisfied).

6.2. RANDOM-RESTART LOCAL SEARCH

Random-restart local search (or multi-start local search; see Marti, 2003) combines Monte Carlo search and local search, by randomly selecting a new initial solution every time a local search algorithm terminates at a local optimum. The analysis in Section 6.1 for Monte Carlo search also shows that the false-negative probability is one (for all macro iterations) for random-restart local search, by redefining $p(G)$ to be the probability that a randomly generated initial solution in Ω will terminate at an element of G. Moreover, random-restart local search will not converge in probability to G (i.e., from Theorem 3, (D2), (D3), and (D4) all hold, but (D1) is not satisfied).

6.3. THRESHOLD ACCEPTING

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 macro iteration k, where $t(k) \to 0$ as $k \to +\infty$. Therefore, there exists $\varepsilon > 0$ sufficiently small and a macro iteration k_0 such that $|t(k)| < \varepsilon$ and $P\{R_k(\omega(i), \omega) \geqslant \delta(\omega(i), \omega)\} = 0$ for all $\omega(i) \in L, \omega \in \eta(\omega(i))$, and all $k \geqslant k_0$, hence (D4) in Theorem 3 does not hold. This implies that this common implementation of threshold accepting does not converge in probability to G. However, if $t(k)$ is set such that it does not approach zero, hence $r(k) \geqslant \delta$ for some $\delta > 0$ and for all macro iterations k, then (D4) in Theorem 3 may hold and the probability of a false negative is one at all macro iterations k. However, setting $t(k)$ in this way may not be feasible in practice, since it requires full knowledge of the solution space (with respect to the depth of all local and global optima; see Hajek, 1988). Although the given formulation of threshold accepting does not converge in probability to G, it often yields satisfactory results in practice (Nissen and Paul, 1995; Abboud et al., 1998; Franz

et al., 2001). This observation further supports the belief that asymptotic convergence is not necessarily a good predictor of finite-time performance.

6.4. SIMULATED ANNEALING

Simulated annealing is a particular GHC algorithm with $R_k(\omega(i), \omega) = -t(k)\ \ln(v_i), \omega(i) \in \Omega, \omega \in \eta(\omega(i)), k = 1, 2, \ldots$, where $t(k)$ is the temperature parameter (hence, defines a cooling schedule as $t(k) \to 0$) and $\{v_i\}$ are independent and identically distributed $U(0, 1)$ random variables. The necessary condition (D4) in Theorem 3 can be related to the convergence conditions for simulated annealing presented in Hajek (1988). In particular, Hajek (1988) shows that simulated annealing converges in probability to a global optimum if and only if $\sum_{k=1}^{+\infty} e^{-(d^*/t(k))} = +\infty$, where the temperature parameters $t(k)$ define a nonincreasing cooling schedule (that approaches zero as $k \to +\infty$), and d^* is the maximum depth of all local optima (i.e., the maximum gap in objective function value between an element of L and the solution in H that can reach an element of G via local search, where the maximum is taken over all elements of L). This result assumes that the depth of all elements in G is infinity, hence once a global optimum is reached, simulated annealing cannot escape from it (with probability one). Since the neighborhood function η is defined such that the solution space is reachable, then at each macro iteration k that is sufficiently large, there is a positive probability that the algorithm will need to escape from each element of L and move to an element of G. In particular, at each macro iteration k sufficiently large, the conditional probability $r(k)$ has a component that includes the probability of escaping from the deepest local optimum. Therefore, using the law of total probability,

$$r(k) = \sum_{\omega \in L} r(k|\omega \in L \text{ is visited at macro iteration } k-1)$$
$$P\{\omega \in L \text{ is visited at macro iteration } k-1\}.$$

Therefore, there exists a lower bound for $r(k)$ that is a linear function of $P\{\text{moving from the deepest element of } L \text{ to an element of } G\} = P\{\text{Accepting hill climbing moves out of the deepest element of } L \text{ to an element of } G\} = O(e^{-(d^*/t(k))})$, since the hill climbing random variable at macro iteration k is exponential with mean $1/t(k)$. Therefore, if $\sum_{k=1}^{+\infty} e^{-(d^*/t(k))} = +\infty$, then condition (D4) in Theorem 3 is satisfied.

Another consequence of Theorem 3 is that different simulated annealing algorithms may not converge in probability to G, but they may visit G in probability. For example, fixed temperature implementations of simulated annealing are provably non-convergent (since the temperature parameter does not approach zero), but visit G in probability (since $r(k) > \varepsilon > 0$ for all k for some ε fixed). Cohn and Fielding (1999) and

Fielding (2000) present interesting theoretical and empirical results that suggest that there is an optimal fixed temperature for simulated annealing for different classes of problems. Orosz and Jacobson (2002a, b) also present results with fixed temperature simulated annealing algorithms, including analytical expressions for the expected number of iterations needed to reach a prespecified objective function value. The results in Theorem 3 are consistent with the observations in Cohn and Fielding (1999) and Fielding (2000). Moreover, from Lemma 1, the rate at which $\prod_{j=1}^{k}[1 - r(j)]$ converges to zero as $k \to +\infty$ (or equivalently, the rate at which $\sum_{j=1}^{k} r(j)$ diverges to infinity as $k \to +\infty$) may provide a measure for comparing two fixed temperature simulated annealing algorithms. Work is in progress to study this measure and provide a practical approach for such an analysis, hence provide an alternative means to obtain the results reported in Cohn and Fielding (1999) and Fielding (2000).

7. Summary

This paper introduces the false negative probability as a performance measure that reflects how effectively a GHC algorithm has performed to date in visiting a global optimum as well as how effectively a GHC algorithm can be expected to perform in the future to visit such a solution. This paper also presents expressions and bounds for the false negative probability, and illustrates how these expressions and bounds can be computed for particular GHC algorithms. These expressions are used to create necessary convergence conditions for GHC algorithms. One of these conditions is then used to prove that a common implementation of threshold accepting is nonconvergent.

Work is in progress to show how the false negative probability can be used to develop guidelines to design effective run strategies for GHC algorithms that do not converge in probability to a global optimum. These guidelines can be useful in determining stopping conditions once the marginal value of additional iterations is deemed negligible. Work is also in progress to identify new performance measures for GHC algorithms that complement the false negative probability.

On a broader scale, the framework developed in this paper shows how different GHC algorithms for discrete optimization problems can be compared and evaluated using a single performance measure. The results in this paper represent a first step towards the development of a performance theory for GHC algorithms, hence foster new avenues for future research on the evaluation of GHC algorithm performance, independent of the specific problems being addressed.

Acknowledgements

This research has been supported in part by the Air Force Office of Scientific Research (F49620-01-1-0007) and the National Science Foundation (DMI-9907980). The authors would like to thank the editor, Professor P.M. Pardalos, and two anonymous referees for their helpful comments and feedback on the paper that has resulted in a significantly improved manuscript.

References

1. Aarts, E. and Korst, J. (2002), Selected topics in simulated annealing. In: Hansen, P. and Ribeiro, C.C. (eds), *Essays and Surveys on Metaheuristics*, pp. 1–37. Kluwer Academic Publishers, Norwell, MA.
2. Abboud, N., Sakawa, M. and Inuiguchi, M. (1998), School scheduling using threshold accepting, *Cybernetics and Systems* 29(6), 593–611.
3. Berman, A. and Plemmons, R.J. (1994), *Nonnegative Matrices in the Mathematical Sciences*, Society for Industrial and Applied Mathematics, Philadelphia, PA.
4. Billingsley, P. (1979), *Probability and Measure*, John Wiley and Sons, New York.
5. Charon, I. and Hudry, O. (2001), The noising method: a generalization of some metaheuristics, *European Journal of Operational Research* 135(1), 86–101.
6. Cohn, H. and Fielding, M. (1999), Simulated annealing: searching for an optimal temperature schedule, *SIAM Journal of Optimization* 9(3), 779–802.
7. Dueck, G. and Scheuer, T. (1990), Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing, *European Journal of Operational Research* 46, 271–281.
8. Fielding, M. (2000), Simulated annealing with an optimal fixed temperature, *SIAM Journal of Optimization* 11(2), 289–307.
9. Fox, B.L. (1993), Integrating and accelerating tabu search, simulated annealing, and genetic algorithms, *Annals of Operations Research* 41, 47–67.
10. Franz A., Hoffmann K.H. and Salamon P. (2001), Best possible strategy for finding ground states, *Physical Review Letters* 86(23), 5219–5222.
11. Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability*: *A Guide to the Theory of NP-Completeness,* Freeman and Company, San Francisco, CA.
12. Glover, F. and Laguna, M. (1997), *Tabu Search,* Kluwer Academic Publishing, Norwell, MA.
13. Hajek, B. (1988), Cooling schedules for optimal annealing, *Mathematics of Operations Research* 13, 311–329.
14. Henderson, D., Jacobson, S.H. and Johnson, A.W. (2003), The theory and practice of simulated annealing. In: Glover, F. and Kochenberger, G. (eds), *Handbook on Metaheuristics*, pp. 287–319. Kluwer Academic Publishers, Norwell, MA.
15. Jacobson, S.H. and Solow, D. (1993), The effectiveness of finite improvement algorithms for finding global optima, *Zeitschrift fur Operations Research* (*ZOR*) – *Methods and Models of Operations Research* 37(3), 257–272.
16. Johnson, A.W. and Jacobson, S.H. (2002a), On the convergence of generalized hill climbing algorithms, *Discrete Applied Mathematics* 119(1–2), 37–57.
17. Johnson, A.W. and Jacobson, S.H. (2002b), A class of convergent generalized hill climbing algorithms, *Applied Mathematics and Computation* 125(2–3), 359–373.

18. Johnson, D.S., Papadimitriou, C.H. and Yannakakis, M. (1988), How easy is local search? *Journal of Computers and Systems Science* 37(1), 79–100.
19. Kirkpatrick, S., Gelatt, C.D., Jr. and Vecchi, M.P. (1983), Optimizaton by simulated annealing, *Science* 220, 671–680.
20. Lin, S. and Kernighan, B.W. (1973), An effective heuristic for the traveling salesman problem. *Operations Research* 21, 498–516.
21. Marti, R. (2003), Multi-start methods. In: Glover, F. and Kochenberger, G. (eds.), *Handbook on Metaheuristics*, pp. 355–368. Kluwer Academic Publishers, Norwell, MA.
22. Mitra, D., Romeo, F. and Sangiovanni-Vincentelli, A.L. (1986), Convergence and finite-time behavior of simulated annealing, *Advances in Applied Probability* 18, 747–771.
23. Nissen, V. and Paul, H. (1995), A modification of threshold accepting and its application to the quadratic assignment problems, *OR Spekrum* 17(2–3), 205–210.
24. Orosz, J.E. and Jacobson, S.H. (2002a), Finite-time performance analysis of static simulated annealing algorithms, *Computational Optimization and Applications* 21(1), 21–53.
25. Orosz, J.E. and Jacobson, S.H. (2002b), Analysis of static simulated annealing algorithms, *Journal of Optimization Theory and Application* 115(1), 165–182.
26. Sullivan, K.A. and Jacobson, S.H. (2001), A convergence analysis of generalized hill climbing algorithms, *IEEE Transactions on Automatic Control* 46(8), 1288–1293.