Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

FINITE-SAMPLE PERFORMANCE GUARANTEES FOR ONE-DIMENSIONAL STOCHASTIC ROOT FINDING

Samuel M. T. Ehrlichman
Shane G. Henderson

School of Operations Research and Industrial Engineering
Cornell University
Ithaca, NY 14850, U.S.A.

ABSTRACT

We study the one-dimensional root finding problem for increasing convex functions. We give gradient-free algorithms for both exact and inexact (stochastic) function evaluations. For the stochastic case, we supply a probabilistic convergence guarantee in the spirit of selection-of-the-best methods. A worst-case bound on the work performed by the algorithm shows an improvement over naïve stochastic bisection.

1 INTRODUCTION

Let $\mathcal{H} = \mathcal{H}(m, c)$ be the set of all functions $h : [0, m] \to \mathbb{R}$ which are increasing, convex, continuous at the endpoints, having left derivative at m bounded above by $c < \infty$, and having $h(0) < 0 < h(m)$. In this paper we give algorithms for computing an approximate zero of $h \in \mathcal{H}$ under varying assumptions about the information available about h. More precisely, our algorithm returns a point $\bar{x} \in [0, m]$ satisfying $|h(\bar{x})| < \delta$ for some pre-specified $\delta > 0$ in the deterministic setting, and returns such a point with high probability in the stochastic setting. We will refer to such an $\bar{x}$ as a δ-*root*. This differs from the usual notion of an approximate root in numerical analysis, which requires that the *interval* bracketing the root be small upon termination of the root finding algorithm.

Our interest in this form of the stochastic root finding problem arises from the one-dimensional American option pricing problem. In pricing American options, one wishes to know at each decision point in time whether to exercise an option or continue holding it. One should exercise if the "exercise value" exceeds the "continuation value." Under conditions that ensure monotonicity, the optimal policy is to exercise if and only if the price of the underlying asset lies on one side of the root of a certain convex function. The absolute value of this function at the reported root is an appropriate measure of the error associated with the approximation.

One of the first papers on simulation-based approaches to American option pricing, Grant, Vora, and Weeks (1996), proposes an algorithm wherein the root is approximately computed at each stage of a dynamic program. This procedure is quite straightforward and intuitive, but the actual root finding subroutine employed is ad hoc and does not come with a convergence guarantee.

A standard approach to stochastic root finding is stochastic approximation, which has an enormous literature; see, e.g., Kushner and Yin (2003). However, stochastic approximation is fundamentally a *search* procedure, and does not provide a probabilistic guarantee. This is also true of Simultaneous-Perturbation Stochastic Approximation (Spall 2003). Chen and Schmeiser (2001) develop retrospective root finding algorithms that progressively narrow in on a root, and, like stochastic approximation, the theory is directed towards proving convergence in a limiting sense, rather than offering finite-time performance guarantees. This is also true of the root finding procedures developed in Pasupathy and Schmeiser (2003), Pasupathy and Schmeiser (2004).

Our work is related to algorithms developed for the root finding problem with exact function evaluations, especially Gross and Johnson (1959), but also of note are Potra (1994), in which a quadratic rate of convergence is established for a root finding algorithm applied to a subclass of smooth convex functions, and Rote (1992), which surveys and analyzes the sandwich algorithm for shrinking a polygonal envelope of a convex function. None of the papers we have seen attempt to identify a δ-root, instead adopting other error measures.

Therefore, the first main contribution of our work, namely the development and analysis of an algorithm for δ-root finding with *exact* function evaluations, shares the problem context of these papers, but we work with a different error measure. The second main contribution extends this algorithm and its analysis to the case of *inexact* function-evaluations.

Perhaps the most natural approach for identifying a δ-root of a monotone function in the deterministic setting is the bisection algorithm. To apply bisection to our problem, we evaluate h at the midpoint of an interval that is known to contain a root, and the interval is then reduced to the left half or the right half depending on the sign of the result. The algorithm terminates once it locates a δ-root $\bar{x}$. Our assumptions on h imply that such an $\bar{x}$ is revealed by the time the bracket has width less than or equal to $2\delta/c$. In that case, the bracket midpoint is guaranteed to be a δ-root. Here, the "bracket" refers to the smallest interval (a,b) such that both $h(a)$ and $h(b)$ have been estimated and such that it is known that $h(a) < 0 < h(b)$. Since the bracket width is halved with each function evaluation, the maximum number of function evaluations used in the bisection algorithm is $\lceil \log_2 cm/\delta \rceil - 1$.

When h cannot be evaluated exactly, the situation is more complicated. Suppose now that for a given x, we are only able to obtain an interval estimate that is known to contain $h(x)$ with a given probability. This is the case in the American option-pricing setting mentioned above. Our goal is now to design a procedure that delivers a a δ-root with high probability.

Bisection can again be made to work for this case. One ensures that each interval estimate returned for a given x has size at most δ. If said interval lies wholly above (wholly below) 0, then one asserts that $h(x) > 0$ ($h(x) < 0$), and the bisection algorithm proceeds as before. If the interval that is returned *contains* 0, then the algorithm halts and asserts that x is a δ-root. Under the assumption that each interval estimate returned during the procedure does indeed contain the true function value, this bisection procedure will return a true δ-root. And since we can predetermine a bound on the number of points at which the function should be evaluated in order to obtain an approximate root, one can design the procedure to have a probabilistic convergence guarantee. We shall refer to the algorithm just described as the *stochastic bisection algorithm.*

Our algorithm is very similar to stochastic bisection; a sequence of interval estimates, $(y_1^-, y_1^+), (y_2^-, y_2^+), \ldots$, is computed such that with high probability, $y_j^- \le h(x_j) \le y_j^+$ for each j. The algorithm terminates once it can return a point which is guaranteed to be a δ-root conditional on the computed interval estimates containing the corresponding true function values. Where our algorithm differs from stochastic bisection is in our construction of the sequence $x_1, x_2, \ldots$ of points where the function value is sampled. By exploiting the geometry of convex functions, we compute a region in which the graph of h is known to lie. We are then able to choose points more intelligently than bisection, and consequently can derive a lesser upper bound on the number of necessary function evaluations.

The remainder of the paper is organized as follows. In Section 2, we exhibit a particular function for which bisection is inefficient. We show pictorially how our algorithm works compared to bisection for this function in the case where function evaluations are exact. In Section 3, we show how one can compute upper and lower approximations of a convex function given interval estimates of the function's value at finitely many points. Section 4 gives the algorithm for exact function evaluations and contains our main result, the bound on the required number of function evaluations. Section 5 extends the algorithm to the inexact function evaluation case and provides a probabilistic analog to the result in Section 4. Section 6 concludes.

2 AN EXAMPLE

In this section, we consider the deterministic case (evaluations of h are exact).

Consider the function $h_{\text{bad}} \in \mathcal{H}$ given by $h_{\text{bad}}(x) = [c(x-m)+2\delta]_+ - \delta$. This function is equal to the constant $-\delta$ on the interval $[0, m-2\delta/c]$, then increases linearly at rate c on the interval $[m-2\delta/c, m]$. According to our definition, the interior of the latter interval is precisely the set of δ-roots of h_{bad}.

It is easy to see that the bisection algorithm evaluates $h_{\text{bad}}(x)$ at the sequence of points $x = m/2, 3m/4, \cdots, (2^k - 1)m/2^k$, where $k = \lceil \log_2 cm/\delta \rceil - 1$, after which the algorithm terminates. In other words, the theoretical maximum number of function evaluations is actually required for this example; see Figure 1.

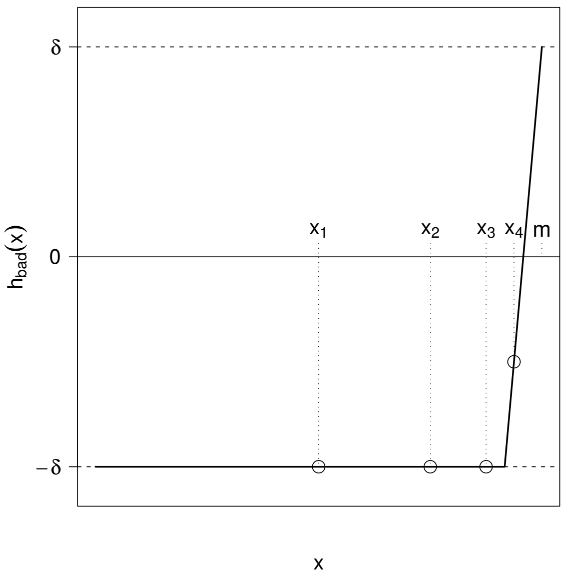

Figure 1: Bisection approach on an example function h_{bad}.

In this example, it is evident that choosing sampling points to the right of the midpoint of the bracket would result in fewer total function evaluations; see Figure 2. Of course, this would not be a sound strategy for general increasing functions. For example, if the true function were increasing and concave, such as $\tilde{h}_{\text{bad}}$ given by $\tilde{h}_{\text{bad}}(x) = -h_{\text{bad}}(m-x)$, then we would prefer to sample to the left of the midpoint instead. Indeed, for general monotone, increasing functions, it is clear that sampling at the midpoint is optimal in the minimax sense.

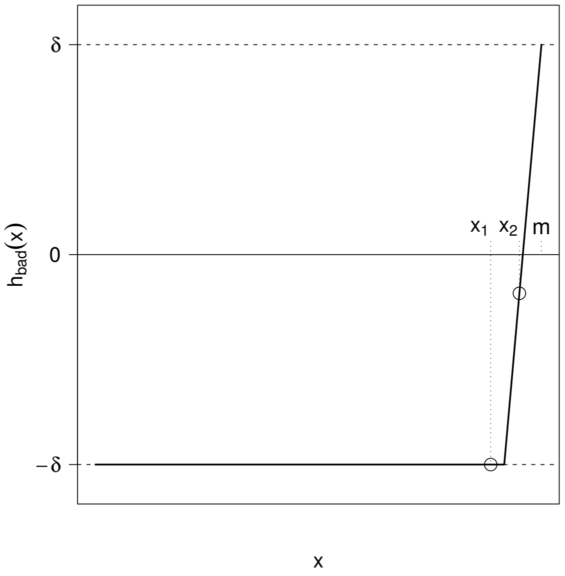

Figure 2: Our approach on an example function h_{bad}.

For $h \in \mathcal{H}$, it turns out that sampling a particular point, distinct from the bracket midpoint, is provably superior to bisection. We make this claim precise in Section 4, but give a heuristic argument here. Suppose our new sampling point is strictly to the right of the midpoint. Then if the new point x has $h(x) < 0$, the bracket is reduced by a factor *strictly exceeding* $1/2$, whereas if $h(x) > 0$, then convexity implies that a new upper bound $\tilde{c} < c$ on the slope of h at its root is established. In the latter case, the algorithm may now terminate upon finding a bracket of width at most $2\delta/\tilde{c} > 2\delta/c$. The key idea behind the proof of faster-than-bisection convergence is that one can balance the impact of these two possibilities.

3 UPPER AND LOWER ENVELOPES

Suppose we have estimated the value of h at k points $x_1, \ldots, x_k$, yielding interval estimates $(y_1^-, y_1^+), \ldots, (y_k^-, y_k^+)$ of $h(x_1), \ldots, h(x_k)$. Let $\mathcal{H}_k$ denote the set of functions $h \in \mathcal{H}$ satisfying $y_j^- \le h(x_j) \le y_j^+$, for $j = 1, \ldots, k$. Define the lower and upper envelope functions l_k and u_k by $l_k(x) = \inf\{h(x) : h \in \mathcal{H}_k\}$ and $u_k(x) = \sup\{h(x) : h \in \mathcal{H}_k\}$; see Figure 3. The vertical line segments indicate the intervals (y_j^-, y_j^+), $j = 1, 2, 3$.

It turns out that it is straightforward to construct u_k and l_k incrementally as additional information about h is revealed. Details are given in (Ehrlichman and Henderson 2007). Our procedure for updating the envelope functions is an extension of the "sandwich" approximation for convex functions (Rote 1992) to the case where function evaluations are inexact. Figure 4 shows how the envelopes might change as a result of including a fourth point.

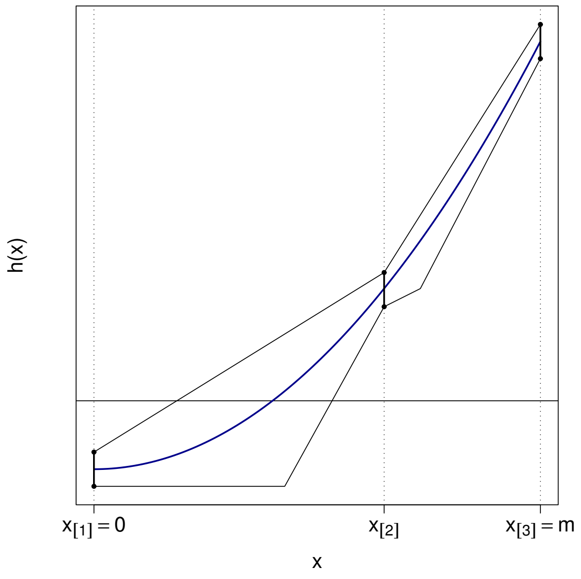

Figure 3: Upper and lower envelopes.

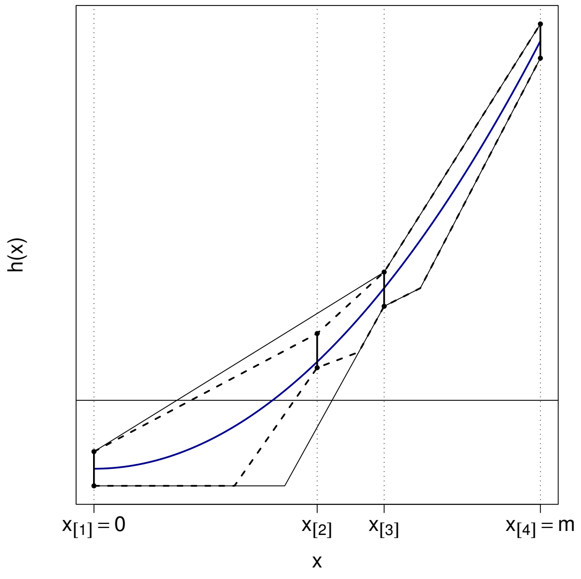

Figure 4: Updating the envelope functions. The dashed lines show the new envelopes after sampling at $x_{[2]}$.

By construction, u_k is convex and strictly increasing, and l_k is non-decreasing. We denote by l_k^{-1} the right inverse of l_k, given by $l_k^{-1}(y) = \sup\{x : l_k(x) = y\}$.

The envelopes l_k and u_k provide information that is quite useful for locating a δ-root. One observation we may immediately make is that

$$x_{[i]} < u_k^{-1}(0) \le h^{-1}(0) \le l_k^{-1}(0) < x_{[i+1]},$$

where $x_{[j]}$ denotes the jth order statistic of $\{x_1,\ldots,x_k\}$ and $(x_{[i]},x_{[i+1]})$ is the current bracket. The above inequality exhibits an interval strictly contained within the bracket that is known to contain the root of h. This suggests that our measure of progress in seeking a δ-root should be related to the width of this smaller interval, rather than to the bracket itself.

A more direct way in which the envelopes come into play is in the algorithm's stopping criterion. Suppose we are in a situation where

$$l_k^{-1}(-\delta) < u_k^{-1}(\delta). \tag{1}$$

Then *every* $x \in (l_k^{-1}(-\delta), u_k^{-1}(\delta))$ is a δ-root, provided that the interval estimates defining $\mathcal{H}_k$ are accurate. In particular, this may allow the algorithm to terminate *without ever having sampled at a δ-root*. Figure 5 depicts the upper and lower envelopes in such a case; a box is drawn about the region in the graph which is known to contain only δ-roots.

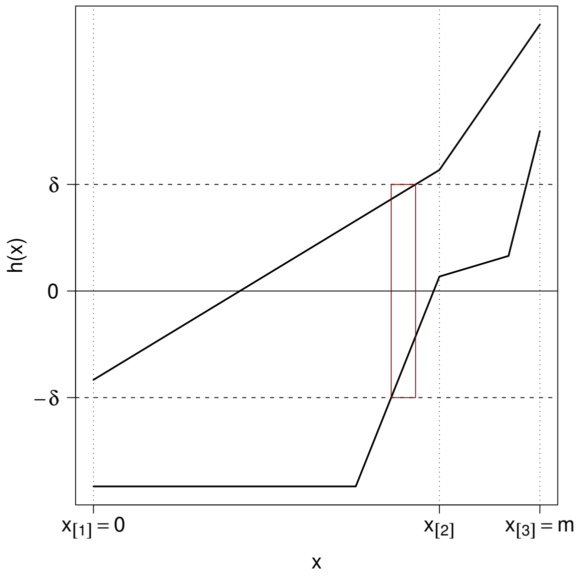

Figure 5: Stopping condition.

Finally, it turns out that the construction of the lower envelope l_k produces as a byproduct the sequence of intervals $(z_{[1]}^-, z_{[1]}^+),\ldots,(z_{[k]}^-, z_{[k]}^+)$, where for each $j=1,\ldots,k$, $(z_{[j]}^-, z_{[j]}^+)$ is the union of the subdifferentials at $x_{[j]}$ of all functions $h \in \mathcal{H}_k$. Observe that our order statistic notation is consistent; that is, if $x_j < x_s$, then $z_j^- < z_s^-$ and $z_j^+ < z_s^+$. Moreover, without sampling h at all, we know from our assumptions on $\mathcal{H}$ that $0 \le z_j^- \le z_j^+ \le c$ for all j.

In the nest section, we will see how these quantities are used in the algorithm and its analysis.

4 EXACT FUNCTION EVALUATIONS

In this section, we derive a rule for selecting the next point at which to evaluate h and state a bound on the number of function evaluations which this rule engenders. We work in the setting where function evaluations are exact. In terms of our existing notation, this means that we assume $h(x_j) = l_k(x_j) = y_j^- = y_j^+ = u_k(x_j)$ for all $j=1,\ldots,k$. We denote this common quantity by y_j. Since h is non-decreasing, the order statistics of $\{x_1,\ldots,x_k\}$ and $\{y_1,\ldots,y_k\}$ satisfy $y_{[j]} = h(x_{[j]})$, for $j=1,\ldots,k$.

Suppose the current bracket is $(x_{[i]},x_{[i+1]})$, i.e. $y_{[i]} < 0 < y_{[i+1]}$. As noted at the end of Section 2, the measure of progress towards locating a δ-root should be a function of both the size (in some sense) of the search region and of the maximum possible right derivative at the root.

We measure the size of the search region by the area of the *search triangle* T_k, depicted in Figure 6 and defined by the following three lines,

1. the x-axis,
2. the line joining the points $\left(x_{[i]},y_{[i]}\right)$ and $\left(x_{[i+1]},y_{[i+1]}\right)$, and
3. the line passing through $\left(x_{[i+1]},y_{[i+1]}\right)$ with slope $z_{[i+1]}^+$.

We define a quantity q_k, which measures the work to find a δ-root that remains after k steps, to be proportional to the product of $z_{[i+1]}^+$ and the area of the search triangle,

$$q_k = 2\cdot z_{[i+1]}^+ \cdot \text{area}\left(T_k\right) = \left(y_{[i+1]}\right)^2\left(\frac{z_{[i+1]}^+}{\Delta} - 1\right), \tag{2}$$

where $\Delta = (y_{[i+1]} - y_{[i]})/(x_{[i+1]} - x_{[i]})$.

We refer to q_k as the *potential* after k steps. The following proposition shows that q_k is a reasonable measure of the remaining work. The proof, which is elementary and geometric, is given in (Ehrlichman and Henderson 2007).

Proposition 1. *If the bracket is $\left(x_{[i]},x_{[i+1]}\right)$, then*

$$\begin{aligned} 0 &\le l_k\left(x_{[i+1]} - y_{[i+1]}/z_{[i+1]}^+\right) \\ &\le u_k\left(x_{[i+1]} - y_{[i+1]}/z_{[i+1]}^+\right) \\ &\le \frac{1}{2}\sqrt{q_k}. \end{aligned}$$

In particular, Proposition 1 implies that the stopping condition (1) is met when $q_k \le 4\delta^2$; in this case $x_{[i+1]} - y_{[i+1]}/z_{[i+1]}^+$ is a δ-root.

We now discuss the strategy for selecting the next point $x_* \in (x_{[i]},x_{[i+1]})$ at which to evaluate h. Let $\{\tilde{x}_j,\tilde{y}_j, j =$

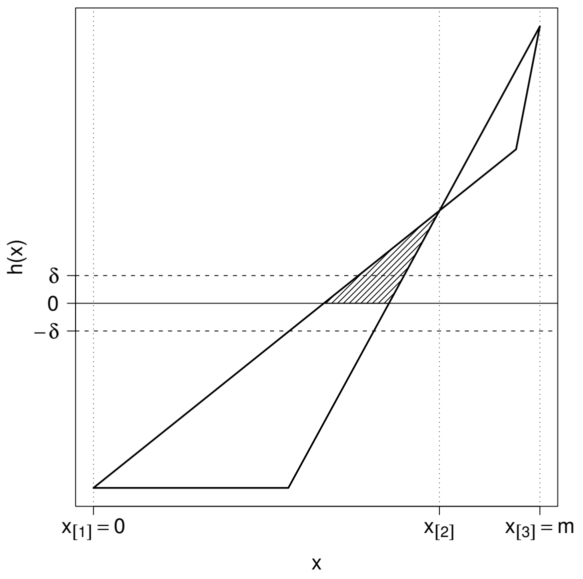

Figure 6: Search triangle (shaded), exact function evaluations.

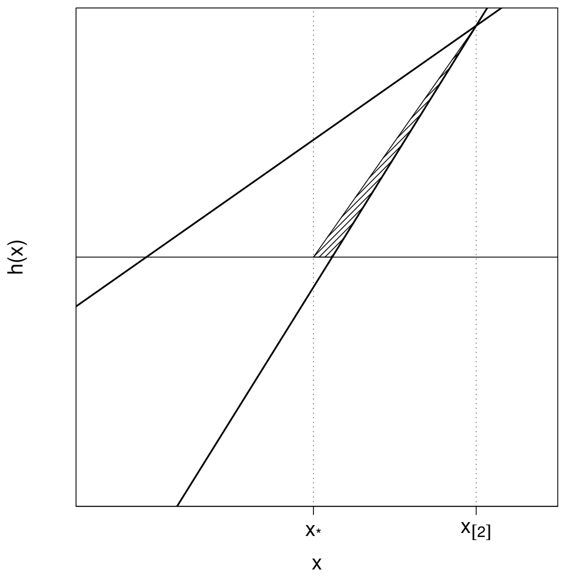

Figure 7: New search triangle when $h(x_*) < 0$, exact function evaluations.

$1, \ldots, k+1\}$ denote the information defining $\mathcal{H}_{k+1}$; let $\tilde{z}_j^+ = \sup \bigcup_{h \in \mathcal{H}_{k+1}} \partial h(\tilde{x}_{[j]})$, $j = 1, \ldots, k+1$.

Suppose first that $h(x_*) < 0$, so that the new bracket is $(\tilde{x}_{[i+1]}, \tilde{x}_{[i+2]}) = (x_*, x_{[i+1]})$. Let q_{k+1}^- denote the new potential in this case. Then the only way in which the terms comprising q_{k+1}^- differ from those in q_k is that Δ is replaced by

$$\tilde{\Delta}^- = \frac{\tilde{y}_{[i+2]} - \tilde{y}_{[i+1]}}{\tilde{x}_{[i+2]} - \tilde{x}_{[i+1]}} > \frac{y_{[i+1]}}{x_{[i+1]} - x_*}.$$

Therefore,

$$\begin{aligned} q_{k+1}^- &= \left(\tilde{y}_{[i+2]}\right)^2 \left(\tilde{z}_{[i+2]}^+ / \tilde{\Delta}^- - 1\right) \\ &= \left(y_{[i+1]}\right)^2 \left(z_{[i+1]}^+ / \tilde{\Delta}^- - 1\right) \\ &< y_{[i+1]} \left(z_{[i+1]}^+ \left(x_{[i+1]} - x_*\right) - y_{[i+1]}\right). \end{aligned} \tag{3}$$

A search triangle whose potential dominates that of the new search triangle $\tilde{T}_{k+1}$ is shown in Figure 7.

Suppose instead that $h(x_*) = y_* > 0$. Then the new potential is given by

$$q_{k+1}^+ = y_*^2 \left(\tilde{z}_{[i+1]}^+ / \tilde{\Delta}^+ - 1\right),$$

where

$$\tilde{\Delta}^+ = \frac{\tilde{y}_{[i+1]} - \tilde{y}_{[i]}}{\tilde{x}_{[i+1]} - \tilde{x}_{[i]}} = \frac{y_* - y_{[i]}}{x_* - x_{[i]}} > \frac{y_*}{x_* - \left(x_{[i+1]} - y_{[i+1]}/\Delta\right)}$$

and $\tilde{z}_{[i+1]}^+ = (y_{[i+1]} - y_*)/(x_{[i+1]} - x_*)$. The resulting expression bounding q_{k+1}^+ is maximized when $y_* = \frac{1}{2}(y_{[i+1]} - \Delta(x_{[i+1]} - x_*))$, yielding

$$q_{k+1}^+ < \frac{1}{4} y_{[i+1]} \frac{\left(y_{[i+1]} - \Delta\left(x_{[i+1]} - x_*\right)\right)^2}{\Delta\left(x_{[i+1]} - x_*\right)}. \tag{4}$$

The dominating search triangle is shown in Figure 8.

The right-hand side of (3) (respectively, of (4)) is monotone decreasing (respectively, increasing) in x_* on the interval $(u_k^{-1}(0), l_k^{-1}(0))$. Since we must have $q_{k+1} \in \{q_{k+1}^-, q_{k+1}^+\}$, it follows that we can minimize the maximum value q_{k+1} can attain by equating the right-hand sides of (3) and (4), then solving for x_*. Accordingly, we get

$$x_* = x_{[i+1]} - y_{[i+1]} \frac{1/\Delta}{2\sqrt{z_{[i+1]}^+/\Delta} - 1}. \tag{5}$$

We combine (2), (3), and (5) to conclude that

$$\frac{q_{k+1}}{q_k} < \frac{\sqrt{z_{[i+1]}^+/\Delta} - 1}{2z_{[i+1]}^+/\Delta + \sqrt{z_{[i+1]}^+/\Delta} - 1}. \tag{6}$$

Consider the right-hand side of (6) as a function of $\eta = \sqrt{z_{[i+1]}^+/\Delta}$. From the graph of said function, depicted in Figure 9, we can see that $q_{k+1}/q_k \leq 1/9$, with equality at $\eta = 2$. This fact and Proposition 1 suffice to establish the main result of this section, Theorem 2.

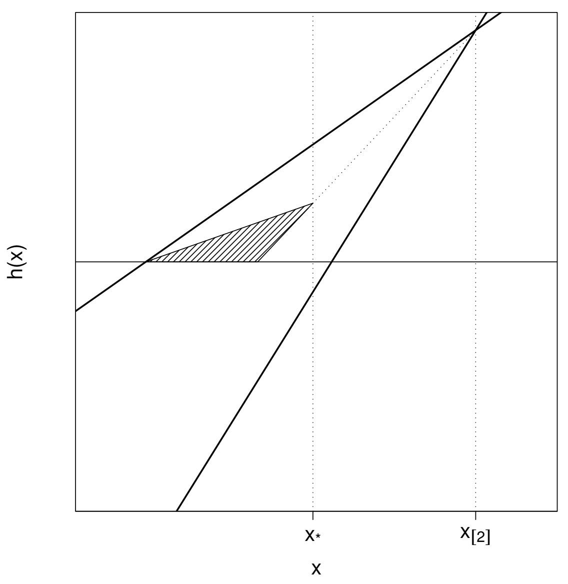

Figure 8: New search triangle when $h(x_*) > 0$, exact function evaluations.

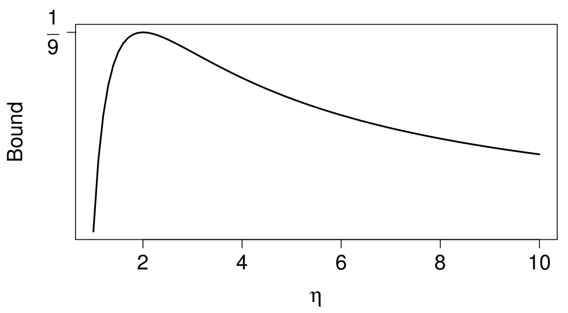

Figure 9: Bound on q_{k+1}/q_k as a function of $\eta = \sqrt{z_{[i+1]}/\Delta}$.

Theorem 2. *Assume $y_{[1]} = h(0) < 0$ and $y_{[2]} = h(m) > 0$ are known. Then the convex δ-root finding algorithm requires at most*

$$\left\lceil \log_3 \frac{\sqrt{q_2}}{2\delta} \right\rceil < \left\lceil \log_3 \frac{cm}{2\delta} \right\rceil$$

(exact) function evaluations.

5 INEXACT FUNCTION EVALUATIONS

Two complications arise when we consider the case in which function evaluations are inexact. One is that (6) fails to hold in this setting, so there is some work to be done in order to get a result analagous to Theorem 2. The other is the fact that our knowledge about $\mathcal{H}_k$ is accurate only with high probability. In this section, we discuss how to address these two issues.

5.1 An Analagous Convergence Result

Many of the results of the previous section go through with only minor modifications in our current setting, as long as we make the assumption that our interval estimates are of width no greater than δ. Under that assumption, let us now define the notion of the search triangle T_k to be as in the previous section except with $y_{[i]}$ and $y_{[i+1]}$ replaced by $y^+_{[i]}$ and $y^+_{[i+1]}$. See Figure 10, and compare Figure 6. We abuse the order statistic notation slightly here, taking $(y^-_{[j]}, y^+_{[j]})$ to be the interval estimate for $h(x_{[j]})$, $j = 1, \dots, k$.

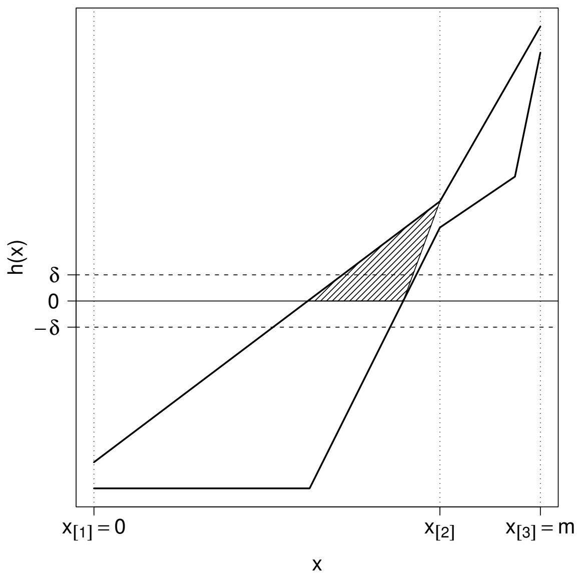

Figure 10: Search triangle in inexact function evaluation case.

Our definition of the potential in this setting, analagous to (2), is

$$q_k = 2 \cdot z_{[i+1]} \cdot \text{area}\,(T_k) = \left(y^+_{[i+1]}\right)^2 \left(\frac{z^+_{[i+1]}}{\Delta} - 1\right), \quad (7)$$

where $\Delta = (y^+_{[i+1]} - y^+_{[i]})/(x_{[i+1]} - x_{[i]})$. It turns out that Proposition 1 holds here with only minimal modification.

Proposition 3. *If the bracket is $\left(x_{[i]}, x_{[i+1]}\right)$, then*

$$\begin{aligned} -\delta &\le l_k\left(x_{[i+1]} - y^+_{[i+1]}/z^+_{[i+1]}\right) \\ &\le u_k\left(x_{[i+1]} - y^+_{[i+1]}/z^+_{[i+1]}\right) \\ &\le \frac{1}{2}\sqrt{q_k}. \end{aligned}$$

We sample at new points according to the rule

$$x_* = x_{[i+1]} - y^+_{[i+1]} \frac{1/\Delta}{2\sqrt{z^+_{[i+1]}/\Delta} - 1}; \qquad (8)$$

compare with (5). Depending upon the sign of $h(x_*)$, this gives new search triangles as depicted in Figures 11 and 12. In the case $h(x_*) < 0$, we get the same guaranteed reduction factor as in the previous section. However, if $h(x_*) > 0$, the resulting expression bounding the new potential turns out to be

$$\frac{q^+_{k+1}}{q_k} < \left(\frac{y^+_{[i+1]} + \delta}{y^+_{[i+1]}}\right)^2 \frac{\sqrt{z^+_{[i+1]}/\Delta} - 1}{2z^+_{[i+1]}/\Delta + \sqrt{z^+_{[i+1]}/\Delta} - 1};$$

compare with (6).

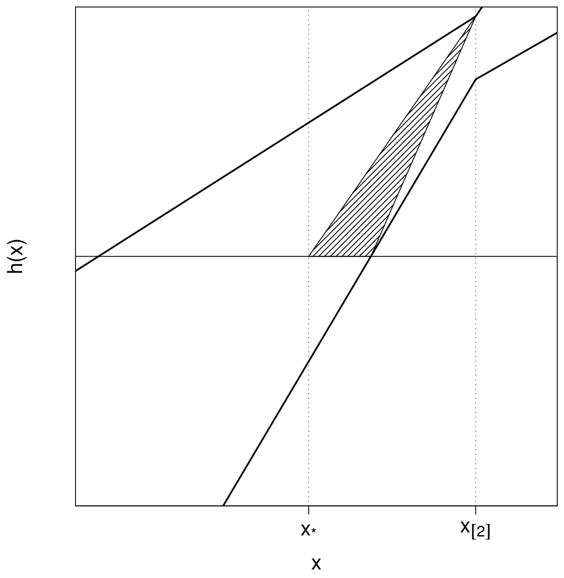

Figure 11: New search triangle when $h(x_*) < 0$, inexact function evaluations.

As long as there is still a lot of work to do, the extra factor of $((y^+_{[i+1]} + \delta)/y^+_{[i+1]})^2$ has only modest impact. Only in the final three steps does the rate of progress slow significantly. Theorem 4 summarizes this result. The idea of the proof is to split the analysis of the algorithm up into two stages. The first stage reduces the potential until it is less than $(19\delta)^2$. This happens at a fixed rate. The second stage reduces the potential to $4\delta^2$, which requires at most three more steps. Details are provided in (Ehrlichman and Henderson 2007).

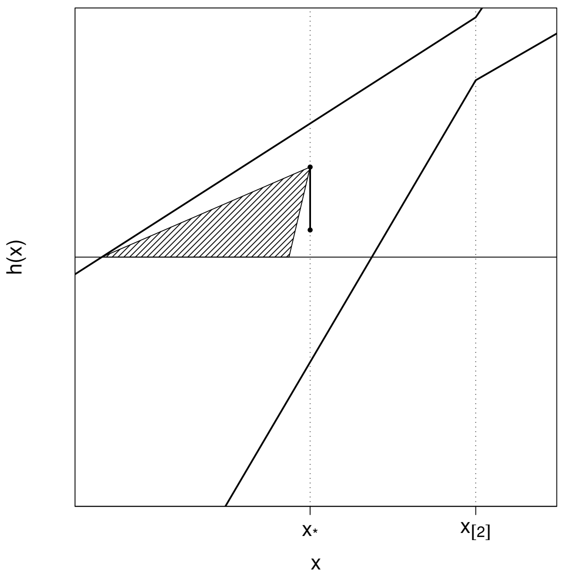

Figure 12: New search triangle when $h(x_*) > 0$, inexact function evaluations.

Theorem 4. *Assume y^-_j, y^+_j are known, $j = 1, 2$. Then the convex δ-root finding algorithm requires at most*

$$\left\lceil \log_\theta \frac{\sqrt{q_2}}{19\delta} \right\rceil + 3 < \left\lceil \log_\theta \frac{cm + \delta}{19\delta} \right\rceil + 3$$

(inexact) function evaluations, where each evaluation gives an interval estimate of width less than or equal to δ. Here, θ is a constant satisfying $\theta > 2.73$.

5.2 Providing a Probabilistic Guarantee

So far we have assumed that all intervals are guaranteed to enclose the true function values. In the context of stochastic root finding, however the best that one can hope for is that a given interval covers the true function value with high probability. This is certainly the case when the intervals are *confidence* intervals as constructed through a central-limit theorem procedure, or through the use of Chebyshev-type inequalities as on pp.21-22 of Fishman (1996). By carefully choosing the confidence levels of such intervals, we can construct a probabilistic guarantee for our root finding procedure. More precisely, we can make the statement:

> For given constants δ and $\alpha \in (0, 1)$, the probability that the (random) point our algorithm returns is indeed a δ-root is at least $1 - \alpha$.

This statement is similar in principle to the probability-of-correct-selection guarantees that come with ranking and selection procedures; see Kim and Nelson (2006).

A straightforward approach to obtaining a probabilistic guarantee was outlined in the introduction. A key point is that we have a bound (Theorem 4) on the number of intervals that need to be generated before we can stop the algorithm. We then use Bonferroni's inequality to set the confidence level associated with each interval so that the probability that all intervals contain their respective function values is at least $1-\alpha$. For example, if we want $1-\alpha=0.95$, and the bound on the number of intervals required is 5, then we can set the confidence level associated with each individual interval to be 0.99.

This somewhat brute-force approach to designing the overall root finding procedure ignores the information contained in the intervals that are successively obtained. In particular, it requires that each interval have a uniformly high probability that it covers the true value, and since the intervals need to have width at most δ, large simulation run-lengths may be necessary at each x value.

An improvement is to use a method we call *adaptive* α*-spending* that still ensures that the algorithm has an α-guarantee, but will, in many cases, reduce the simulation run-lengths required to obtain the intervals. The key idea is that, at each step of the algorithm, we re-evaluate a new bound on the number of steps that remain, and adjust the coverage probabilities of the remaining intervals accordingly. Continuing the example above, the first interval may have confidence level 0.99, but if we subsequently find that we will need at most 2 more intervals (via Theorem 4), then the confidence level of the second interval can be chosen to be 0.98.

It can be rigorously shown (Ehrlichman and Henderson 2007) that adaptive α-spending still gives a $1-\alpha$ approximation, and that it is guaranteed to require less computational effort than the brute-force approach sketched above. Unfortunately, no such adaptive strategy is available for the stochastic bisection method because the bound on the number of intervals required to stop never reduces by more than one at each step.

6 CONCLUSION

We conclude by examining the performance of our algorithm on the function h_{bad} of Section 2. Assume that h_{bad} has already been sampled at the endpoints, and that these interval estimates are centered at the true values $h_{\text{bad}}(0)=-\delta$ and $h_{\text{bad}}(m)=\delta$. Our earlier discussion of h_{bad} showed that the theoretical maximum number of function evaluations is actually required for stochastic bisection to produce a δ-root. Table 1 compares this quanitity (column "SB") both with the theoretical maximum number of evaluations required by our procedure according to Theorem 4 (column "Bound"), and the realized number of evaluations assuming symmetric confidence intervals about the true function values (column "Realized"). We take $\delta=.01$, $c=1$, and vary the scale of the problem by varying m.

Table 1: Performance on h_{bad}.

m	SB	Bound	Realized
10^2	13	5	2
10^4	19	8	3
10^6	26	10	3
10^8	33	12	4
10^{10}	39	14	4

Of course, our algorithm does not necessarily outperform stochastic bisection in practice on every possible function $h\in\mathcal{H}$. (Just consider a function in which the midpoint of the initial bracket is itself the root.) But it does provide small-sample performance guarantees which are nearly always an improvement.

ACKNOWLEDGMENTS

This research was partially supported by NSF Grant DMI 0400287.

REFERENCES

Chen, H., and B. Schmeiser. 2001. Stochastic root finding via retrospective approximation. *IIE Transactions* 33 (3): 259–275.

Ehrlichman, S. M. T., and S. G. Henderson. 2007. Stochastic root finding in one dimension for increasing convex functions. Working paper, Cornell University Dept. of ORIE.

Fishman, G. 1996. *Monte Carlo: concepts, algorithms, and applications*. Springer New York.

Grant, D., G. Vora, and D. Weeks. 1996. Simulation and the Early-Exercise Option Problem. *Journal of Financial Engineering* 5 (3): 211–227.

Gross, O., and S. Johnson. 1959. Sequential Minimax Search for a Zero of a Convex Function. *Mathematical Tables and Other Aids to Computation* 13 (65): 44–51.

Kim, S.-H., and B. L. Nelson. 2006. Selecting the best system. In *Simulation*, Volume 13 of *Handbooks in Operations Research and Management Science*. Amsterdam: Elsevier.

Kushner, H., and G. Yin. 2003. *Stochastic approximation algorithms and applications*. second ed. Springer.

Pasupathy, R., and B. W. Schmeiser. 2003. Some issues in multivariate stochastic root finding. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. Sanchez, D. Ferrin, and D. Morrice. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pasupathy, R., and B. W. Schmeiser. 2004. Retrospective approximation algorithms for the multidimensional stochastic root-finding problem. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. Ingalls, M. Rossetti, J. Smith, and B. Peters. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Potra, F. 1994. Efficient hybrid algorithms for finding zeros of convex functions. *Journal of Complexity* 10 (2): 199–215.

Rote, G. 1992. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing* 48 (3): 337–361.

Spall, J. 2003. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc. New York, NY, USA.

AUTHOR BIOGRAPHIES

SAMUEL M. T. EHRLICHMAN is a Ph.D. student in the School of Operations Research and Information Engineering at Cornell University. His interests include simulation-optimization, with emphasis on applications in financial engineering. He received a John McMullen Graduate Fellowship from Cornell and an NDSEG Fellowship. His web page can be found via <http://www.orie.cornell.edu>.

SHANE G. HENDERSON is an associate professor in the School of Operations Research and Information Engineering at Cornell University. He is the simulation area editor at *Operations Research*, and an associate editor for the *ACM Transactions on Modeling and Computer Simulation* and *Operations Research Letters*. He co-edited the handbook *Simulation* as part of Elsevier's series of Handbooks in Operations Research and Management Science, and also co-edited the Proceedings of the 2007 Winter Simulation Conference. He likes cats but is allergic to them. His research interests include discrete-event simulation and simulation optimization, and he has worked for some time with emergency services. His web page can be found via <http://www.orie.cornell.edu>.