Generalized Hill Climbing Algorithms: Theory and Practice*

by

Sheldon H. Jacobson
Simulation and optimization Laboratory
Department of Mechanical and Industrial Engineering
University of Illinois
Urbana, Illinois 61801-2906

shj@uiuc.edu
(217) 244-7275 (office)
(217) 244-6534 (fax)
netfiles.uiuc.edu/shj/www/shj.html

* This work has been supported in part by the Air Force Office of Scientific Research (FA9550-04-1-0110).

Background and Motivation

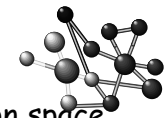
- Many discrete optimization problems (DOPs) can be challenging (NP-hard).
  - Traveling Salesman Problem
  - Satisfiability
  - Scheduling Problems
- Researchers have been forced to develop problem specific heuristics or apply general search strategies.

Tabu search (memory)
Genetic algorithms (mutation) } Local Search Algorithms
Simulated annealing (metals)

Local Search Strategies

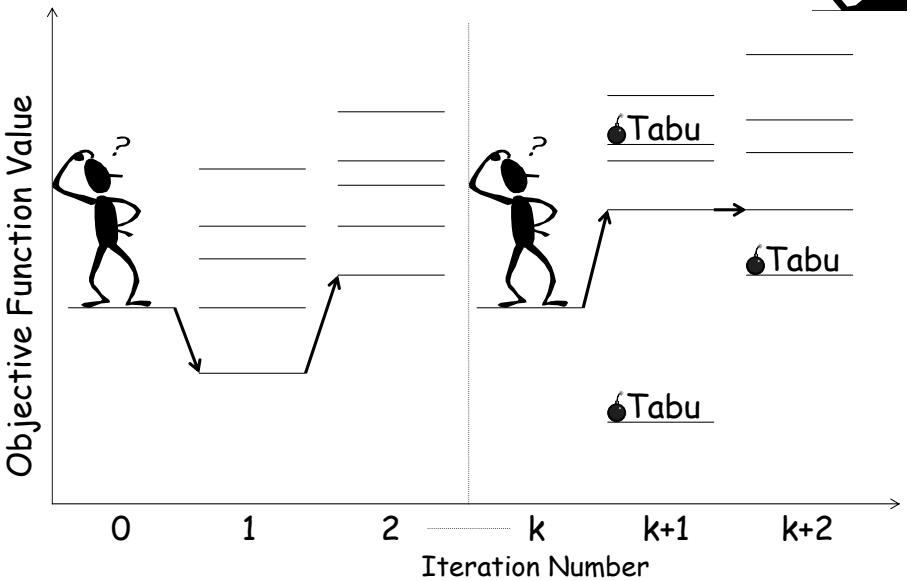
* Tabu Search (Glover)
  - Uses memory to guide local search towards global optima

* Genetic Algorithms
  - Uses mutation to search the solution space

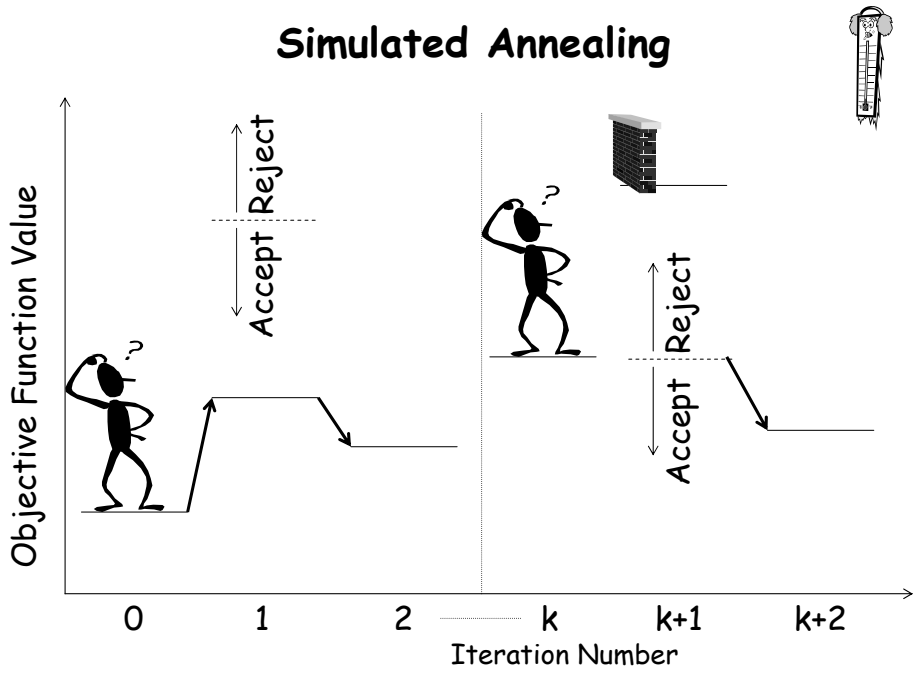


* Simulated Annealing
  - Mimics the physical annealing process of metals (heating and cooling) to move towards global optima

Tabu Search



Simulated Annealing



Does there exist a well-defined framework under which all these local search algorithms can be modeled and analyzed?

Is there a way to work *in between* local search algorithms like simulated annealing, tabu search, and genetic algorithms?

Generalized Hill Climbing Algorithms

Definitions

A discrete optimization (minimization) problem can be defined by

* a set of discrete objects, called the *solution space*, $\Omega = \{\omega_1,\omega_2,\ldots,\omega_N\}$.

* an *objective function,* $f:\Omega\rightarrow[0,+\infty)$.

Definitions (cont'd)

To apply a local search algorithm, define

* a *neighborhood function,* $\eta: \Omega \rightarrow 2^{\Omega}$, where $\eta(\omega) \subset \Omega$ for all $\omega \in \Omega$

* each neighborhood function creates a set of local optima, $L_\eta = \{\omega'' \in \Omega: f(\omega'') \leq f(\omega)$ for all $\omega \in \eta(\omega)\}$

For the traveling salesman problems, with N cities and symmetric cost matrix $C \in R^{N \times N}$,

* Ω is the set of all possible permutations ((N-1)!/2) of the N cities.

* **f** is the cost of traveling through the N cities, for each possible permutation of cities, where the cost of traveling between any two cities is given by the cost matrix C.

* η is a neighborhood function that allows movement between the elements of Ω (e.g., 2-opt, 3-opt).

Generalized Hill Climbing Algorithms

Define a neighborhood function η
Define a set of hill climbing random variables R_k: $\Omega \times \Omega \rightarrow \Re \cup \{-\infty, +\infty\}$
Set the iteration indices $i = 0$, $k = 1$
Select an initial solution $\omega(0) \in \Omega$
Set the best-to-date solution $\omega^* \leftarrow \omega(0)$
Repeat
  Repeat
    Generate a neighboring solution $\omega \in \eta(\omega(i))$
    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 $\omega(i+1) \leftarrow \omega(i)$
    Set $i \leftarrow i+1$.
    If $f(\omega(i)) < f(\omega^*)$, then $\omega^* \leftarrow \omega(i)$
  Until *STOP INNER*
  $k \leftarrow k+1$
Until *STOP OUTER*
Report ω^*

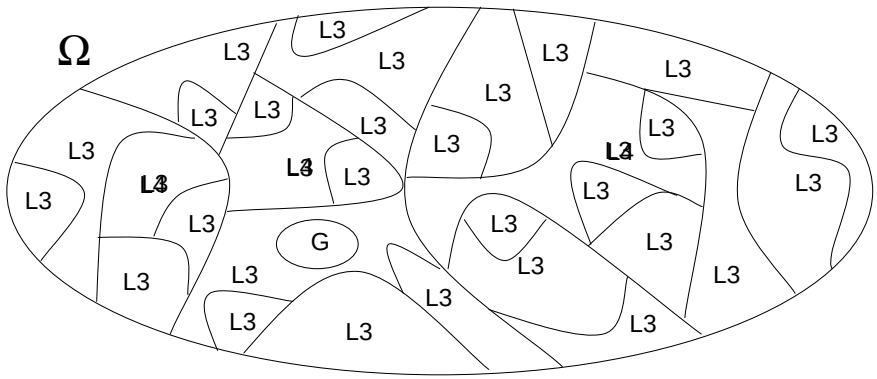
Key Features of GHC Algorithms

- Choice of hill climbing random variables R_k
  - Easy to generate
  - Bounded versus unbounded range
- Choice of neighborhood function η
  - Reachability/Accessibility
  - Size and computational cost of generating neighbors
  - Number of local optima (local basins of attraction)

Local Basins of Attraction

$L\lambda$ = Local Optimum for λ-sized neighborhoods
G = Global Optimum

$L3 \supseteq L4 \supseteq G$



Particular GHC Algorithms

* $R_k = Q_k$ (Threshold Accepting)

* $R_k = -t_k/ln(U)$, where U~U(0,1) (Simulated Annealing)

* $R_k = +\infty$, where $\eta(\omega) = \Omega$ (Pure Monte Carlo Search)

* $R_k = 0$ (Pure Local Search)

* $R_k = \lceil ln(1-U)/ ln(1-P_k) \rceil$, where U~U(0,1)
(Geometric Accepting)

Particular GHC Algorithms (cont'd)

* $R_k = e_k + [-2ln(U1)]^{1/2} \sin(2\pi U2)\, v_k$, where U1, U2 ~ U(0,1)
(Normal Accepting)

* $R_k = t_k(-ln(U))^{1/\alpha}$, where U~U(0,1), $\alpha > 0$
(Weibull Accepting)

* $R_k = -t_k[ln(U_1)+ln(U_2)]$, where U_1, U_2 ~U(0,1)
(Erlang Accepting)

• $R_k(\omega(k),\omega') = I\{\omega' \notin L\}/[1-I\{\omega' \notin L\}]-I\{\omega' \in L\}/[1-I\{\omega' \in L\}]$

where L is a tabu list of solutions (Tabu Search)

Ordinal Hill Climbing (OHC)

* Combines features of ordinal optimization and generalized hill climbing algorithms.

* Iteratively moves between *sets* of solutions

* <u>OO Component</u>: Uses ordinal information to rank the sets of solutions at each iteration

<u>GHC Component</u>: Controls the number of solutions that remain in the ranked set from iteration to iteration

Ordinal Hill Climbing (OHC) Algorithm

Select a set of M initial solutions $D(0) \subset \Omega$
Set the iteration number k = 0
Define the hill climbing variable R_k, where
i) $\Sigma_{m=1,2,\ldots,M} P\{R_k = m\} = 1$, *ii*) $R_0 = 1$ with probability one.
Repeat
Order the solutions in $D(k)$ from smallest to largest objective function values
Generate R_k (= r_k)
Keep the best (smallest objective function value) r_k solutions from the set $D(k)$. Call this set E_1.
Generate M-R_k new designs from the solution space Ω. Call this set E_2.
Set $D(k+1) \leftarrow E_1 \cup E_2$.
$k \leftarrow k+1$
Until stopping criterion is met

OHC Algorithm Parameters

* M, the size of the selected solution sets, *D(k)*.

* *D*(0), the set of initial solutions.

* Hill climbing variables R_k, where

i) $\Sigma_{m=1,2,\ldots,M} P\{R_k = m\} = 1$,

ii) $R_0 = 1$ with probability one.

Also, $R_k \rightarrow M$ w.p.1 as $k \rightarrow +\infty$.

* Generating $M\text{-}R_k$ new solutions at each iteration.

Genetic Algorithms and OHC Bridges

GA: Uses the concept of genetics and evolution to guide the search towards good solutions

GA Terminology:

parents = solutions at current solution

offspring = possible solutions at next iteration (next parents)

mating = move from one iteration to the next (create offspring of parents)

fitness = objective function value

Genetic Algorithm Components

* Reproduction
  - process by which individual parents are evaluated for mating and inclusion in future populations.

* Crossover
  - process by which offspring are generated based upon the objective function values of the parents.

* Mutation
  - involves the random alteration of a parameter value in the offspring

Variations of GA that are OHC Algorithms

"Simple" GA, with reproduction and crossover components.

"Simple" GA with a mutation component.

Open Question: Can sophisticated GA implementations be modeled as OHC algorithms?

GHC Algorithms Implementation Issues

- $R_k \rightarrow_P 0$ as $k \rightarrow +\infty$ (Approaches pure local search)
- Rate at which $R_k \rightarrow_P 0$ can impact convergence (Hajek 1988)
  - If $R_k \rightarrow_P 0$ too quickly, the algorithm can get "stuck" in an undesirable local optimum.
  - If $R_k \rightarrow_P 0$ too slowly, the algorithm may never visit any local optimum or not enough local optima.
- For simulated annealing, $R_k = - t_k / ln(U)$, where $U \sim U(0,1)$ and t_k is the temperature parameter:

$t_k = \alpha\ t_{k-1}$, $0 < \alpha < 1$

$t_k = -\kappa / ln(1+k)$, $\kappa > 0$

$t_k = t_{k-1} / (1 + \kappa t_{k-1})$, $\kappa > 0$

Convergence Results

* Mostly for simulated annealing
* Model algorithm execution using Markov chain
* Homogeneous versus inhomogeneous approaches
  - inner versus outer loop
* Rely on exponential acceptance function and accessibility of solution spaces

Global Optimization Performance Measures

Motivation

- Traditional performance measures (such as asymptotic convergence) do not capture what is done in practice
- Convergence criteria are rarely what a practitioner needs.

Definitions

- G = Set of globally optimal solutions
- $L \equiv L\eta$ = Set of locally (not globally) optimal solutions
- $H = \Omega \setminus (L \cup G)$ = Set of solutions that are not optima
- *Macro iterations* count the number of visits to $L \cup G$.

Discrete-Time Markov Chain Model

For k fixed, the $L \cup G$ solution transitions can be modeled as homogenous discrete-time Markov chain with macro iterations transition matrix

$$P_M^k = \begin{bmatrix} P_{GH}^k [\sum_{j=0}^{+\infty} (P_{HH}^k)^j] P_{HG}^k + P_{GG}^k & P_{GH}^k [\sum_{j=0}^{+\infty} (P_{HH}^k)^j] P_{HL}^k + P_{GL}^k \\ P_{LH}^k [\sum_{j=0}^{+\infty} (P_{HH}^k)^j] P_{HG}^k + P_{LG}^k & P_{LH}^k [\sum_{j=0}^{+\infty} (P_{HH}^k)^j] P_{HL}^k + P_{LL}^k \end{bmatrix}$$

GHC Algorithm Performance Measures

$P\{B(k)\} \equiv P\{$Algorithm *does not* visit G over the first k macro iterations$\}$

Measures the probability of *not* having visited a global optimum over the first k visits to $L \cup G$.

$r(k) \equiv P\{B^c(k) \mid B(k-1)\}$

One (macro) step transition probability

$P\{B^c \mid B(k)\}$ where $B = \bigcap_{k=1}^{+\infty} B(k)$

False negative probability

Performance Measure Properties

* $B(k) \supset B(k+1)$ (Telescoping events)

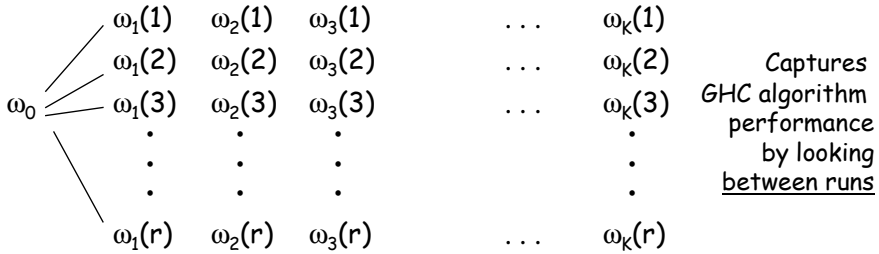
* $P\{B(K)\} = \prod_{k=1}^{K} [1-r(k)]$

* $P\{B\} = \prod_{k=1}^{+\infty} [1-r(k)] = P\{\bigcap_{k=1}^{+\infty} B(k)\}$

* $P\{B^c|B(k)\} \geq P\{B^c|B(k+1)\}$ for all $k \in Z^+$

  (False negative probability is non-increasing in k)

Sequences of Solutions Generated by a GHC Algorithm



$\omega_k(j) \in L \cup G$ k = 1, 2, …, k, j = 1,2,…,r (large)

K = Number of macro iterations

r = Number of possible sequences of solutions that can be generated by a GHC algorithm $\leq |L \cup G|^K$ (Exponential Explosion!)

Applications of Performance Measures

Convergence Performance Measures

* A GHC algorithm *converges in probability* to G if $P\{C(k)\} \to 1$ as $k \to +\infty$, where

  $P\{C(k)\} = P\{\omega_k(.) \in G\}$

  = P{Algorithm is in G at macro iteration k}

Practical Performance Measures

* A GHC algorithm *visits G in probability* if
  - $P\{B^c \mid B(k)\} = 1$ for all macro iterations k
  - $P\{B^c(k)\} \to 1$ as $k \to +\infty$.

Results

1) If a GHC algorithm converges in probability to *G*, then the algorithm visits *G* in probability.

2) A GHC algorithm **visits *G* in** probability iff $\sum_{k=1}^{+\infty} r(k) = +\infty$.

> •If *{r(k)}* approach zero too quickly, then the GHC algorithm will not converge in probability to *G*. This observation is used to prove that Threshold Accepting does not converge (**Open question resolved**!)

Results (N&S Convergence Conditions)

A GHC algorithm converges in probability to *G* iff

i) $\Sigma_{k=1,2,\ldots} r(k) = +\infty$

ii) $P\{C^c(k) \mid B^c(k-1)\} \rightarrow 0$ as $k \rightarrow +\infty$.

i): the algorithm must not approach pure local search too quickly
ii): the algorithm must be able to revisit *G* with increasing probability

> i): speed (rate) of how algorithm changes
> ii): direction of algorithm's change

Results (cont'd)

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

$$r(k) = \sum_{\omega_2 \in L} \sum_{\omega_1 \in G} q(\omega_2,k\text{-}1)\, P_{LH}^k\ (\omega_2,\bullet) \sum_{j=0}^{\infty} \left(P_{HH}^k\right)^j P_{HG}^k\ (\bullet, \omega_1),$$

where for all $\omega \in L$, at macro iteration k,
$Q(\omega,k) \equiv B(k) \cap$ {algorithm is at solution ω at macro iteration k}
with $q(\omega,k) = P\{Q(\omega,k)\}$.

> * This expression allows one to determine the dominant factor in an algorithm to determine whether $\sum_{k=1}^{+\infty} r(k) =$ or $< +\infty$.

Results (cont'd)

The most dominant term that determines whether a GHC algorithms visit *G* in probability is $O(P\{R_k(\omega_2,\omega) \geq \delta(\omega_2,\omega)\})$ for $\omega_2 \in L$, $\omega \in \eta(\omega_2) \cap H$ (as $R_k \rightarrow_P 0$ as $k \rightarrow +\infty$).

* If the hill climbing random variables are defined such that the probability of escaping 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, then the resulting GHC algorithm <u>will not converge</u> in probability to *G*.

Illustrative Examples

***Pure Monte Carlo Search** $\sum_{k=1}^{+\infty} r(k) = +\infty$, hence visits G in probability, but does not converge in probability to G.

***Random Restart Local Search**: $\sum_{k=1}^{+\infty} r(k) = +\infty$, hence visits G in probability, but does not converge in probability to G.

***Threshold Accepting**: $\sum_{k=1}^{+\infty} r(k) < +\infty$, hence does not visit G in probability, and does not converge in probability to G.

***Simulated Annealing**: $\sum_{k=1}^{+\infty} r(k) = +\infty$ translates into Hajek's (1988) convergence condition on the cooling schedule.

GHC versus Random Restart

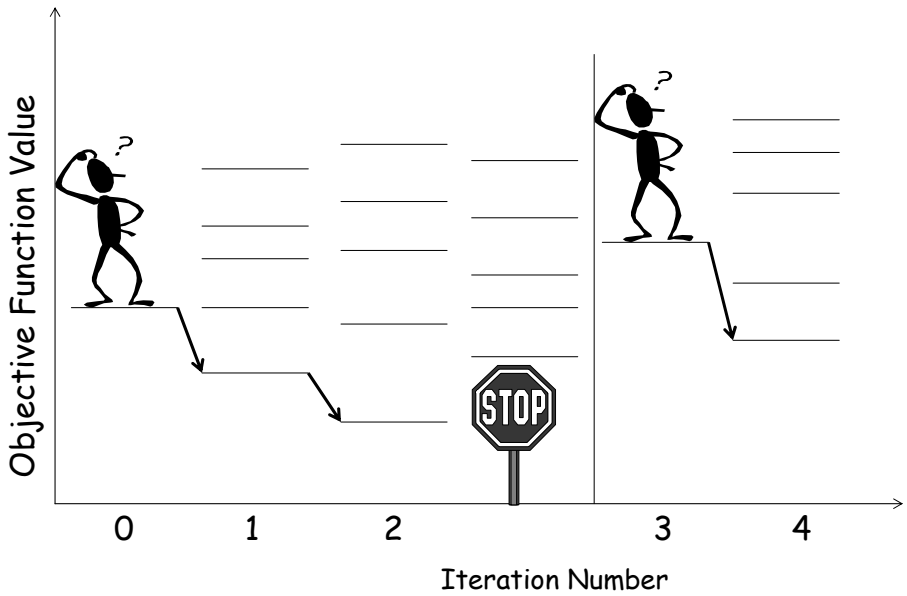
Anecdotal empirical evidence suggests that simply (randomly) restarting an algorithm yields better performance results than sophisticated modifications to GHC algorithms.

Can this observation be rigorously evaluated?

Random Restart (RR) Local Search

Set the iteration indices j,k
Set the number of restarts K (Macro Iterations)
Set k=1
Repeat
    Generate an initial solution $\omega(0) \in \Omega$
    Set j=1
    Repeat
        Generate a neighboring solution $\omega' \in \eta(\omega(j))$
        Compute $\delta = f(\omega') - f(\omega(j))$
        If $\delta \leq 0$, then $\omega(j+1) \leftarrow \omega'$
        If $\delta > 0$, then $\omega(j+1) \leftarrow \omega(j)$
        $j \leftarrow j+1$
    Until a local optima is reached
    Set $k \leftarrow k+1$
    Set ω^* to be the best solution to date
Until K restarts have been completed

Random Restart Local Search



Definitions

Let LS denote a single macro iteration (restart) of algorithm RR

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

$P\{G\text{-stop}\} = P\{\text{Algorithm LS terminates in } G\}$
$= \Sigma_{\omega \in \Omega}\ p(\omega) * P\{\text{LS is initialized at } \omega\}$
$= [\ |\ G\ |\ + \Sigma_{\omega \in \Omega \setminus G}\ p(\omega)\ ]\ /\ |\ \Omega\ | > 0.$

$P\{L\text{-stop}\} = P\{\text{LS terminates in } L\}$
$= [|\ L\ |\ + \Sigma_{\omega \in \Omega \setminus L}\ (1 - p(\omega))\ ]\ /\ |\ \Omega\ | > 0.$

* $P\{G\text{-stop}\} + P\{L\text{-stop}\} = 1$ (LS always terminates in $L \cup G$)
* $P\{L\text{-stop}\}$ is a function of the neighborhood function η
* Choice of neighborhood function impacts the effectiveness of RR

Result

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 is defined on the solution space such that $P\{L\text{-stop}\} < 1$, then there exists a macro iteration K_0 such that for all $K \geq K_0$,

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

Moreover, $K_0 \leq \ln[\alpha] / \ln[P\{L\text{-stop}\}]$, where $\Pi_{k=1,2,\ldots}\ (1 - r(k,G)) \geq \alpha > 0$

*** RR is more effective than a nonconvergent GHC algorithm, provided the number of restarts is sufficiently large***

Result

Let A be a GHC algorithm that visits G in probability. Let RR be a random restart local search algorithm. Let $\varphi_0 = -\ln(P\{L\text{-stop}\})$. Then

i) if $P\{B_A(K)\} = O(e^{-\varphi K})$ for K large and $\varphi \geq \varphi_0$, then 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 $\varphi < \varphi_0$, then there exists a macro iteration K_0 such that for all $K \geq K_0$, $P\{(B_{RR}(K))^c\} \geq P\{(B_A(K))^c\}$

iii) if $P\{B_A(K)\} = o(e^{-\varphi K})$ for K large and $\varphi \geq \varphi_0$, then 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 $\varphi \leq \varphi_0$, then there exists a macro iteration K_0 such that for all $K \geq K_0$, $P\{(B_{RR}(K))^c\} \geq P\{(B_A(K))^c\}$

RR may or may not be more effective than a convergent GHC algorithm. The key factor is the choice of neighborhood function, which impacts $\varphi_0 = -\ln(P\{L\text{-stop}\})$.

Note on Inconclusive Cases

$P\{B_A(K)\} = o(e^{-\varphi K})$ 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 cases, hence the performance of each of these GHC algorithms becomes problem-specific.

Value of Results

•Random restart local search is more/less effective than a GHC algorithm, based upon the rate at which $P\{B_A(K,G)\}$ approaches zero.

•Using Hajek's (1988) convergence result for simulated annealing, random restart local search often outperforms simulated annealing!

Implications of Results

* Random restart local search performance results

  - Visits G more effectively for nonconvergent and some convergent GHC algorithms
  - LS is typically analyzed in finite time, yet its value is asymptotic
  - SA is typically analyzed asymptotically, yet its value is over finite-time

Concluding Comments

•GHC Algorithms model many different local search algorithms

•Performance and convergence are distinct ways to measure the effectiveness of GHC algorithms

•Using this framework to better understand finite-time performance is an important next step

•As problems become larger and more complex, general purpose algorithms will need to be better fine-tuned for specific applications.

Publications in this Area

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

Fleischer, M., Jacobson, S.H., 1999, "Information Theory and the Finite-Time Behavior of the Simulated Annealing Algorithm: Experimental Results," *INFORMS Journal on Computing*, 11(1), 35-43.

Kumar, A., Jacobson, S.H., Sewell, E.C., 2000, "Computational Analysis of a Flexible Assembly System Design Problem," *European Journal of Operational Research*, 123(3), 453-472.

Sullivan, K.A. and Jacobson, S.H., 2000, "Ordinal Hill Climbing Algorithms for Discrete Manufacturing Process Design Optimization Problems," *Discrete Event Dynamical Systems*, 10(4), 307-324.

Vaughan, D., S.H. Jacobson, and D. Armstrong, 2000, "A New Neighborhood Function for Discrete Manufacturing Process Design Optimization Using Generalized Hill Climbing Algorithms," *ASME Journal of Mechanical Design*, 122(2), 164-171.

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

Publications in this Area (cont'd)

Johnson, A.W., Jacobson, S.H., 2002, "A Class of Convergent Generalized Hill Climbing Algorithms," *Applied Mathematics and Computation*, 125(2-3), 359-373.

Johnson, A.W., Jacobson, S.H., 2002, "On the Convergence of Generalized Hill Climbing Algorithms," D*iscrete Applied Mathematics*, 119(1-2), 37-57.

Orosz, J.E., Jacobson, S.H. 2002, "Finite-time Performance Analysis of Static Simulated Annealing Algorithms," *Computational Optimization and Applications*, 21(1), 21-53.

Orosz, J.E., Jacobson, S.H., 2002, "Analysis of Static Simulated Annealing Algorithms," *Journal of Optimization Theory and Application*, 115(1), 165-182.

Fleischer, M.A., Jacobson, S.H., 2002, "Scale Invariance Properties in the Simulated Annealing Algorithm," *Methodology and Computing In Applied Probability*, 4(3), 219-241.

Henderson, D., Vaughan, D.E., Jacobson, S.H., Wakefield, R.R., Sewell, E.C., 2003, "Solving the Shortest Route Cut and Fill Problem Using Simulated Annealing," *European Journal of Operational Research*, 145(1), 72-84.

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

Publications in this Area (cont'd)

Jacobson, S.H., Yucesan, E., 2004, "Global Optimization Performance Measures for Generalized Hill Climbing Algorithms," *Journal of Global Optimization*, 29(2), 173-190.

Armstrong, D.E., Jacobson, S.H., 2004, "Polynomial Transformations and Data Independent Neighborhood Functions," *Discrete Applied Mathematics*, 143(1-3), 272-284.

Vaughan, D.E., Jacobson, S.H., 2004, "Formulating the Meta-Heuristic Tabu Search in the Generalized Hill Climbing Framework," *Methodology and Computing in Applied Probability*, 6(3), 343-354.

Jacobson, S.H., Yucesan, E., 2004, "Analyzing the Performance of Generalized Hill Climbing Algorithms," *Journal of Heuristics*, 10(4), 387-405.

Vaughan, D.E., Jacobson, S.H., Hall, S.N., Mclay, L.A., 2005, "Simultaneous Generalized Hill Climbing Algorithms for Addressing Sets of Discrete Optimization Problems," *INFORMS Journal on Computing* (Accepted).

Armstrong, D.E., Jacobson, S.H., 2005, "Data Independent Neighborhood Functions and Strict Local Optima," *Discrete Applied Mathematics*, 146(3), 233-243.

Thank you for coming!

Comments and Questions?