Analyzing the Performance of Local Search Algorithms Using Generalized Hill Climbing Algorithms

Article *in* Operations Research/ Computer Science Interfaces Series · January 2002

DOI: 10.1007/978-1-4615-1507-4_20

Sheldon H Jacobson
University of Illinois Urbana-Champaign

Journal of Heuristics, 10: 387–405, 2004

Analyzing the Performance of Generalized Hill Climbing Algorithms*

SHELDON H. JACOBSON
Simulation and Optimization Laboratory, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana, IL 61801-2906, USA
email: shj@uiuc.edu(netfiles.uiuc.edu/shj/www/shj.html)

ENVER YÜCESAN
Technology Management Area, INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France
email: enver.yucesan@insead.edu(http://www.insead.edu/facultyresearch/tm/yucesan)

Submitted in January 2003 and accepted by David Woodruff in February 2004 after 1 revision

Abstract

Generalized hill climbing algorithms provide a framework to describe and analyze metaheuristics for addressing intractable discrete optimization problems. The performance of such algorithms can be assessed asymptotically, either through convergence results or by comparison to other algorithms. This paper presents necessary and sufficient convergence conditions for generalized hill climbing algorithms. These conditions are shown to be equivalent to necessary and sufficient convergence conditions for simulated annealing when the generalized hill climbing algorithm is restricted to simulated annealing. Performance measures are also introduced that permit generalized hill climbing algorithms to be compared using random restart local search. These results identify a solution landscape parameter based on the basins of attraction for local optima that determines whether simulated annealing or random restart local search is more effective in visiting a global optimum. The implications and limitations of these results are discussed.

Key Words: meta-heuristics, simulated annealing, performance evaluation, convergence

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, p. 123). 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 hard (i.e., NP-hard) discrete optimization problems (Garey and Johnson, 1979). Generalized Hill Climbing (GHC) algorithms (Jacobson et al., 1998), such as simulated annealing (Kirkpatrick et al., 1983), the noising method (Charon and

*This research is supported in part by the Air Force Office of Scientific Research (F49620-01-1-0007, FA9550-04-1-0110).

Hudry, 2001) 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 often require particular problem instance characteristics to be exploited (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 numerous results in the literature on the asymptotic performance of simulated annealing. For example, 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. Cohn and Fielding (1999) provide an analysis of fixed temperature simulated annealing. Theodosopoulus (1999) discusses general acceptance probabilities and shows that using random restarts can improve convergence rates. For extensive reviews of simulated annealing convergence results, see Aarts and Korst (2002) and Henderson et al. (2003).

The current literature focuses mainly on asymptotic convergence properties. This paper presents necessary and sufficient (asymptotic) convergence conditions for GHC algorithms, including simulated annealing as a special case. In addition, new performance measures are introduced and used to evaluate and compare GHC algorithms with random restart local search. Such a comparison provides insights into both the asymptotic and the finite-time performance of discrete optimization metaheuristics. Therefore, the main contribution of this paper is the formulation of a new performance measure for GHC algorithms, the global visit probability, which is used to:

- develop necessary and sufficient convergence (in probability) conditions for such algorithms,
- compare these convergence conditions to convergence conditions for simulated annealing, in terms of the rate at which the cooling schedule approaches zero,
- show how both convergent and nonconvergent generalized hill climbing algorithms can be compared to random restart local search.

These results are illustrated by considering convergent cooling schedules for simulated annealing algorithms, and showing that the necessary and sufficient convergence conditions on the cooling schedules are equivalent to the necessary and sufficient conditions presented in this paper, when the generalized hill climbing algorithm is simulated annealing. Moreover, the global visit probability is used to show that the asymptotic relative performance of simulated annealing algorithms and random restart local search depends on the form of the cooling schedule and the neighborhood function (hence the resulting problem instance solution landscape). In particular, the analysis suggests that random restart local search can outperform simulated annealing given a sufficiently large number of restarts executed; the primary value of using simulated annealing may therefore be for finite-time executions that obtain near-optimal solutions reasonably quickly.

The paper is organized as follows: In Section 2, the GHC algorithm framework is presented, together with relevant concepts from discrete optimization. In Section 3, the finite global visit probability is introduced as a performance measure for GHC algorithms. In Section 4, this probability is used to define necessary and sufficient convergence conditions for GHC algorithms. These conditions are shown to be equivalent to the conditions for simulated annealing reported in Hajek (1988). Section 5 shows how the finite global visit probability can be used to compare and contrast convergent and nonconvergent GHC algorithms with random restart local search. These results identify a solution landscape parameter based on the basins of attraction for local optima that determines whether simulated annealing or random restart local search is more effective in visiting a global optimum. Section 6 summarizes the results presented.

2. Generalized Hill Climbing algorithms for discrete optimization problems

To describe GHC algorithms, several definitions are needed. For a discrete (minimization) optimization problem, define the *solution space*, Ω, as a finite set of all possible solutions. Define an *objective function f*: $\Omega \rightarrow [0, +\infty)$ that assigns a non-negative value to each element of the solution space. Assume that $|f(\omega)| < +\infty$ for all $\omega \in \Omega$. Two important components of GHC algorithms are the *neighborhood function*, $\eta : \Omega \rightarrow 2^{\Omega}$, where $\eta(\omega) \subseteq \Omega$ for all $\omega \in \Omega$, and the (independent) *hill climbing random variables* R_k: $\Omega \times \Omega \rightarrow \Re$, $k = 1, 2, \ldots$, where $\Re$ is the set of real numbers. 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), hence a solution landscape for the problem instance solution space. The neighborhood function is assumed to be symmetric (i.e., if $\omega' \in \eta(\omega'')$, then $\omega'' \in \eta(\omega')$ for all $\omega', \omega'' \in \Omega$) 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 $\omega \in \Omega$ 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 hill climbing random variables determine whether a randomly generated neighboring solution is accepted during a particular inner loop iteration associated with outer loop iteration k. 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. Although the range of the hill climbing random variables can be the set of reals, in practice they are typically restricted to the set of non-negative reals (which is what will be assumed for the rest of the paper). Therefore, for minimization problems, a neighboring solution is accepted (hence becomes the new current solution) if its objective function value is not significantly larger than the current solution's objective function value (as measured by the value generated for the hill climbing random variable). The GHC algorithm is described in pseudo-code form:

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

Assume that the hill climbing random variables have finite means and finite variances (i.e., $\mathrm{E}[|R_k(\omega(i), \omega)|] < +\infty$ 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) are *reachable* (i.e., for all $\omega', \omega'' \in \Omega$, there exists a sequence 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 how to avoid 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, $G = \{\omega^* \in \Omega : f(\omega^*) \leq f(\omega) \text{ for all } \omega \in \Omega\}$
- a set of local optima that are not global optima, $L \equiv L(\eta) = \{\omega \in \Omega \backslash G : f(\omega) \leq f(\omega') \text{ for all } \omega' \in \eta(\omega)\}$
- a set of non-optimal (i.e., locally improvable) 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 $G \cap L = \emptyset$, $G \cap H = \emptyset$, and $L \cap H = \emptyset$. Note also that for all $\omega \in G$, $\eta(\omega) \cap L = \emptyset$, and for all $\omega \in L$, $\eta(\omega) \cap G = \emptyset$ (i.e., a global optimum and a local optimum cannot be neighbors).

Several common local search algorithms can be described within the GHC algorithm framework. For example, a form of simulated annealing can be described as a GHC algorithm by setting $R_k(\omega(i), \omega) = -t(k)\ln(\nu_i), \omega(i) \in \Omega, \omega \in \eta(\omega(i)), k = 1, 2, \ldots,$ where $t(k)$ is the temperature parameter (hence, defines a cooling schedule as $t(k) \to 0$) and $\{\nu_i\}$ are independent and identically distributed $U(0, 1)$ random variables. 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 (Tovey, 1983), the noising method (Charon and Hudry, 2001), and Weibull accepting (see Jacobson et al., 1998; Johnson and Jacobson; 2002a, b for a discussion on how these algorithms can be fit into the GHC algorithm framework).

The iterations of a GHC algorithm can be classified based on the solutions being visited at each iteration. A *macro iteration* is a set of iterations that moves the algorithm from any element of $G \cup L$ to any element of $G \cup L$ (including itself), passing only through elements of H. Without loss of generality, assume that the GHC algorithm run is initialized at a solution in L (i.e., $\omega(0) \in L$), 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).

To illustrate the macro iteration concepts, in the GHC pseudo-code, the outer loops can be designed to correspond to macro iterations provided inner loop stopping criterion, *STOP INNER*, is true when the current solution is a local or a global optimum. Therefore, all the solutions visited during the inner loop iterations associated with macro iteration k will be in H, where the number of iterations between macro iterations is a random variable.

3. A performance measure for Generalized Hill Climbing algorithms

This section introduces the finite global visit probability as a performance measure for GHC algorithms. To describe this measure, consider a GHC algorithm applied to an instance of a discrete optimization problem, where $R_k(\omega(i), \omega) \geq 0, \omega(i) \in \Omega, \omega \in \eta(\omega(i))$, for all macro iterations k. At macro iteration k, define the event

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

and the complementary event

$$B^c(k) \equiv \{\text{The algorithm visits at least one element of } G \text{ over the first } k \text{ iterations}\}, \tag{2}$$

where $P\{B^c(k)\}$ is called the *finite global visit probability*.

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),

$$P\{B(k)\} \to P\{B\} = \bigcap_{k=1}^{+\infty} B(k) \text{ as } k \to +\infty,$$

where $P\{B\}$ is the *global visit probability*. Note that Hoos (1999) also looks at determining whether the global visit probability is zero or greater than zero for stochastic algorithms applied to a given class of problems. In particular, Hoos (1999) defines algorithms to be *probabilistic approximate complete* for a class of problems when $P\{B\} = 0$, while algorithms for which $P\{B\} > 0$ for a given class of problems are said to be *essentially incomplete* for this class. Hoos (1999) and Hoos and Stutzle (2000) use these concepts to study and analyze the performance of stochastic algorithms for SATISFIABILITY.

After k macro iterations, a GHC algorithm yields 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 $\omega^k \in \operatorname{argmin}\{f(\omega_j), j = 1, 2, \ldots, k\}$). Therefore, ω^k is the best solution reported to date. The key issue is determining whether the event $C(k) \equiv \{\omega_k \in G\}$ occurs, by using the performance measure for the solutions obtained over the first k macro iterations, $P\{\omega^k \in G\} = P\{B^c(k)\}$. See Propositions 1 and 2 in the Appendix for results that relate convergence in probability to G (i.e., $P\{C(k)\} \to 1$ as $k \to +\infty$), almost sure convergence to G (i.e., $P\{\liminf_k C(k)\} = 1$), and visits G infinitely often (i.e., $P\{\limsup_k C(k)\} = 1$).

4. Convergence conditions for Generalized Hill Climbing algorithms

4.1. *Necessary and sufficient convergence conditions*

This section uses the finite global visit probability to derive necessary and sufficient convergence conditions for GHC algorithms. Recall that $P\{B(0)\} = 1$ (i.e., all GHC algorithm runs are initialized at an element of L, hence $\omega(0) \in L$). Furthermore, unless otherwise stated, assume that $P\{B^c(k)\} < 1$ for all macro iterations k (which will hold for most GHC algorithms, and when it does not, a global optimum can be found in finite time, hence convergence is not an issue). For macro iteration k, define the *one-step macro iteration transition probability*

$$r(k) \equiv P\{B^c(k) \mid B(k-1)\} = P\{C(k) \mid B(k-1)\}. \tag{3}$$

This probability is needed to obtain the necessary and sufficient convergence conditions. First, Lemma 1 expresses the relationship between (3) and (1).

Lemma 1. *Consider a GHC algorithm. Then*

(i) $$P\{B(k)\} = \prod_{j=1}^{k}(1 - r(j)) \text{ for all macro iterations } k. \tag{4}$$

(ii) $$P\{B\} = \prod_{j=1}^{+\infty}(1 - r(j)). \tag{5}$$

Proof: See Jacobson and Yücesan (2004). □

Theorem 1 provides necessary and sufficient convergence conditions (in probability) for a GHC algorithm.

Theorem 1. *Consider a GHC algorithm. Then the algorithm converges in probability to G if and only if the following two conditions are satisfied*:
(i) $\sum_{k=1}^{+\infty} r(k) = +\infty$, *and*
(ii) $P\{C^c(k) \mid B^c(k-1)\} \to 0$ *as* $k \to +\infty$.

Proof: See the Appendix. □

Theorem 1 provides necessary and sufficient conditions for a GHC algorithm to converge in probability to G, as the number of macro iterations approaches infinity. Condition (i) requires that $r(k)$ not converge to zero too quickly as k approaches infinity. This means that the conditional probability that a GHC algorithm visits an element of G for the first time at macro iteration k approaches zero sufficiently slowly such that the infinite summation diverges. Condition (ii) requires that the conditional probability that a GHC algorithm visits an element of G beyond the first visit approaches one as the number of macro iterations approaches infinity.

4.2. Application to simulated annealing

The conditions in Theorem 1 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 G if and only if

$$\sum_{k=1}^{+\infty} e^{-(d^*/t(k))} = +\infty, \tag{6}$$

where $t(k)$ is a nonincreasing cooling schedule at iteration k (that approaches zero as $k \to +\infty$), and d^* is the maximum depth of all elements in L (i.e., the maximum gap in objective function value between an element of L and the solutions in H that can reach an element of G via deterministic 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 (with probability one) from it (note that this assumption implies that global optimality can be verified for an NP-hard problem, which is of itself NP-complete for many such problems; see Armstrong and Jacobson, 2003). Therefore, under this assumption, condition (ii) in Theorem 1 is always satisfied for simulated annealing. Moreover, since the neighborhood function η is defined such that the solution space is reachable, then at each macro iteration k 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 probability $r(k)$ has a component that includes the probability of escaping from the deepest local optimum. Therefore, using the law of total probability,

$$\begin{aligned} r(k) = \sum_{\omega \in L} & P\{B^c(k) \mid B(k-1) \cap \{\omega \in L \text{ is visited at macro iteration } k-1\}\} \\ & * P\{\{\omega \in L \text{ is visited at macro iteration } k-1\} | B(k-1)\} \end{aligned} \tag{7}$$

Since the number of local optima must be finite, the objective function is finite, and the hill climbing random variable at macro iteration k is exponential with mean $1/t(k)$, then $r(k)$ can be bounded above and below by a linear function of $P\{$Moving from the deepest element of L to an element of $G\} = P\{$Accepting hill climbing moves out of the deepest element of L to an element of $G\} = \Theta(\mathrm{e}^{-(d^*/t(k))})$ (see Cormen et al., 1997 for a formal definition of this notation). Therefore, a sufficient condition for condition (i) in Theorem 1 is (6).

To establish that (6) is a necessary condition for condition (i) in Theorem 1, using the same analysis as described above, at each macro iteration k sufficiently large, the conditional probability $r(k)$ has a component (see (7)) that includes the probability of escaping from each element of L. Therefore for each element of L, there exists a component for $r(k)$ that is Θ $(e^{-(d(\omega)/t(k))})$, $\omega \in L$, where $d(\omega)$ is the depth of ω, with $d^* = \max_{\omega \in L} d(\omega)$. Then if condition (i) in Theorem 1 holds, the infinite summation over each of these components must be unbounded, or else with positive probability, the algorithm visits and gets trapped at an element of L, hence is unable to move from this element of L to an element of G. This means that for all $\omega \in L$, $\sum_{k=1}^{+\infty} P\{B^c(k) \mid B(k-1) \cap \{\omega \in L$ is visited at macro iteration $k-1\}\} = +\infty$, which establishes that $\sum_{k=1}^{+\infty} e^{-(d^*/t(k))} = +\infty$ is a necessary condition for condition (i) in Theorem 1.

5. Using the finite global visit probability to compare the performance of GHC algorithms

5.1. *Random restart local search and GHC algorithms*

This section presents results that demonstrate how the finite global visit probability can be used to compare different GHC algorithms using the performance of random restart local search as a benchmark.

Random restart local search is executed by randomly selecting an initial solution (i.e., uniformly generated over the solution space) and by applying deterministic local search (i.e., iteratively moving to the best neighbor of the current solution) until a local optimum is found. This process is repeated until k local optima are obtained. The best of these k local optima is then reported. Therefore, each restart corresponds to a single macro iteration. Let *LS* denote a single macro iteration (restart) of random restart local search, where the neighborhood function η is defined such that the solution space is reachable. Using the GHC algorithm framework described in Section 2, the hill climbing random variables $R_k(\omega(i), \omega) = 0$ for all $\omega(i) \in \Omega$, $\omega \in \eta(\omega(i))$ and the best neighboring solution is selected at each inner loop iteration. Once an element of $G \cup L$ is found, a new element of Ω is randomly (uniformly) generated to begin the next (inner loop) set of iterations.

Other authors have compared simulated annealing with random restart local search. Ferreira and Zerovnik (1993) develop bounds on the probability that simulated annealing obtains an optimal (or near-optimal) solution. They also show that random restart local search dominates simulated annealing, measured by the probability of visiting a global optimum, as the number of restarts grows. Fox (1994) notes that this result is only true if both the number of accepted and *rejected* moves are counted. He also provides a clever example to illustrate this point, and notes that comparing random restart local search and simulating annealing may not be prudent. Fox (1993, 1995) presents modifications of simulated annealing that circumvent this counting issue, hence yielding more effective simulated annealing algorithm implementations. Garnier and Kallel (2002) look at ways to estimate the number of local optima, as well as the size and distribution of the attraction basins for such optima over a particular solution landscape. Such results provide insights into the effectiveness of restart procedures. Hart (1999) provides sequential stopping rules for stochastic search algorithms (including random restart local search) for continuous problems.

To show how random restart local search can be used to compare different GHC algorithms, at each macro iteration (i.e., at each restart), define the conditional probability

$$p(\omega) \equiv P\{LS \text{ terminates in } G \mid LS \text{ is initialized at } \omega \in \Omega\}. \tag{8}$$

Using the law of total probability, define

$$P\{G(\eta)\} \equiv P\{LS \text{ terminates in } G\} = \sum_{\omega\in\Omega} p(\omega) * P\{LS \text{ is initialized at } \omega \in \Omega\} \tag{9}$$

and

$$P\{L(\eta)\} \equiv P\{LS \text{ terminates in } L\} = 1 - P\{G(\eta)\}, \tag{10}$$

where $P\{G(\eta)\}$ and $P\{L(\eta)\}$ are functions of the solution landscape, hence the neighborhood function and the neighborhood probability mass function. For a uniform neighborhood probability mass function, $P\{G(\eta)\} = (|G| + \sum_{\omega\in H} p(\omega))/|\Omega|$ and $P\{L(\eta)\} = (|L| + \sum_{\omega\in H}(1 - p(\omega)))/|\Omega|$. By definition, $P\{L(\eta)\}$ measures the size of the combined basins of attraction of all local optima, L (i.e., the size of the set of solutions in the solution space for which deterministic local search terminates at a local optimum that is not a global optimum). Therefore, for a uniform neighborhood probability mass function, if η_1 and η_2 are two neighborhood functions defined on a solution space Ω where $\eta_1(\omega) \subseteq \eta_2(\omega)$ for all $\omega \in \Omega$, then $L(\eta_1) \supseteq L(\eta_2)$ and $P\{L(\eta_1)\} \leq P\{L(\eta_2)\}$. In general, enriching the neighborhood function such that local optima are eliminated will decrease $P\{L(\eta)\}$, hence increase $P\{G(\eta)\}$.

Without loss of generality, assume that $P\{L(\eta)\} > 0$, since if $P\{L(\eta)\} = 0$, then $P\{G(\eta)\} = 1$, hence deterministic local search will always find a global optimum with every restart (i.e., $P\{B^c(1)\} = 1$). Under this assumption, Theorem 2 provides a convergence comparison between random restart local search and a GHC algorithm that does not visit G in probability (i.e., $P\{B^c\} < 1$).

Theorem 2. *Let A be a GHC algorithm that does not visit G in probability. Let RR be a random restart local search algorithm. If the neighborhood function and the neighborhood probability mass function are defined on the solution space such that* $0 < P\{L(\eta)\}$, *then there exists a macro iteration* k_0 *such that for all* $k \geq k_0$,

$$P\{(B_{RR}(k))^c\} \geq P\{(B_A(k))^c\},$$

where k corresponds to both the number of macro iterations for A and the number of restarts for RR.

Moreover, if $\alpha = P\{B_A\} = \prod_{j=1}^{+\infty}(1 - r(j))$, then $k_0 \leq \ln(\alpha)/\ln(P\{L(\eta)\})$.

Proof: See the Appendix. □

Theorem 2 shows that if algorithm A does not visit G in probability (hence does not converge to G), then there exists a macro iteration beyond which random restart local search yields better results, as measured by the finite global visit probabilities at macro iteration k, $P\{(B_A(k))^c\}$ and $P\{(B_{RR}(k))^c\}$. Note that Theorem 2 applies to implementations of GHC algorithms that have a positive probability of getting trapped at a local optimum, such as some forms of simulated annealing with a geometric cooling schedule and most common implementations of threshold accepting. Therefore, given a sufficiently large number of restarts, random restart local search will dominate such algorithms, as measured by the finite global visit probability.

Given a GHC algorithm A, Theorem 3 provides sufficient conditions on the rate at which $P\{B_A(k)\}$ converges to zero such that the performance of random restart local search and algorithm A can be compared. To describe these conditions, define the non-negative value $\varphi_0 = -\ln(P\{L(\eta)\})$.

Theorem 3. *Let A be a GHC algorithm that visits G in probability. Let RR be a random restart local search algorithm. If the neighborhood function and the neighborhood probability mass function are defined on the solution space such that* $0 < P\{L(\eta)\}$*, and k simultaneously corresponds to both the number of macro iterations for A and the number of restarts for RR, then*

(i) *if* $P\{B_A(k)\} = O(e^{-\varphi k})$ *for k large and* $\varphi \geq \varphi_0$*, there exists a macro iteration* k_0 *such that for all* $k \geq k_0$, $P\{(B_{RR}(k))^c\} \leq P\{(B_A(k))^c\}$,

(ii) *if* $P\{B_A(k)\} = O(e^{-\varphi k})$ *for k large and* $\varphi < \varphi_0$*, there exists a macro iteration* k_0 *such that for all* $k \geq k_0$, $P\{(B_{RR}(k))^c\} \geq P\{(B_A(k))^c\}$,

(iii) *if* $P\{B_A(k)\} = o(e^{-\varphi k})$ *for k large and* $\varphi \geq \varphi_0$*, there exists a macro iteration* k_0 *such that for all* $k \geq k_0$, $P\{(B_{RR}(k))^c\} \leq P\{(B_A(k))^c\}$,

(iv) *if* $1/P\{B_A(k)\} = o(e^{\varphi k})$ *for k large and* $\varphi \leq \varphi_0$*, there exists a macro iteration* k_0 *such that for all* $k \geq k_0$, $P\{(B_{RR}(k))^c\} \geq P\{(B_A(k))^c\}$,

where $P\{B_A(k)\} = O(e^{-\varphi k})$ *for k large means that* $P\{B_A(k)\}\,e^{\varphi k}$ *is bounded above by a constant as* $k \to +\infty$, $P\{B_A(k)\} = o(e^{-\varphi k})$ *for k large means that* $P\{B_A(k)\}e^{\varphi k} \to 0$ *as* $k \to +\infty$*, and* $1/P\{B_A(k)\} = o(e^{\varphi k})$ *for k large means that* $1/(P\{B_A(k)\}e^{\varphi k}) \to 0$ *as* $k \to +\infty$.

Proof: See the Appendix. □

Theorem 3 compares the performance of random restart local search and a GHC algorithm that visits G in probability for four different cases. The remaining two cases (i.e., $P\{B_A(k)\} = o(e^{-\varphi k})$ for k large with $\varphi < \varphi_0$ and $1/P\{B_A(k)\} = o(e^{\varphi k})$ for k large with $\varphi > \varphi_0$) are inconclusive, hence the performance of a GHC algorithm becomes problem instance specific, where general results cannot be obtained based on the approach used here. Recall from Theorem 1 that if a GHC algorithm visits G in probability, then it may or may not converge (since condition (i)) is a necessary and sufficient condition for visiting G in probability, but only a necessary condition for convergence). Therefore, GHC algorithms that visit G in probability include all convergent (in probability) GHC algorithms and some GHC algorithms that are not convergent. In addition, note that an identical analysis can be used to show that Monte Carlo search yields the same conclusions obtained in both Theorems 2 and 3, with the new definition $P\{L(\eta)\} \equiv 1 - (|G\,|/|\Omega\,|)$. However, $P\{L(\eta)\}$ for Monte Carlo search will be greater than or equal to $P\{L(\eta)\}$ for random restart local search. This means that there exist GHC algorithms for which cases (ii) and (iv) in Theorem 3 are satisfied for random restart local search, but are not satisfied for Monte Carlo search, while there are no GHC algorithms that visit G in probability for which the reverse is true. Therefore, random restart local search dominates Monte Carlo search when comparing their performance to a GHC algorithm, as measured by the finite global visit probability.

The results in Theorem 3 suggest that $P\{L(\eta)\}$ determines the relative performances of random restart local search and GHC algorithms that visit G in probability. In general, $1-\delta \leq P\{L(\eta)\} < 1$ for some $\delta > 0$ close to zero, which is a function of the GHC algorithm being applied. Therefore, $\varphi_0 = -\ln(P\{L(\eta)\}) \geq -\ln(1-\delta) \geq \delta$. Moreover, the closer $P\{L(\eta)\}$ is to one, the larger the number of restarts needed for random restart local search to dominate a GHC algorithm that visits G in probability (see the proof of Theorem 3). For practical purposes, this suggests that for solution spaces (and associated neighborhood functions) with many local optima, it may be more effective to use a GHC algorithm that visits G in probability. Therefore, the design and structure of the neighborhood function (hence the number and distribution of local optima in the solution space) is a key factor in determining whether random restart local search performs better than a GHC algorithm that visits G in probability.

One limitation of the results in Theorems 2 and 3 is that they do not take into account the number of iterations between macro iterations. For random restart local search, this represents the number of iterations needed to reach a local optimum from each randomly generated initial solution, while for a GHC algorithm, this represents the number of iterations between visits to local optima. Note that as a GHC algorithm A executes, and the hill climbing random variables approach the value zero with probability one, the number of iterations between the macro iterations may be very small, as the algorithm gets trapped in the same local optimum with increasing probability. If this is the case, then for k sufficiently large, $P\{(B_A(k))^c\}$ will be (and stay) close to zero for all future macro iterations. Fox (1993, 1995) notes this point for simulated annealing, and suggests alternative ways to improve the performance of simulated annealing to overcome this situation.

5.2. Application to simulated annealing

Using the necessary and sufficient convergence condition for simulated annealing in Hajek (1988), recall from Section 4.2 that $r(k)$ can be bounded above and below by functions that are $\Theta(e^{-(d*/t(k))})$ for k sufficiently large. Therefore, there exists constants $\gamma_1 > 0$ and $\gamma_2 > 0$ and macro iteration k_0 such that $\gamma_1 e^{-(d^*/t(k))} \leq r(k) \leq \gamma_2 e^{-(d^*/t(k))}$ for all $k \geq k_0$. Therefore, for all $k \geq k_0$,

$$\prod_{j=1}^{k} \left(1 - \gamma_2 e^{-(d^*/t(j))}\right) \leq P\{B_{SA}(k)\} = \prod_{j=1}^{k}(1 - r(j)) \leq \prod_{j=1}^{k} \left(1 - \gamma_1 e^{-(d^*/t(j))}\right). \quad (11)$$

Hajek's cooling schedule condition, $\sum_{k=1}^{+\infty} e^{-(d^*/t(k))} = +\infty$, places restrictions on the rate at which the cooling schedule $t(k)$ approaches zero. Consider cooling schedules that are defined such that $e^{-(d^*/t(k))} = \lambda(1/k)^\delta$ for $k \geq 2$ and $\lambda \in Z^+$, for some $0 < \delta \leq 1$. Note that $t(k)$ could also be defined using iterated logarithms (e.g., $e^{-(d^*/t(k))} = \lambda(1/k(\ln(k))))$ or any other form provided that Hajek's condition on the cooling schedules are satisfied.

From (11), an upper and a lower bound on $e^{\varphi k} P\{B_{SA}(k)\}$, as $k \to +\infty$, can be obtained. In particular, for all $k \geq k_0$,

$$e^{\varphi k} \prod_{j=1}^{k} \left(1 - \gamma_2 e^{-(d^*/t(j))}\right) \leq e^{\varphi k} P\{B_{SA}(k)\} \leq e^{\varphi k} \prod_{j=1}^{k} \left(1 - \gamma_1 e^{-(d^*/t(j))}\right)$$

which leads to

$$e^{\varphi k} \prod_{j=1}^{k}(1 - \gamma_2 \lambda(1/j)^\delta) \leq e^{\varphi k} P\{B_{SA}(k)\} \leq e^{\varphi k} \prod_{j=1}^{k}(1 - \gamma_1 \lambda(1/j)^\delta). \quad (12)$$

Since $\prod_{j=1}^{+\infty}(1 - \gamma_1 \lambda(1/j)^\delta) = 0$ if and only if $\sum_{j=1}^{+\infty}(1/j)^\delta = +\infty$, then the rate at which $\prod_{j=1}^{k}(1 - \gamma_1 \lambda(1/j)^\delta)$ approaches zero as $k \to +\infty$ relative to the rate at which $e^{\varphi k}$ approaches infinity determines which of the four cases described in Theorem 3 apply to this simulated annealing algorithm.

To determine this rate, note that for some $k_0 \in Z^+$ where $\gamma_2 \lambda(1/k_0)^\delta < 1$,

$$\ln\left(\prod_{j=k_0}^{k} \left(1 - \gamma_1 \lambda(1/j)^\delta\right)\right) = \sum_{j=k_0}^{k} \ln\left(1 - \gamma_1 \lambda(1/j)^\delta\right) \leq \sum_{j=k_0}^{k} -\gamma_1 \lambda(1/j)^\delta. \quad (13)$$

The integral approximation for the right hand side of (13) implies that the left hand side of (13) is $O(-\ln(k))$ as $k \to +\infty$ for $\delta = 1$, and $O(-k^{1-\delta})$ as $k \to +\infty$ for $0 < \delta < 1$. Therefore, taking the exponential function in (13), $\prod_{j=1}^{k}(1 - \gamma_1 \lambda(1/j)^\delta)$ is $O(1/k)$ as $k \to +\infty$ for $\delta = 1$, and $O(\exp(k^{\delta-1}))$ as $k \to +\infty$ for $0 < \delta < 1$. Therefore, $e^{\varphi k} \prod_{j=1}^{k}(1 - \gamma_1 \lambda(1/j)^\delta) \to +\infty$ as $k \to +\infty$ for $0 < \delta \leq 1$. The same conclusions are obtained from the lower bound in (12). In particular,

$$\ln\left(\prod_{j=k_0}^{k}(1-\gamma_2\lambda(1/k)^{\delta})\right) = \sum_{j=k_0}^{k}\ln(1-\gamma_2\lambda(1/j)^{\delta})$$
$$\geq \sum_{j=k_0}^{k} -\gamma_2\lambda(1/j)^{\delta}/(1-\gamma_2\lambda(1/k)^{\delta}). \tag{14}$$

The integral approximation for the right hand side of (14) leads to the same results. Therefore, for $0 < \delta \leq 1, e^{\varphi k}\prod_{j=1}^{k}(1-\gamma_2\lambda(1/j)^{\delta}) \to +\infty$ as $k \to +\infty$. This means that cases (i), (ii), and (iii) in Theorem 3 cannot occur for this convergent simulated annealing algorithm. Therefore, either cases (iv) occurs (provided $\varphi \leq \varphi_0$), hence random restart local search dominates this simulated annealing algorithm (as measured by the finite global visit probability), or the results are inconclusive (if $\varphi > \varphi_0$). Note that if φ_0 is very close to zero, hence $P\{L(\eta)\}$ is very close to one, then from the proof of Theorem 3, if case (iv) occurs, the value for k_0 may be prohibitively large, by comparing $e^{\varphi k}$ with functions that are $O(1/k)$ or $O(\exp(k^{\delta-1}))$. Therefore, the form of the cooling schedule for simulated annealing and the choice of neighborhood function that defines the value for $P\{L(\eta)\}$ for random restart local search determine the relative performance of these two algorithms.

5.3. *Illustrative example*

The results in Section 5.1 provide insights into why random restart local search may or may not perform better than a GHC algorithm. Ferreira and Zerovnik (1993) show that random restart local search always dominates simulated annealing in its ability to visit a global optimum. Fox (1994) comments that this result is only true if both the number of accepted and rejected moves are counted, and illustrates this point using a clever example. This section revisits this example in Fox (1994), and shows how the results in Section 5.1 apply to it.

Consider a solution space of size $n+2$, $\Omega = \{-1, 0, 1, 2, \ldots, n\}$, with a neighborhood function η defined as $\eta(j) = \{j-1\}$, $j = 3, 4, \ldots, n$, $\eta(2) = \{-1, 1\}$, $\eta(1) = \{0\}$, $\eta(0) = \{2\}$, and $\eta(-1) = \{2\}$, and with a uniform neighborhood probability mass function. Define the objective function on this solution space as $f(\omega) = |\omega|$ for all $\omega \in \Omega$. Therefore, $G = \{0\}$ and $L = \{-1\}$.

For random restart local search, if all the restart solutions are generated uniformly over the entire solution space, then $P\{L(\eta)\} = (n+1)/(2(n+2)) < 1/2$. Moreover, for a GHC algorithm A with the probability of accepting a hill climbing solution set to one (such as simulated annealing with infinite temperature), then $r(k) = 1/2$. Therefore, from the proof of Theorem 3,

$$P\{(B_{RR}(k))^{c}\} \geq P\{(B_A(k))^{c}\} \Leftrightarrow P\{L(\eta)\}^{k} \leq P\{B_A(k)\}$$
$$\Leftrightarrow ((n+1)/(2(n+2)))^{k} \leq (1/2)^{k}. \tag{15}$$

This expression shows that random restart local search dominates this GHC algorithm (as measured by the finite global visit probability) after any number of restarts. However, this result does not take into account the number of iterations between each restart for random restart local search, and the number of iterations between visits to elements in $G \cup L$ for the GHC algorithm. In particular, for random restart local search, the average number of iterations between restarts is $(n^2 + 2n + 2)/(n + 2)$, while for the GHC algorithm, the average number of iterations between visits to elements in $G \cup L$ from the element in L is 5/2. Therefore, for all $n \geq 3$, for (15) to hold, the expected total number of iterations for random restart local search exceeds the expected total number of iterations for this GHC algorithm. Note that since the elements $\{3, 4, \ldots, n\}$ in the solution space do not serve any meaningful purpose except (in this situation) to provide additional solutions for random restart local search to be initialized at with each restart, hence could be compressed into the single solution $\{2\}$, then the expected number of total iterations between restarts would be 5/2, the same as for the GHC algorithm. Therefore, random restart local search would then dominate this GHC algorithm (as measured by the finite global visit probability). This example illustrates both the limitations and value of using the number of restarts as the only measure when comparing random restart local search with GHC algorithms.

6. Summary

This paper presents necessary and sufficient convergence conditions for GHC algorithms. Moreover, the finite global visit probability is used to show how different GHC algorithms (both convergent and nonconvergent) can be compared using random restart local search. These results are also compared to the necessary and sufficient convergence conditions for simulation annealing. In particular, the necessary and sufficient convergence conditions for GHC algorithms presented here are related to the necessary and sufficient convergence conditions in Hajek (1988) when the GHC algorithm is simulated annealing.

The results presented here are consistent with the simulated annealing results presented in Ferreira and Zerovnik (1993), where Theorem 3 provides a comparison between random restart local search and a class of algorithms that includes simulated annealing as a special case. Moreover, the key factors that determine whether random restart local search or simulated annealing is asymptotically better are the rate at which cooling schedule approaches zero for simulated annealing and the value for $P\{L(\eta)\}$ for random restart local search (hence, the choice of neighborhood function and neighborhood probability mass function). The interplay between these factors is captured by the four cases described in Theorem 3. Moreover, the results presented in Fox (1993) complement the results described here. In particular, the relationship between $P\{L(\eta)\}$ (which is determined by the neighborhood function and the neighborhood probability mass function) and the rate at which $P\{B(k)\}$ approaches zero for simulated annealing (which is determined by the cooling schedule and the rate at which it approaches zero) demonstrate that the design components that are within control of the user of these two algorithms determine their relative effectiveness. Moreover, the results presented in this paper support the statement in Fox (1993) that a random restart simulated annealing algorithm may actually provide the best results, by exploiting the naive restart feature of random restart local search with the intelligent search mechanism embodied in simulated annealing. Fox (1999) also demonstrates how the two approaches can profitably team up.

The majority of theoretical results on simulated annealing are concerned with its asymptotic convergence. The results in Section 4.2 and Theorems 2 and 3 suggest that random restart local search can outperform simulated annealing given a sufficiently large number of restarts executed. The primary value of using simulated annealing may therefore be for finite-time executions that obtain near-optimal solutions reasonably quickly. This, in turn, suggests that one should focus on the finite-time behavior of simulated annealing rather than the asymptotic convergence results that dominate the literature. These results also suggest that the value of random restart local search may only become apparent for sufficiently high run-times. Work is in progress to use and extend these results to identify both convergent and nonconvergent GHC algorithm formulations that perform well over finite horizons, as well as determine the number of random restarts that are needed to satisfy the inequalities in Theorems 2 and 3. Moreover, work is in progress to determine how the framework provided by the macro iteration structure can be further exploited to gain insights into the finite-time and asymptotic performance of GHC algorithms in general.

Appendix

Proposition 1 establishes the standard relationship between convergence in probability to G, almost sure convergence to G, and visits to G infinitely often (see Billingsley 1979 for formal definitions).

Proposition 1. *Consider a GHC algorithm.*

(i) *If this GHC algorithm converges almost surely to G* (*as the number of macro iterations approaches infinity*), *then the algorithm converges in probability to G.*

(ii) *If this GHC algorithm converges in probability to G*, *then the algorithm visits G infinitely often.*

Proof: For (i), if the GHC algorithm converges almost surely to G, then $P\{\lim\ \inf_k C(k)\} = 1$. However, since $P\{\lim\ \inf_k C(k)\} \leq (\lim_k P\{C(k)\}$, then the GHC algorithm converges in probability to G. For (ii), if the GHC algorithm converges in probability to G, then $\lim_k P\{C(k)\} = 1$. However, since $\lim_k P\{C(k)\} \leq (P\{\lim\ \sup_k C(k)\}$, then $P\{\lim\ sup_k\ C(k)\} = 1$, hence the GHC algorithm visits G infinitely often. □

Proposition 2 provides a condition on the hill climbing random variables for a GHC algorithm such that almost sure convergence to G is equivalent to visits G infinitely often.

Proposition 2. *Consider a GHC algorithm such that there exists an outer loop iteration k_0 such that $R_k(\omega, \omega') = 0$ with probability one for all $\omega \in \Omega, \omega' \in \eta(\omega), k \geq k_0$. Then this GHC algorithm converges almost surely to G* (*as the number of outer loop iterations approaches infinity*) *if and only if it visits G infinitely often.*

Proof: Suppose that a GHC algorithm visits G infinitely often. If its hill climbing random variables are defined such that there exists an outer loop iteration k_0 such that $R_k(\omega, \omega') = 0$

with probability one for all $\omega \in \Omega$, $\omega' \in \eta(\omega)$, $k \geq k_0$, then there exists an iteration $K(k_0)$ such that $\omega^* \equiv \omega(i) \in G \cup L$ for all $i \geq K(k_0)$. Therefore, the GHC algorithm visits all solutions finitely many times except solution ω^*, which is visited infinitely often. Moreover, since $\omega^* \equiv \omega(i) \in G \cup L$ for all $i \geq K(k_0)$, then ω^* is visited almost always. However, since the GHC algorithm visits G infinitely often, then $\omega^* \in G$, hence the algorithm converges almost surely to G. The converse is obtained from Proposition 1. □

From Proposition 2, if the hill climbing random variables for a GHC algorithm are defined such that they become zero with probability one beyond a certain outer loop iteration number, then convergence almost surely to G and visits G infinitely often are equivalent. This type of GHC algorithm is commonly used in practice, where the algorithm is designed to be deterministic local search after a *finite* (though possibly large) number of iterations. Note that at present, since GHC convergence results are all asymptotic (e.g., Johnson and Jacobson, 2002a, b; Sullivan and Jacobson, 2001), this result has limited practical value. On the other hand, since a finite length convergence result would establish the equivalence of convergence almost surely and convergence in probability, this observation provides strong evidence that the formulation of such a finite length convergence result is highly unlikely.

Proof of Theorem 1: By the law of total probability,

$$\begin{aligned} P\{C^c(k)\} &= P\{C^c(k) \mid B^c(k-1)\}P\{B^c(k-1)\} + P\{C^c(k) \mid B(k-1)\}P\{B(k-1)\} \\ &= P\{C^c(k) \mid B^c(k-1)\}P\{B^c(k-1)\} + P\{B(k)\} \end{aligned}$$

($\Leftarrow$) Suppose that conditions (i) and (ii) hold. First, note that $\sum_{k=1}^{+\infty} r(k) = +\infty$ if and only if $P\{B^c(k)\} = 1$. To see this, the following equivalent relations will be shown: There exists a real value $\alpha > 0$ such that

$$\sum_{j=1}^{+\infty} r(j) < +\infty \quad \text{if and only if} \quad P\{B(k)\} = \prod_{j=1}^{+\infty}(1 - r(j)) \geq \alpha.$$

To see this, since $\ln(1-x) \leq -x$ for all $0 < x < 1$, then $\ln(\prod_{j=1}^{+\infty}(1 - r(j))) = \sum_{j=1}^{+\infty} \ln(1 - r(j)) \leq \sum_{j=1}^{+\infty} -r(j)$ or equivalently, $\prod_{j=1}^{+\infty}(1 - r(j)) \leq \exp\{-\sum_{j=1}^{+\infty} r(j)\}$. Therefore, let $\prod_{j=1}^{+\infty}(1 - r(j)) \geq \alpha > 0$ and suppose that $\sum_{j=1}^{+\infty} r(j) = +\infty$. Then from $\prod_{j=1}^{+\infty}(1 - r(j)) \leq \exp\{-\sum_{j=1}^{+\infty} r(j)\}$, $\prod_{j=1}^{+\infty}(1 - r(j)) = 0$, which is a contradiction. Conversely, let $\sum_{j=1}^{+\infty} r(j) < +\infty$ and suppose that $\prod_{j=1}^{+\infty}(1 - r(j)) = 0$. Since $\prod_{j=1}^{+\infty}(1 - r(j)) = 0$, then $\ln(\prod_{j=1}^{k}(1 - r(j))) = \sum_{j=1}^{k} \ln(1 - r(j))$ diverges to $-\infty$ (as $k \to +\infty$), or equivalently, $\sum_{j=1}^{k} \ln(1/(1 - r(j)))$ diverges to $+\infty$. However, since $\ln(1+x) \leq x$ for all $x \geq 0$, then $\sum_{j=1}^{k} \ln(1/(1 - r(j))) = \sum_{j=1}^{k} \ln(1 + r(j) + r^2(j) + \cdots) \leq \sum_{j=1}^{k}(r(j) + r^2(j) + \cdots) = \sum_{j=1}^{k} r(j)/(1 - r(j))$, which must diverge. Finally, since $\sum_{j=1}^{+\infty} r(j) < +\infty$, then for all $\varepsilon > 0$, then there exists a non-negative integer $j(\varepsilon)$ such that $r(j) \leq \varepsilon$ for all $j \geq j(\varepsilon)$. Therefore, $r(j)/(1 - r(j)) \leq r(j)/(1 - \varepsilon)$ for all $j \geq j(\varepsilon)$, which implies that $\sum_{j=j(\varepsilon)}^{+\infty} r(j)/(1 - r(j)) \leq \sum_{j=j(\varepsilon)}^{+\infty} r(j)/(1 - \varepsilon)$. However, if the right hand side of this inequality is finite, then the left hand side must be finite, which contradicts that the left hand side diverges.

Given this relation, then from conditions (i) and (ii), as well as (6), $P\{C^c(k)\} \to 0$ as $k \to +\infty$.

($\Rightarrow$) Suppose that $P\{C(k)\} \to 1$ as $k \to +\infty$. Then this implies that $P\{B^c(k)\} \to 1$ as $k \to +\infty$, which is true if and only if $\sum_{k=1}^{+\infty} r(k) = +\infty$, hence condition (i) must hold. Moreover, since $P\{C(k)\} \to 1$ as $k \to +\infty$, then condition (ii) must also hold. □

Proof of Theorem 2: By definition, $P\{(B_{RR}(k))^c\} = 1 - (P\{L(\eta)\})^k$ for all k. Moreover, from Lemma 1

$$P\{(B_A(k))^c\} = 1 - \prod_{j=1}^{k}(1 - r(j)).$$

In addition, since the GHC algorithm does not visit G in probability, hence $P\{(B_A)^c\} < 1$, then from Theorem 1, $\sum_{j=1}^{+\infty} r(j) < +\infty$. Since $\sum_{j=1}^{+\infty} r(j) < +\infty$, then from (4), there exists a positive integer k_0 and real values α, α', $0 < \alpha < \alpha' \leq 1$, such that $\alpha \leq P\{B_A(k)\} = \prod_{j=1}^{k}(1 - r(j)) \leq \alpha'$ for all $k \geq k_0$. Also, by definition, since $P\{L(\eta)\} < 1$, then from (5), there exists a macro iteration k_0 such that

$$P\{L(\eta)\} \leq \left(\prod_{j=1}^{k}(1 - r(j))\right)^{1/k} \qquad \text{for all} \quad k \geq k_0.$$

Therefore, for all macro iterations $k \geq k_0$,

$$\begin{aligned} & P\{L(\eta)\} \leq \left(\prod_{j=1}^{k}(1 - r(j))\right)^{1/k} \\ \Leftrightarrow \quad & (P\{L(\eta)\})^k \leq \prod_{j=1}^{k}(1 - r(j)) \\ \Leftrightarrow \quad & 1 - (P\{L(\eta)\})^k \geq 1 - \prod_{j=1}^{k}(1 - r(j)) \\ \Leftrightarrow \quad & P\{(B_{RR}(k))^c\} \geq P\{(B_A(k))^c\}. \end{aligned}$$

To show the second part of the theorem, when this relationship holds at equality, then

$$\begin{aligned} \Leftrightarrow \quad & P\{L(\eta)\} = \left(\prod_{j=1}^{k}(1 - r(j))\right)^{1/k} \\ \Leftrightarrow \quad & \ln(P\{L(\eta)\}) = (1/k)\ln\left(\prod_{j=1}^{k}(1 - r(j))\right). \end{aligned} \tag{16}$$

Recall that since $P\{B_A\} > 0$, then $\prod_{j=1}^{+\infty}(1 - r(j)) \geq \alpha$. Substituting this inequality into (16) leads to

$$\ln(P\{L(\eta)\}) \geq (1/k)\ln(\alpha),$$

which implies that $k_0 \leq \ln(\alpha)/\ln(P\{L(\eta)\})$. Therefore, $\ln(\alpha)/\ln(P\{L(\eta)\})$ is an upper bound on the number of restarts k_0 such that random restart local search dominates, as measured by the finite global visit probability, a GHC algorithm A that does not visit G in probability. □

Proof of Theorem 3: To show case (i), from the proof of Theorem 2, $P\{(B_{RR}(k))^c\} \leq P\{B_A(k)^c\}$ if and only if $P\{L(\eta)\}\} \geq (P\{B_A(k)\})^{1/k}$. Therefore, if $P\{B_A(k)\} = O(e^{-\varphi k})$ for k large with $\varphi \geq \varphi_{.0}$, then there exists a macro iteration k_0 such that for all $k \geq k_0$, $P\{L(\eta)\} \geq (P\{B_A(k)\})^{1/k}$. To see this, by definition, $P\{B_A(k)\} = O(e^{-\varphi k})$ for k large means that $P\{B_A(k)\}e^{\varphi k}$ is bounded above by a constant as $k \to +\infty$. Therefore, at macro iteration k,

$$\begin{aligned} & P\{(B_{RR}(k))^c\} \leq P\{B_A(k)^c\} \\ \Leftrightarrow \quad & 1 - (P\{L(\eta)\})^k \leq P\{B_A(k)^c\} \\ \Leftrightarrow \quad & (P\{L(\eta)\})^k \geq P\{B_A(k)\} \\ \Leftrightarrow \quad & (P\{L(\eta)\})^k e^{\varphi k} \geq P\{B_A(k)\}e^{\varphi k} \end{aligned} \tag{17}$$

Letting $k \to +\infty$, the right hand side of (17) is bounded above by a constant. Since $\varphi_0 = -\ln(P\{L(\eta)\})$, then $e^{-\varphi_0} = P\{L(\eta)\}$. Substituting this equation into the left hand side of (17) yields $e^{(\varphi-\varphi_0)k}$. Therefore, since $\varphi \geq \varphi_0$, then the inequality in (17) holds as $k \to +\infty$ (hence for k sufficiently large). A similar argument establishes (ii), (iii), and (iv). □

Acknowledgment

The authors wish to thank two anonymous referees for their feedback and comments on an earlier version of the paper. Their suggestions have lead to a significantly improved final version of the manuscript.

References

Aarts, E. and J. Korst. (2002). "Selected Topics in Simulated Annealing." In P. Hansen and C.C. Ribeiro (Eds.), *Essays and Surveys on Metaheuristics*, Chapter 1, Norwell, MA, Kluwer Academic Publishers, pp. 1–37.

Armstrong, D.E. and S.H. Jacobson. (2003). "Studying the Complexity of Global Verification for NP-hard Discrete Optimization Problems." *Journal of Global Optimization* 27(1), 83–96.

Billingsley, P. (1979). *Probability and Measure*, New York, John Wiley and Sons.

Charon, I. and O. Hudry. (2001). "The Noising Method: A Generalization of Some Metaheuristics." *European Journal of Operational Research* 135(1), 86–101.

Cohn, H. and M. Fielding. (1999). "Simulated Annealing: Searching for an Optimal Temperature Schedule." *SIAM Journal On Optimization* 9(3), 779–802.

Cormen, T.H. C.E. Leiserson, and R.L. Rivest. (1997). *Introduction to Algorithms*, New York: McGraw-Hill.

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.

Ferreira, A.G. and J. Zerovnik. (1993). "Bounding the Probability of Success of Stochastic Methods for Global Optimization." *Computers and Mathematics Applications* 25(10/11) 1–8.

Fox, B.L. (1993). "Integrating and Accelerating Tabu Search, Simulated Annealing, and Genetic Algorithms." *Annals of Operations Research* 41, 47–67.

Fox, B.L. (1994). "Random Restarting versus Simulated Annealing." *Computers and Mathematics Applications* 27, 33–35.

Fox, B.L. (1995). "Faster Simulated Annealing." *SIAM Journal of Optimization* 5, 488–505.

Fox, B.L. (1999). *Strategies for Quasi-Monte Carlo*. Norwell, MA: Kluwer Academic Publishing.

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

Garnier, J. and L. Kallel. (2002). "Efficiency of Local Search with Multiple Local Optima." *SIAM Journal of Discrete Mathematics* 15(1), 122–141.

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

Hajek, B. (1988). "Cooling Schedules for Optimal Annealing." *Mathematics of Operations Research* 13, 311–329.

Hart, W.E. (1999). "Sequential Stopping Rules for Random Optimization Methods with Applications to Multistart Local Search." *SIAM Journal on Optimization* 9(1), 270–290.

Henderson, D, S.H. Jacobson, and A.W. Johnson. (2003). "The Theory and Practice of Simulated Annealing." In F. Glover and G. Kochenberger (Eds.), *Handbook on Metaheuristics*, Chapter 10, Norwell, MA: Kluwer Academic Publishers, pp. 287–319.

Hoos, H.H. (1999). "On the Run-Time Behavior of Stochastic Local Search Algorithms for SAT." *Proceedings of the AAAI-99* MIT Press, pp. 661–666.

Hoos, H.H. and T. Stutzle. (2000). "Local Search Algorithms for SAT: An Empirical Evaluation." *Journal of Automated Reasoning* 24, 421–481.

Jacobson, S.H. and K.A. Sullivan, and A.W. Johnson. (1998). "Discrete Manufacturing Process Design Optimization Using Computer Simulation and Generalized Hill Climbing Algorithms." *Engineering Optimization* 31, 247–260.

Jacobson, S.H. and D. Solow. (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.

Jacobson, S.H. E. Yücesan. (2004). "Global Optimization Performance Measures for Generalized Hill Climbing Algorithms." *Journal of Global Optimization* 29(2), 177–193.

Johnson, A.W. and S.H. Jacobson. (2002a). "A Convergence Result for a Class of Generalized Hill Climbing Algorithms." *Applied Mathematics and Computation* 125(2/3), 359–373.

Johnson, A.W. and S.H. Jacobson. (2002b). "On the Convergence of Generalized Hill Climbing Algorithms." *Discrete Applied Mathematics* 119(1/2), 37–57.

Johnson, D.S., C.H. Papadimitriou and M. Yannakakis. (1988). "How Easy is Local Search?" *Journal of Computers and Systems Science* 37(1), 79–100.

Kirkpatrick, S., C.D. Gelatt Jr., and M.P. Vecchi. (1983). "Optimizaton by Simulated Annealing." *Science* 220, 671–680.

Lin, S. and B.W. Kernighan. (1973). "An Effective Heuristic for the Traveling Salesman Problem." *Operations Research* 21, 498–516.

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.

Sullivan, K.A. and S.H. Jacobson. (2001). "A Convergence Analysis of Generalized Hill Climbing Algorithms." *IEEE Transactions on Automatic Control* 46(8), 1288–1293.

Theodosopoulus, T.V. (1999). "Some Remarks on the Optimal Level of Randomization in Global Optimization." In P. Pardalos, S. Rajasekaran, and J. Rolim (Eds.), *Randomization Methods in Algorithm Design* in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, pp. 303–318.

Tovey, C.A. (1983). "On the Number of Iterations of Local Improvement Algorithms." *Operations Research Letters* 2, 231–238.