International Journal of Computational and Numerical Analysis and Applications

Volume 1 No. 4 2002, 353-367

ESTIMATING THE CONVERGENCE RATE OF A RESTARTED SEARCH PROCESS

Andrea Hu[1], R. Shonkwiler[2], M.C. Spruill[3 §]
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332-0160, USA
[1]e-mail: andreahu@yahoo.com
[2]e-mail: shenk@math.gatech.edu
[3]e-mail: spruill@math.gatech.edu

Abstract: When a deterministic algorithm for finding the minimum of a function on a given set is employed, it may reach a local minimum and remain there forever after. Restarting repeatedly and independently by a random choice of a starting point when the algorithm reaches a settling point engenders a probability of λ^n/s of not having seen the goal state by the nth epoch. The rate λ may be expressed precisely, if only theoretically, as the solution to the equation $\lambda^{-1}\phi_{H|N}(\lambda^{-1}) = (1-\theta_0)^{-1}$, where $\phi_{H|N}$ is the probability generating function of the time to reach a settling point given a start leading to a non-global extremum. Here, θ_0 is the probability of a random start leading directly to the goal and in problems of interest will be quite small. A simple bound has $\lambda > \frac{(1-\theta_0)(1+E[H|N])}{\theta_0+(1-\theta_0)(1+E[H|N])}$ so that slow geometric rates when θ_0 is small are even slower when expected times between restarts in non-goal states are large. Nevertheless, geometric rates imply that independent identical

Received: December 3, 2001

[§]Correspondence author

parallel processing has potential benefits in speeding up acquisition of the goal states, and the expression of the precise rate in terms of the probability generating function provides a method of statistical real-time estimation of the resources required to obtain an answer by a prescribed time.

AMS Subject Classification: 65C05, 65K10, 49M05
Key Words: restarting deterministic algorithms, geometric convergence rate, parallel speedup

1. Introduction

In this paper, the statistical properties of the progress towards the minimum are studied for a class of stochastic algorithms. Results extend those of Shonkwiler and Van Vleck [4] and introduce on-the-fly estimation of the rate of convergence of the algorithm and prove its asymptotic normality.

Members of the class of stochastic algorithms treated here are generated by randomly restarting a deterministic iterative scheme when it reaches a settling point. Given a real valued function C defined on a topological domain Ω with neighborhood system N, we analyze restart methods for finding some point $x^* \in opt(C) = G$, the set of global minimizers of C over Ω, where $x^* \in G$ entails $C(x^*) \leq C(x), x \in \Omega$. If C is sufficiently smooth, there are powerful numerical methods for identifying local minimizers. Often these methods utilize local derivative, or gradient, information to construct a sequence of points $x_0, x_1, x_2, \cdots \to x_\infty$ for which x_∞ is a local minimum as determined to within some tolerance. It is assumed here that whatever iterative improvement scheme g is employed (more properly g_C) it satisfies the following properties for C fixed:

(A1) for each $x \in \Omega$ there is a smallest $k = k(x) < \infty$ such that $g^j(x) = g^k(x)$ for all $j \geq k$,

and

(A2) if $x = g(x)$, then there is a neighborhood $N_x \in N$ of x such that for no $y \in N_x$ do we have $C(y) < C(x)$.

The algorithm g generates a sequence of points x_0, $x_1 = g(x_0)$, $x_2 = g^2(x_0)$, $\ldots$ in Ω. Under the assumptions, the limit of such a sequence is a local minimizer, which could also be a global minimizer as well, beyond which progress toward the objective is not possible by the application of g alone. All points of the domain which are attracted to a given local minimizer define its basin. Furthermore, the relation $x \equiv y$ if and only if $\lim_{k\to\infty} g^k(x) = \lim_{j\to\infty} g^j(y)$ is an equivalence relation on Ω. The resulting equivalence classes, B_i, $i = 0, 1, \ldots$, are the *basins* of Ω relative to g_C. The *settling point* or *local minimizer* x_∞ of basin B is given as the limit, $\lim_{k\to\infty} g^k(x)$, where x is any point of B.

Once $g(x) = x$, the process must be *restarted* in order to have any chance generally of converging to a global minimizer. For this, we use a fixed probability distribution μ over Ω.

For a fixed function C on a fixed domain Ω and a fixed restarting measure μ there is an associated Ω-valued stochastic process, $X(t)$, which we call the search process and can be described as follows. Let $Y_1, \ldots Y_n, \ldots$ be independent and identically distributed Ω-valued random variables distributed according to μ. Then $X(1) = Y_1$ and generally $X(t+1) = g(X(t))$ unless these are the same, in which case $X(t+1)$ is the next in the sequence of Y's.

If M is the union of all the basins whose settling points are global minima, our primary interest is in $T = \min\{t : X(t) \in M\}$, the first hitting time of M by the search process. In our main result, we shall prove that if μ places positive mass at all points of the finite set Ω, then there are $\lambda \in (0,1)$, $\delta \in (0,\lambda)$, and a constant $s > 1$ such that for any $\epsilon > 0$, one has $(\delta+\epsilon)^{-n}|P[T > n] - \lambda^n/s| \to 0$ as $n \to \infty$. Furthermore λ is the unique positive solution to the equation $\lambda^{-1}\phi_{H|N}(\lambda^{-1}) = (1-\theta_0)^{-1}$, where $\phi_{H|N}$ is the probability generating function of the random time to first settling point given that the starting state is one which leads to a non-global extremum. Results on expectation of time to hit the goal, consequences of independent identical processing, and how to estimate the parameter λ on the fly are also presented.

The restarting of algorithms in the manner we have described generally goes under the name of *mutltistart* (see [3]). The main inspiration for our research on this topic is the work of Shonkwiler and Van Vleck [4] who studied restarting from the perspective of its consequences on first hitting times. Our results here augment theirs since we show that generally $s > 1$. Mendivil, Shonkwiler, and Spruill [2] extended their results to non-stationary and more general processes through the use of renewal equations. One of the topics not covered in those papers and treated here is the estimation, based on data accumulated through a run, of the algorithm's convergence rate.

In the next section it is shown that corresponding to the Markov chain X on Ω is a Markov chain X^* whose state space Ω^* is simple enough to allow us to compute Perron-Frobenius eigenvectors and has a subset M^* such that if $T^* = \min\{n : X_n^* \in M^*\}$, then for all n, $P[T > n] = P[T^* > n]$.

2. Notation and a Simple Canonical Representation

The set Ω is taken, without loss of generality, to be a finite set and is the disjoint union $\bigcup_{i=0}^{b} B_i$ of basins of attraction which partition the space. Let $M = \bigcup_{i=0}^{g} B_i, g < b$ denote the union of basins of attraction to global minima and, without loss of generality, assume $B_0 = M$. Define for each $i = 0, \ldots, b$, $\lambda_i = \max\{k(x) : x \in B_i\}$. Introduce the set Ω^*:

$$\Omega^* = \{(i,k) : i = 0,1,\ldots,b, k = 0,1,\ldots,\lambda_i\}.$$

Denote the transition matrix of the Markov chain $X(n)$ by $P(x,x') = \mu(x')$ if $g(x) = x$, $P(x,x') = 1$ if $x' = g(x) \neq x$, and $P(x,x') = 0$ otherwise. Let

$$A(i,k) = \{x \in B_i : k(x) = k\},$$

define for $(i,j) \in \Omega^*$

$$\mu^*((i,j)) = \sum_{x \in A(i,j)} \mu(x),$$

and the transition function Q on $\Omega^* \times \Omega^*$ by $Q((i,j),(i',j')) = 1$ if $i = i'$ and $j' = j - 1 \geq 0$, (in the same basin and g yields a new candidate point) $Q((i,j),(i',j')) = \mu^*((i',j'))$ if $j = 0$, (restart since we are at the bottom of a basin) and $Q((i,j),(i',j')) = 0$ otherwise.

Let $M^* = \{(i,j) \in \Omega^* : i = 0\}$. Since the B's form a partition, $X^*(t) = (i(X(t)), k(X(t)))$, where $i(X(t))$ is the basin in which $X(t)$ is located, is well defined. Clearly the distribution of T is the same as that of $\min\{t : i(X(t)) = 0\} = T^*$, and since the transition function Q is easily verified to be that of X^*, we can and shall work with the chain X^* to obtain information about the function $P[T > t]$.

3. Rate of Convergence of Tail Probabilities

It will be proven that the tail probabilities $P[T > n]$ tend to zero geometrically for restarted processes and the precise rate ascertained in this section.

Begin by simplifying notation. We shall assume that our process X is the canonical process of Section 2 and call the transition matrix P, with the state probability vector arranged as

$$p' = (p_{0,0}, p_{0,1}, \cdots p_{0,\lambda_0}, p_{1,0}, p_{1,1}, \cdots, p_{1,\lambda_1}, p_{2,0}, \cdots, p_{2,\lambda_2}, \cdots, p_{b,0}, \\ \cdots, p_{b,\lambda_b}),$$

so that the probability distribution on states at epoch t is $p_0'P^t$ for an initial distribution p_0.

Denote by $\tilde{P}$ the submatrix of the transition P obtained by deleting the first $\lambda_0 + 1$ rows and columns and correspondingly, the vector $\tilde{p}_0$ is formed by deleting the first $\lambda_0 + 1$ entries of the initial probability vector, p_0. Then starting according to the initial distribution p_0, the probability

$$P[T > n] = \tilde{p}_0' \tilde{P}^n 1,$$

where 1 is an appropriately sized vector of all 1's. From the Perron-Frobenius theory of positive matrices it is known that if $\lambda \in (0, 1)$ is the Perron-Frobenius eigenvalue (spectral radius) of $\tilde{P}$ with left eigenvector ω normalized so $\omega' 1 = 1$, right eigenvector χ normalized so $\omega' \chi = 1$, and if $\delta < \lambda$ is the next largest magnitude of an eigenvector of $\tilde{P}$, then for any $\epsilon > 0$

$$(\delta + \epsilon)^{-n} ||\tilde{P}^n - \lambda^n \chi \omega|| \to 0 \tag{3.1}$$

as $n \to \infty$.

Write $\mu^*((i,j)) = r_{i,j}$, the restart distribution, and define the polynomial

$$f(\eta) = \sum_{i=1}^{b} \sum_{k=0}^{\lambda_i} \eta^{k+1} r_{i,k} - 1.$$

For reasons which shall become apparent, f is called the *structure polynomial*. Define $s = (\chi' \tilde{p}_0)^{-1}$.

Theorem 3.1. *If all $r_{i,k}$ are positive for $i \geq 1$ and if there is at least one x for which $g(x) \neq x$, then the Perron-Frobenius eigenvalue λ of $\tilde{P}$ is η_0^{-1}, where η_0 is the unique zero greater than 1 of the polynomial*

$$f(\eta) = \sum_{i=1}^{b} \sum_{k=0}^{\lambda_i} \eta^{k+1} r_{i,k} - 1.$$

The corresponding right eigenvector χ is proportional to v, where $v_{i,k} = \eta_0^k$, and the corresponding left eigenvector ω is proportional to the vector w, where

$$w_{i,k} = \sum_{u=k}^{\lambda_i} r_{i,u} \eta_0^{u-k},$$

$0 \leq k \leq \lambda_i, 1 \leq i \leq b$. Furthermore, defining

$$\theta_0 = \sum_{k=0}^{\lambda_0} r_{0,k},$$

if the deleted initial probability vector $\tilde{p}_0$ is the deleted vector $\tilde{r}$, then

$$s = \frac{\eta_0(\eta_0 - 1) f^{(1)}(\eta_0)}{\theta_0} > \eta_0 > 1.$$

Proof. First observe that the polynomial f satisfies $f(1) = 1 - \theta_0 - 1 = -\theta_0 < 0$, $f^{(1)}(\eta) > 0$, and $f^{(2)}(\eta) > 0$, so that there is a unique solution $\eta > 1$ to $f(\eta) = 0$. It is then a simple matter to verify that the given vector v is a right eigenvector and that the corresponding eigenvalue is η^{-1}, where $f(\eta) = 0$. It follows from Varga [5] that η^{-1} is the P-F eigenvalue of $\tilde{P}$. Writing $w' = (w_{1,0}, w_{1,1}, \ldots, w_{1,\lambda_1}, w_{2,0}, \ldots, w_{b,\lambda_b})$ and setting $r_{i,v} = w_{i,v} = 0$ for $v > \lambda_i$, one has the (i', k') coordinate of the vector

$w'\tilde{P} = \sum_{(i,v)} w_{i,v}\tilde{P}((i,v),(i',k')) = \sum_{i=1}^{b}\sum_{v=0}^{\lambda_i} w_{i,v}\tilde{P}((i,v),(i',k'))$ as

$$(w'\tilde{P})_{(i',k')} = \sum_{i=1}^{b} w_{i,0}\tilde{P}((i,0),(i',k')) + \sum_{i=1}^{b}\sum_{v=1}^{\lambda_i} w_{i,v}\tilde{P}((i,v),(i',k'))$$

$$= r_{i',k'}\sum_{i=1}^{b} w_{i,0} + w_{i',k'+1} = r_{i',k'}\sum_{i=1}^{b}\sum_{v=0}^{\lambda_i} r_{i,v}\eta^{v}$$

$$+ \sum_{v=k'+1}^{\lambda_{i'}} r_{i',v}\eta^{v-k'-1}.$$

Using $f(\eta) = 0$ on the last expression, we get

$$(w'\tilde{P})_{(i',k')} = r_{i',k'}\frac{1}{\eta} + \frac{1}{\eta}\sum_{v=k'+1}^{\lambda_{i'}} r_{i',v}\eta^{v-k'} = \frac{1}{\eta}\sum_{v=k'}^{\lambda_{i'}} r_{i',v}\eta^{v-k'} = \frac{1}{\eta}w_{i',k'}.$$

It has been demonstrated that the claimed eigenvalue is the Perron-Frobenius eigenvalue and the corresponding right and left eigenvectors have been found; our next step is to prove the formula for s. It is first shown that for arbitrary η, $w'(\eta)v(\eta) = f^{(1)}(\eta)$. We have

$$w'(\eta)v(\eta) = \sum_{i=1}^{b}\sum_{s=0}^{\lambda_i} w_{i,s}v_{i,s} = \sum_{i=1}^{b}\sum_{s=0}^{\lambda_i}\left(\sum_{x=s}^{\lambda_i} r_{i,x}\eta^{x-s}\right)\eta^{s}$$

and, interchanging the order of summation in the right-most sums, it is:

$$= \sum_{i=1}^{b}\sum_{x=0}^{\lambda_i}\sum_{s=0}^{x} r_{i,x}\eta^{x-s+s} = \sum_{i=1}^{b}\sum_{x=0}^{\lambda_i}(x+1)r_{i,x}\eta^{x-s+s} = f^{(1)}(\eta).$$

The left eigenvector ω is normalized so that $\omega' 1 = 1$. Since $\omega = aw$

one has

$$1 = aw'1 = a\sum_{i=1}^{b}\sum_{v=0}^{\lambda_i} w_{i,v} = a\sum_{i=1}^{b}\sum_{v=0}^{\lambda_i}\sum_{u=v}^{\lambda_i} r_{i,u}\eta^{u-v}$$

$$= a\left(\sum_{i=1}^{b}\sum_{u=0}^{\lambda_i} r_{i,u}\sum_{v=0}^{u}\eta^{u-v}\right) = a\sum_{i=1}^{b}\sum_{u=0}^{\lambda_i} r_{i,u}\eta^{u}\frac{1-(\eta^{-1})^{u+1}}{1-(\eta^{-1})}$$

$$= \frac{a}{\eta-1}\left[f(\eta)+1-\sum_{i=1}^{b}\sum_{u=0}^{\lambda_i} r_{i,u}\right] = \frac{a}{\eta-1}[f(\eta)+\theta_0].$$

Therefore, $a = \frac{\eta-1}{f(\eta)+\theta_0}$.

From the normalization of the right eigenvector we have from above $1 = \omega'\chi = aw'cv = acf^{(1)}(\eta_0)$ so that $c = 1/af^{(1)}(\eta_0)$. Since

$$s^{-1} = \alpha'\chi = r'cv = \frac{1}{af^{(1)}(\eta_0)}r'v = \frac{f(\eta_0)+\theta_0}{(\eta_0-1)f^{(1)}(\eta_0)}r'v$$

$$= \left(\frac{f(\eta_0)+\theta_0}{(\eta_0-1)f^{(1)}(\eta_0)}\right)\sum_{i=1}^{b}\sum_{u=0}^{\lambda_i} r_{i,u}\eta_0^{u}$$

$$= \left(\frac{f(\eta_0)+\theta_0}{(\eta_0-1)f^{(1)}(\eta_0)}\right)\frac{f(\eta_0)+1}{\eta_0},$$

one has, using $f(\eta_0) = 0$ and dropping the subscript,

$$s = \frac{\eta(\eta-1)f^{(1)}(\eta)}{\theta_0} = \frac{\eta(\eta-1)f^{(1)}(\eta)}{-f(1)}$$

$$= \eta\frac{f^{(1)}(\eta)}{(f(\eta)-f(1))/(\eta-1)} = \eta\frac{f^{(1)}(\eta)}{f^{(1)}(\xi)},$$

where the last step is a consequence of the mean value theorem and, finally, since $f^{(2)} > 0$, $s > \eta > 1$. □

Example 3.2. To illustrate the ideas, consider a small traveling salesman problem (TSP) on 7 cities. For the TSP on 7 cities there are 6!

possible tours so that $N = 720$ is the number of points in Ω. (Pairing every tour with its reverse leads to 360 distinct potential solutions but we ignore these pairings). The cities, placed in a 10×10 square, have locations 0:(2,2), 1:(7,3), 2:(4,5), 3:(8,7), 4:(1,6), 5:(6,9), 6:(3,8). The space Ω of tours is 0 (0,1,2,3,4,5,6,0); 1 (0,1,2,3,4,6,5,0); 2 (0,1,2,3,5,4,6,0); $\ldots$; 719 (0,6,5,4,3,2,1,0). The minimum tour length is 24.27 achieved by Tour 123 (0,2,1,3,5,6,4,0) and its reverse Tour 478. Tours always begin and end with city 0 so its mention is eliminated henceforth. The iterative algorithm g is taken as "successive city swap" defined by: given a Tour, then (Step 1) start at the leftmost tour position $i = 1$, and (Step 2) swap the cities in positions i and $i + 1$, (Step 3) if the tour length has not increased, replace Tour by this modification and go to Step 1, otherwise increment i and (Step 4) if $i = 6$ return Tour, otherwise go to Step 2. Among the basins defined by g is the depth 4 basin shown in Figure 1 ending with Tour $6 = (1, 2, 4, 3, 5, 6)$ which is a local minimizer.

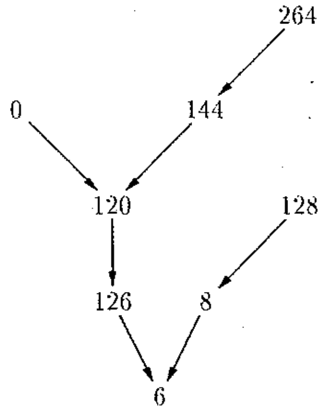

Figure 1

That the set of tours Ω is a topological space can be seen by taking as basic open sets the sets $B(x) = \bigcup_{k=-\infty}^{\infty}\{g^k(x)\}$ for $x \in \Omega$. The basin structure induced by g and the particular locations of the cities consists of 44 basins of which 2 are goal basins. The goal basins comprise 62 points so that $\theta_0 = 62/720 = 0.0861$. The length of the longest goal basin path is $d_0 = 6$. The longest non goal basin path has length $d = 9$ so the structure polynomial is of degree 10. The structure polynomial is exactly

$$f(\xi) = \frac{1}{720}(42\xi + 130\xi^2 + 174\xi^3 + 148\xi^4 + 93\xi^5 + 44\xi^6 + 18\xi^7 + 6\xi^8 + 2\xi^9 + \xi^{10}) - 1.$$

Its principal root is $\eta = 1.0254$, hence it follows from Theorem 3.1 that $\lambda = \eta^{-1} = 0.9753$. Observe that $\tilde{P}$ is a 658 by 658 square matrix and λ is the largest root of its 168th degree characteristic polynomial, but it has been found here as the solution of a 10th degree polynomial. One also calculates $s = 1.067$ and observes that $s > \eta$, as predicted by Theorem 3.1. □

Geometric convergence to zero of the the tail probabilities has been established. From the well known fact that $E[T] = \sum_{n\geq 0} P[T > n]$ and this tail convergence, Shonkwiler and Van Vleck [4] were able to show that under independent identical processing the ratio of expected times $\frac{E[T]}{E[T_m]}$ to hit the goal basin M satisfies $\frac{E[T]}{E[T_m]} \to ms^{m-1}$ as $\lambda \to 1$. Theorem 3.1 shows that under widely applicable assumptions, $s > 1$, so that quite generally, speedup by independent parallel processing, measured in terms of expected time to goal, is super-linear in large difficult problems, ones for which λ is close to 1.

The parameter λ, which is termed the retention rate by Shonkwiler and Van Vleck, has also a description in terms of the probability distribution of times to restart in non-goal basins.

Corollary 3.3. *Under the conditions of Theorem 3.1, λ solves*

$$\lambda^{-1}\phi_{H|N}(\lambda^{-1}) = (1-\theta_0)^{-1}$$

where $\phi_{H|N}$ is the probability generating function of the random time to first settling point given that the starting state is one which leads to a non-global extremum.

Proof.

$$\phi_{H|N}(z) = \sum_{n\geq 0} P[H = n|N]z^n = \sum_{n\geq 0}\sum_{i=1}^{b} \frac{r_{i,n}}{1-\theta_0} z^n.$$

Therefore

$$\phi_{H|N}(z) = \frac{f(z)+1}{z(1-\theta_0)}. \qquad \square$$

4. Estimating λ

Knowledge of the parameter λ can be used in determining the number n of iterations required to achieve $P[T > n] \approx 1/2$ as $n \approx \frac{-\ln 2}{\ln \lambda}$. If λ is known, then one would know how long to run the process, or using m independent processes, since $P[T_m > n] = P^m[T > n]$, to realize $P[T_m > n] \approx 1/2$ one would require $n_m \approx \frac{-\ln 2}{m \ln \lambda}$ iterations each to be run on m independent processors.

Corollary 3.3 can be used in connection with data accumulated during a run to estimate the parameter λ as shall be described below. However, and more simply, utilizing the convexity of the probability generating function one can obtain the bound $\bar{\lambda} = \frac{(1-\theta_0)(1+E[H|N])}{\theta_0+(1-\theta_0)(1+E[H|N])} \leq \lambda$ based on a linear approximation of the function $\eta\phi_{H|N}(\eta)$. Note that as $E[H|N] \to 0$ the bound becomes $1-\theta_0$ and as $E[H|N] \to \infty$ the bound approaches 1. Since the bound is monotonic in $E[H|N]$ it perhaps appears odd that $1-\theta_0 < \bar{\lambda}$ for every value of $E[H|N]$, for if one simply chose at random according to μ each time without employing the algorithm, then the rate of approach of the tail probability $P[T > n]$ to zero would be $(1-\theta_0)^n$, an apparently better rate. However, it should be kept in mind that the search is for a goal state, and if the algorithm is not being run, then this method would generally yield only a state in M, the basin of attraction M to the goal states G. Assuming the "size" of goal states is $\mu(G) \ll \mu(M) = \theta_o$ the actual rate of approach to a goal state for simple random search is $(1-\mu(G))^n \gg (1-\theta_0)^n$.

One statistic, available as the runs proceed, is the number of times I_j, it takes j iterations until the algorithm must be restarted, $j = 0, \ldots$. There are two possibilities; if the problem is one in which the goal is recognized when it is found, in such problems as the inverse fractal problems (see [1]) then one can with certainty estimate the conditional probabilities $p_j = P[H = j|N]$ of requiring j steps consistently by $\hat{p}_{j,n} = I_j/N_n$, where N_n is the number of restarts required by the *nth* epoch. In the other case in which the goal is not recognized, the worst case scenario is assumed at each stage, that the goal has not yet been found.

In the latter case, as long as the goal has not been found the procedure is consistently estimating the correct conditional probabilities, and if the goal has already been found, then the estimate is irrelevant. It is clear from elementary probabilistic considerations that $\hat{p}_{j,n} \to p_j$ for each $j = 0, 1, 2, \ldots$, where the convergence is almost sure.

We take our estimate of the probability generating function $\phi_{H|N}(z)$ to be

$$\hat{\phi}_{H|N}(z) = \sum_{j \geq 0} \hat{p}_{j,n} z^j$$

and, assuming for the moment that θ_0 is known, our estimate of λ is $\hat{\lambda}_n = 1/\eta_n$, where η_n solves

$$\eta_n \hat{\phi}_{H|N}(\eta_n) = (1 - \theta_0)^{-1}.$$

Theorem 4.4. *Under the conditions of Theorem 3.1 and conditional on having not yet hit the basin containing the global minimum by time epoch n,*

$$\sqrt{N_n}(\hat{\lambda}_n - \lambda) \to N(0, V),$$

where

$$V = \lambda^2 \frac{\phi(\lambda^{-2}) - \phi^2(\lambda^{-1})}{(\phi(\lambda^{-1}) - \lambda^{-1}\phi'(\lambda^{-1}))^2},$$

and convergence is in law.

Proof. Consider the function g of the $K+3$ variables $p_0, \ldots, p_K, x, u$, defined by

$$g(\mathbf{p}, x, u) = \sum_{j=0}^{K} p_j x^{j+1} - u = x\phi(x) - u.$$

For each probability vector $\mathbf{p}$ with all coordinates positive and $u > 1$ we have seen in Theorem 3.1 that there is a unique solution $x(\mathbf{p}, u)$ satisfying $g(\mathbf{p}, x(\mathbf{p}, u), u) = 0$. We are interested in the function $x(\mathbf{p}, u)$ and, applying the inverse function theorem find that for any fixed probability vector $\mathbf{p}_0$ whose entries are all positive and $u_0 > 1$ there is a neighborhood N_{pu} of $(\mathbf{p}_0, u_0)$ such that the function $x = x(\mathbf{p}, u)$ is continuously differentiable on N_{pu} with derivative

$$\nabla_{\mathbf{p}u} x(\mathbf{p}, u) = -\left[\frac{\partial g(\mathbf{p}, x, u)}{\partial x} \bigg|_{x = x(\mathbf{p}, u)} \right]^{-1} \nabla_{\mathbf{p},u} g(\mathbf{p}, x, u)|_{x = x(\mathbf{p}, u)}.$$

Since $\sqrt{N_n}(\hat{\mathbf{p}}_n - \mathbf{p}) \to N(\mathbf{0}, \mathbf{\Sigma})$, where $\Sigma_{i,j}$ is $-p_i p_j$ if $i \neq j$ and $p_i q_i$ if $i = j$, the usual application of the δ-method holding $u = (1-\theta_0)^{-1}$, yields

$$\sqrt{N_n}(\hat{\lambda}_n - \lambda) \to N(0, \lambda^4 (\nabla_{\mathbf{p}} x)' \mathbf{\Sigma} \nabla_{\mathbf{p}} x)$$

which upon simplification is the claimed result. □

By replacing λ and ϕ by their sample estimates $\hat{\lambda}$ and $\hat{\phi}$ in V one also has as a usual and practical consequence of Slutsky's theorems, that

$$\frac{(\hat{\lambda}_n - \lambda)}{\sqrt{\hat{V}/N_n}} \to N(0,1).$$

The result above depends on knowledge of the value θ_0 and it is not likely to be available. If one possessed a root-n consistent estimator $\hat{\theta}_{0,n}$ of it, then a like result could be obtained by taking $u_n = \hat{\theta}_{0,n}$. However, it is unclear under what circumstances such an estimator would be available and useful. For example, the estimator $\hat{\theta}_{0,n}$ which consists of the proportion of restarts which have occurred at the current best (= smallest) value of C attained is consistent, but as soon as one instance of the minimum has been obtained, the reason for estimating λ has disappeared! Before that event, the estimator is indicating the size of a basin whose size may be unrelated to the one of interest. It seems that there must be some knowledge of θ_0 to take full advantage of the machinery. Some use of preliminary and ongoing runs or of conservative estimates of θ_0 can be made as expressed in the following, whose proof is left to the reader.

Corollary 4.5. *Under the conditions of Theorem 3.1 with $\hat{\lambda}_n(\epsilon) = x(\hat{\mathbf{p}}_n, 1+\epsilon)$ and conditional on having not yet hit the basin containing the global minimum by time epoch n,*

$$\sqrt{N_n}(\hat{\lambda}_n(\epsilon) - \lambda(\epsilon)) \to N(0, V(\epsilon)),$$

where $\lambda^{-1}(\epsilon) = x(\mathbf{p}_0, 1+\epsilon)$ and $V(\epsilon)$ is as above with λ replaced by $\lambda(\epsilon)$. Furthermore, $\lambda(\epsilon)$ is greater or less than λ in accordance with whether θ_0 is greater or less than $\frac{\epsilon}{1+\epsilon} = \theta_\epsilon$.

Under the conditions of Corollary 4.5 we have an estimator of an upper or lower bound on the rate of convergence depending upon how

θ_ϵ is chosen. Taking $1+\epsilon = 2^{1/N_m}$, for example, where a preliminary run of m epochs results in N_m restarts, if θ_0 were greater than $\frac{\epsilon}{1+\epsilon}$, then the probability the goal would have been seen in N_m restarts would exceed 0.5. Thus either θ_0 is less than θ_ϵ or it is reasonably likely that the preliminary runs have discovered the goal basin. The choice of 0.5 is arbitrary and can be replaced, according to the desires of the searcher, with an appropriate value. The resulting $\lambda(\epsilon)$ is only a lower bound on the true geometric rate so that $\hat{\lambda}(\epsilon)$ will not be conservative; an underestimate of resources is the result. Prior information on the size of θ_0 is important as are the numbers of restarts as the search proceeds. A choice of θ_ϵ sufficiently small will lead to conservative estimates of the rate.

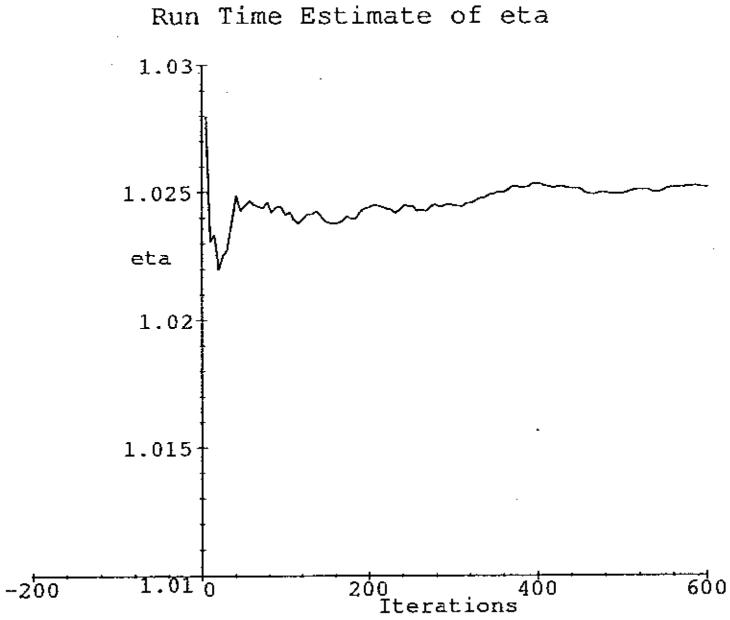

Figure 2

Example 4.6. Continuing Example 3.2, in which $\theta_0 = 0.0861$ and $\lambda = 0.97527561$, one finds $V = 5.475 \times 10^{-4}$. A choice of $\theta_\epsilon = 0.09$ and an application of Corollary 4.5 yields $\lambda(\epsilon) = 0.97412905$ while for $\theta_\epsilon = 0.06$ the result is $\lambda(\epsilon) = 0.98289893$. The lower bound obtained compares with the simple but slightly better lower bound $\bar{\lambda} = 0.974266$.

Simulation yields the following plot of the current estimate $\hat{\lambda}^{-1}$ (namely Figure 2).

References

[1] A. Deliu, J. Geronimo, R. Shonkwiler, On the inverse fractal problem for two-dimensional attractors, *Philos. Trans. Roy. Soc. London Ser. A*, **355**, No. 1726 (1997), 1017–1062.

[2] F. Mendivil, R. Shonkwiler, M.C. Spruill, Restarting search algorithms with applications to simulated annealing, *Adv. Appl. Prob.*, **33** (2001), 242–259.

[3] F. Mendivil, R. Shonkwiler, M.C. Spruill, Optimization by stochastic methods, Book Chapter in: *Handbook of Stochastic Analysis and Applications* (Ed-s: Kannan, Lakshimikanthan), Marcel Dekker, New York (2001), 627–679.

[4] R. Shonkwiler, and E. Van Vleck, Parallel speed-up of Monte Carlo methods for global optimization, *J. Complexity*, **10**, No. 1 (1994), 64–95.

[5] R. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs (1963).