An Optimal Algorithm for Search of Extrema of a Bimodal Function

BORIS S. VEROY*

Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, Colorado 80933-7150

Received May 28, 1986

This paper describes and analyzes an algorithm which computes an interval of length $t$ in which a minimizer (or a maximizer) of a periodical bimodal function $h$ is located using a minimal number of evaluations of the function $h$. A dynamic programming approach is used in order to demonstrate the optimality of the algorithm. © 1986 Academic Press, Inc.

INTRODUCTION

The minimax search strategy for a unimodal function was described by Kiefer (1953) and analyzed by Kiefer (1957, 1959), Johnson (1957), Bellman and Dreyfus (1962), Beamer and Wilde (1970), and Weymark and Strongin (1966). Fibonacci search and the method of golden division were extended to many variables by Krolak and Cooper (1963), Newman (1965), Kuzovkin (1968), Shmuel (1972), and Korotchenko (1981). Different randomized algorithms for multieextremal functions were provided by Shubert (1972), Gilinskas (1976), Timonov (1977), and Hill (1969), and many others. Alphanumeric search algorithms were analyzed by Knuth (1973). Different search algorithms were described by Wilde (1964). One can find more about the general theory of optimal algorithms in Traub and Woźniakowski (1980). In this paper we provide a deterministic (nonrandomized) search strategy which is optimal in the minimax sense (formal definition follows below) for a periodic bimodal function. The preliminary results were presented at the National ACM-82 Conference in Indianapolis, and were described by Veroy (1984). Examples of several applications of a search for an extremum of a periodic bimodal function are provided by Veroy (1986a, b).

DEFINITION 1. Let us consider a function $h$ on an interval $(a, b)$, and let

*Current address: Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, NJ 07102.
the function $h$ have one minimum $h(r)$ and one maximum $h(s)$ on this interval, where we call $r$ and $s$ a minimizer and a maximizer, respectively (see Fig. 1). For the minimizer $r$ we assume that there is a finite interval $[r - y, r + y]$ such that for every $x_1$ and $x_2$, $r - y < x_1 < x_2 < r$ implies $h(x_1) > h(x_2)$, and $r < x_3 < x_4 < r + y$ implies $h(x_3) < h(x_4)$ for every $x_3$ and $x_4$. The maximizer is defined analogously. A function with these properties is called bimodal, on the interval $(a, b)$.

DEFINITION 2. Function $h$ defined on $\mathbb{R}$ is a periodic bimodal function if $h(x) = h(x + Pn)$, $n = 1, 2, \ldots$, $P = b - a$, and $h$ is bimodal on the interval $(a, b)$.

The goal of this paper is to describe and analyze an algorithm which computes an interval of length $t$ in which a minimizer (or a maximizer) of a periodic bimodal function $h$ is located using a minimal number of $h$ evaluations in the worst case. We assume that $h$ is computable for every $x$ and does not have a plateau. It is not required that $h$ be smooth or even continuous.

We will describe a search strategy which is minimax in the following sense: Let $H$ be a set of all bimodal functions $h$ of period $P$, $S$, be a set of all possible strategies $s$, to compute an interval of length $t$ in which a minimizer (or a maximizer) of $h$ is located, and $N(H, s)$ be a number of required evaluations to compute the $t$-interval for a function $h$ using a strategy $s_i$. Then for all possible functions, $h$, a minimax strategy $s^*_i$ is such that

$$
\sup_{h \in H} N(h, s^*_i) = \inf_{s_i \in S} \sup_{h \in H} N(h, s_i).
$$

Comment 1. On the interval $[r, r + P]$, $h$ is unimodal and has one maximum $h(s)$ at the point $s$. And on the interval $[s, s + P]$, $h$ has one minimizer $r$. Therefore, if $s$ is known, then $h$ can be treated on $[s + kP, s + (k + 1)P]$ as a unimodal function, where $k$ is an integer.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1.png}
\caption{Periodic bimodal function.}
\end{figure}
It is known from Kiefer (1953, 1957) that the minimax search for a minimizer \( r \) of a unimodal function \( f \) requires only two observations of \( f \) on every step of the algorithm in order to find a subinterval in which \( r \) is located. However, the same problem for a bimodal function requires a slightly more elaborate approach. Indeed, assume we know that \( h(L) < h(R) \). Then from Fig. 2.1, \( r \in (0, L) \); from Fig. 2.2, \( r \in (R, P) \); and \( r \in (L, R) \) from Fig. 2.3.

**Regular Bimodal Function**

**Definition 3.** A regular bimodal function (rbf) is a bimodal function with the following properties:

1. It is either nondecreasing on both ends of the interval and \( h(a) \geq h(b) \), or nonincreasing on both ends and \( h(a) \leq h(b) \);
2. It does not have a plateau on \((a, b)\).

**Lemma 1.** Let \( h \) be a regular bimodal function on an interval \((a, b)\), and let \((p, q)\) be such that \( a \leq p < q \leq b \). Then, \( h \) is either unimodal or regular bimodal on the interval \((p, q)\).

*Proof.* Let \( a < r < s < b \). Then \( h \) is unimodal on \((p, q)\) if either \( q \leq s \) or \( p \geq r \). Otherwise \( h \) is regular bimodal. In the case \( a < s < r < b \), \( h \) is unimodal on \((p, q)\) if \( p \geq s \) or \( q \leq r \).

![Fig. 2. Possible locations of r when h(L) = h(R).](image)
DETECTING THEOREM. Consider the arbitrary points \( m_0 < m_1 < m_2 < m_3 + P \) and \( m_k = m_{k-3} + P \) for \( k \geq 3 \). For a rbf \( h \), let \( h_j = \min(h_2, h_3, h_4) \) and \( h_j = \max(h_2, h_3, h_4) \), where \( h_k = h(m_k) \). Then \( r \in (m_{i-1}, m_{i+1}) \), \( s \in (m_{j-1}, m_{j+1}) \).

Proof. Let us prove that \( r \in (m_{i-1}, m_{i+1}) \). The proof that \( s \in (m_{j-1}, m_{j+1}) \) proceeds identically. Let us assume that the minimizer \( r \in (m_{i-2}, m_{i-1}) \). Then there are four consecutive subintervals \((q_1, q_2) \in (m_{i-2}, r)\), \((q_3, q_4) \in (r, m_{i-1})\), \((q_5, q_6) \in (m_{i-1}, m_i)\), and \((q_7, q_8) \in (m_i, m_{i+1})\) such that the signs of the slopes of \( h \) sequentially alternate on them. However, every bimodal function has no more than three such consecutive subintervals. Thus, \( r \in (m_{i-1}, m_{i+1}) \), since on \((m_{i+1}, m_{i+2})\) \( h \) has the same values as on \((m_{i-2}, m_{i-1})\).

Comment 2. On the interval \((m_{i-1}, m_{i+1})\), \( h \) is either unimodal or is a regular bimodal function (rbf).

In order to develop an optimal search algorithm for a maximizer \( s \) of a rbf \( h \), let us consider the following definitions.

DEFINITION 4. Consider three points \( m_1, m_2, m_3 \) such that \( m_1 < m_2 < m_3 \) and \( m_3 < m_1 + P \), and let \( u = m_2 - m_1 \), \( v = m_3 - m_2 \). Then we will say that a search algorithm is in detecting state \( \{u, v\} \) if \( h_2 < h_1 \) and \( h_2 < h_3 \), where \( h_k = h(m_k) \).

Let us consider a point \( m_4 \in (m_1, m_2) \) or \( m_5 \in (m_2, m_3) \). If \( h(m_4) > h_2 \), then \( \{m_4 - m_1, m_2 - m_4\} \) is a new detecting state, else \( \{m_2 - m_4, v\} \) is a new detecting state. Analogously, if \( h(m_5) > h_2 \), then \( \{m_5 - m_2, m_3 - m_3\} \) is a new detecting state, else \( \{u, m_5 - m_2\} \) is a new detecting state.

It follows from this argument that in all four cases, every additional evaluation of \( h \) decreases the interval of uncertainty. Later we will analyze how to properly select the next \( x \) value for the \( h \) function evaluation, in order to create an optimal algorithm.

DEFINITION 5. \( I_n(u, v) \) is the minimal interval (in the worst case) on which a maximizer \( s \) can be located for \( n \) additional evaluations of the rbf \( h \), if a search starts from the detecting state \( \{u, v\} \).

Then, from the Bellman's principle of optimality (Bellman and Dreyfus, 1962), and from the Detecting Theorem, the relations

\[
I_n(u, v) = \min \left\{ \min_{0\leq x\leq u} \max[I_{n-1}(u-x, x), I_{n-1}(u-x, v)], \min_{0\leq x\leq v} \max[I_{n-1}(u, v-x), I_{n-1}(u, v-x)] \right\}
\]

(2)

describe \( I_n(u, v) \) recursively for any \( n \geq 1 \), and \( I_0(u, v) = u + v \).

Comment 3. Since the next point \( x \) where \( h \) will be evaluated can be
selected either on the interval \( u \) or on the interval \( v \), both branches (2) and (3) provide the best (minimax) choices of \( x \) on the intervals \( u \) and \( v \), respectively.

**Properties of \( I_n(u, v) \)**

1. \( I_n(u, v) = I_n(v, u) \) (symmetricity);
2. \( I_n(u, v) > I_n(u, v) \) if \( m < n \) (effectiveness of search);
3. \( I_n(u', v) \geq I_n(u'', v) \) if \( u' > u'' \) (monotonicity of uncertainty);
4. \( I_n(qu, qv) = qI_n(u, v) \) (homogeneity), \( q > 0 \).

Let us assume for simplicity that \( u \geq v \). Then, from (3) it follows that

(i) \( \max[I_{n-1}(u, v - x), I_{n-1}(v - x, x)] = I_{n-1}(u, u - x) \) for any \( x \) less than \( v \).

(ii) \( \min_{0 < x < u} \max[I_{n-1}(x, u - x), I_{n-1}(u - x, v)] = \min_{0 < x < u} I_{n-1} \times (u - x, \max(x, v)) = \min_{0 < x < u} I_{n-1}(u - x, x) \) since \( I_{n-1}(u - x, \max(x, v)) \) is decreasing on the interval \( 0 < x < v \).

(iii) Finally, let us demonstrate that

\[
\min_{0 < x < v} I_{n-1}(u, v - x) \geq \min_{x \leq x < u} I_{n-1}(u - x, x). \tag{4}
\]

Indeed, \( \min_{0 < x < u} I_{n-1}(u, v - x) = \lim_{x \to 0} I_{n-1}(u, x) \geq \lim_{x \to 0} I_{n-1}(u - x, x) \geq \min_{x \leq x < u} I_{n-1}(u - x, x) \).

**Recursive Definition of \( I_n(u, v) \)**

**Theorem 2.** For \( u \geq v \),

\[
I_n(u, v) = \min_{x \leq x < u} I_{n-1}(u - x, x). \tag{5}
\]

**Proof.** Follows immediately from properties (i) and (ii) and from (4).

**Selection of an Initial Detecting State**

Let an initial interval of uncertainty be \((a, b)\) and let us consider two points, \( m_1 \) and \( m_2 \), such that \( a \leq m_1 < m_2 < b \) or \( a < m_1 < m_2 \leq b \), and let us compute \( h_1 = h(m_1) \) and \( h_2 = h(m_2) \). If \( h_1 < h_2 \), then select \( m_3 \in (m_1, m_1 + P) \), else select \( m_3 \in (m_2 - P, m_2) \). (See Fig. 3.)

In the first case if \( h(m_3) > h_2 \), then a new detecting state is \( \{m_3 - m_2, m_4 - m_3\} \) else it is \( \{m_2 - m_1, m_3 - m_2\} \).

In the second case all considerations are absolutely analogous to those in the previous case.
OPTIMAL DETECTING STATE

Let $c + d \geq u + v$ for all $u$ and $v$ such that $I_n(c, d) = I_n(u, v)$, $c \geq d$, $u \geq v$. In other words, $(c, d)$ is the optimal detecting state with the largest interval of uncertainty $(c + d)$ starting from which the algorithm detects a maximizer of $h$ on the interval of uncertainty with specified length for $n$ additional evaluations of $h$.

THEOREM 3. $I_n(c, d) = I_{n-1}(d, c - d)$.

Proof. $I_n(c, d) = \min_{d \leq x < c} I_{n-1}(x, c - x) = I_{n-1}(q, c - q)$ and let us assume that $q > d$. Now consider $I_n(c, q) = \min_{q \leq x < c} I_{n-1}(x, c - x) = I_{n-1}(q, c - q)$. Hence $I_n(c, d) = I_n(c, q)$, but $c + q > c + d$. Thus $(c, d)$ is not the optimal detecting state. This contradiction proves the theorem.

OPTIMAL STRATEGIES

Let $\{u_k^o, v_k^o\}$ be the optimal detecting state where $k$ is the number of allowed evaluations of $h$.

THEOREM 4. If $I_n(u_k^o, v_k^o) = 1$, then $u_k^o = F_{k+1}$ and $v_k^o = F_k$ for all $k \geq 0$, where $F_0 = 0, F_1 = 1, F_k = F_{k-1} + F_{k-2}$ for all $k > 1$ are Fibonacci numbers.

Proof. $I_0(u_0^o, v_0^o) = u_0^o + v_0^o = 1$. Let $u_0^o = w = 1 - \delta$, $0 < \delta < 1$. Then $v_0^o = w; u_1^o = u_0^o + v_0^o = 1$. It is easy to show by induction that for all $n \geq 1$ (with $F_{-1} = 1$), $u_n^o = F_n + wF_{n-1}, v_n^o = F_{n-1} + wF_{n-2}$, or $u_n^o + v_n^o = F_{n+1} + wF_n$. To maximize $u_n^o + v_n^o$, we select $w = 1 - \delta$, where $\delta \to 0$. Then, asymptotically, $u_n^o = F_{n+1}, v_n^o = F_n$, and $u_n^o + v_n^o = F_{n+2}$. 

Fig. 3. Optimal selection of initial detecting state (searching for maximizer).
Comment 4. It is obvious that a search for both a maximizer \( s \) and a minimizer \( r \) requires at most \( 2n - 2 \) evaluations of \( h \) since the same two initial points, \( m_1 \) and \( m_2 \), can be used at the beginnings of both searches.

**Optimal Search Algorithm**

**Given:** \( P \) is the length of an initial interval of uncertainty, \( P = b - a \); \( t \) is the length of a final interval of uncertainty, \( t \leq P \).

**Goal:** Detect a maximizer \( s \) on the \( t \)-interval using the minimal number of evaluations of a rbf function \( h \).

\[
\begin{align*}
A0: & \quad \text{If } P = t, \text{ then select } s \text{ arbitrarily; stop; } \\
A1: & \quad \text{Find } n > 2 \text{ such that } F_{n-1} < P/t \leq F_n, \text{ where } F_0 = 0, F_1 = 1, \\
& \quad F_k = F_{k-1} + F_{k-2}, k \geq 2; \\
A2: & \quad z = P/tF_n; \\
\end{align*}
\]

Comment 5. \( z \) is a scale of search if \( P/t < F_n \);

\[
\begin{align*}
A3: & \quad \text{Select arbitrarily } m_1 \text{ and } m_2 = m_1 + zF_{n-1}, h_i = h(m_i), i = 1, 2; \\
A4: & \quad \text{If } h_1 > h_2, \text{ then } B = m_2, A = B - zF_n, L = m_1, R = A + zF_{n-1}, \\
& \quad \quad \text{else } A = m_1, B = A + zF_n, R = m_2, L = B - zF_{n-1}; \\
A5: & \quad \text{If } h(L) > h(R), \text{ then } \text{temp} = L, L = 2R - B, B = R, R = \text{temp}; (s \in (A, R)), \\
& \quad \quad \text{else } \text{temp} = R, R = 2L - A, A = L, L = \text{temp}; (s \in (L, B)); \\
\end{align*}
\]

Comment 6. Step A5 does not require storage of all \( F_{n-2}, \ldots, F_1 \) numbers.

\[
\begin{align*}
A6: & \quad \text{Repeat A5 while } B - A > t; \\
A7: & \quad \text{Write } s \in (A, B), \text{ stop.} \\
\end{align*}
\]

Comment 7. The center of \( (A, B) \) is a good approximation for the maximizer \( s \).

**Triad of Related Problems**

Let \( P \) be the length of an initial interval of uncertainty, \( t \) be the length of a final interval of uncertainty, and \( n \) be the total number of evaluations of \( h \) needed to detect a maximizer of rbf functions on a \( t \)-interval. Then

**Problem 1.** Given \( P \) and \( t \), find \( \min n \).

**Problem 2.** Given \( P \) and \( n \), find \( \min t \).

**Problem 3.** Given \( t \) and \( n \), find \( \max P \).
**Main Theorem.**

**Problem 1.** \( \min n = N, \) where \( F_{N-1} < P/t \leq F_N; \)
\( \min n = (\ln P/t)/\ln(\sqrt{5} + 1)/2)(1 + o(t/P)). \)

**Problem 2.** \( \min t = P/F_n. \)

**Problem 3.** \( \max P = tF_n. \)

**Root Oriented Search Is Not Optimal**

It is obvious that on the period \((x*, x* + P)\) there is a unique point \(c\) (a root) such that \(x* < x_1 < c\) and \(c < x_2 < x* + P\) implies either \(h(x_2) > h(x*) > h(x_1)\) or \(h(x_2) < h(x*) < h(x_1)\). In other words, the root \(c\) divides the interval \((x*, x* + P)\) on two separate parts: one on which \(h > h(x*)\) and another where \(h < h(x*)\). The location of the root \(c\) can be found on a \(g\)-interval by means of a binary search using \(\log_2 P/g\) evaluations of \(h\). It is clear that the point \(c\) separates the maximizer and the minimizer on the interval \((x*, x* + P)\). Hence, \(h\) is unimodal on \((x*, c)\) and on \((c, x* + P)\). With a required \(t\)-accuracy, it is sufficient that \(g = t/2\).

Applying the golden division search on every subinterval, we can find the \(t\)-locations of the maximizer \(s\) or the minimizer \(r\) in a time proportional to the value

\[ T = \log_2 M + \log_q M = (\log_q M)(1 + \log_2 q). \]

Here \(q = (\sqrt{5} + 1)/2, M = P/t.\)

To compare the time-complexity of the optimal search algorithm \(A1, \ldots, A7\) with the root oriented search (ROS) consider

\[ (\log_q M)(1 + \log_2 q)/\log_q M = 1 + \log_2 q \approx 1.6945. \]

Therefore, the ROS requires about 70% more time for finding either the maximizer or the minimizer than the optimal search algorithm.

**Optimal Joint Search for Both Extrema**

**Given:**

- \(P\) is the length of an initial interval of uncertainty;
- \(t_r\) is the length of a final interval of uncertainty for the minimizer \(r;\)
- \(t_s\) is the length of a final interval of uncertainty for the maximizer \(s;\)
- \(t = \max(t_r, t_s)\).

**Goal:** Detect both the minimizer \(r\) and the maximizer \(s\) on the \(t\)-intervals using the minimal number \(N_t(P)\) of evaluations in the worst case.
SEARCH OF EXTREMA OF BIMODAL FUNCTION

THEOREM 5. If \( 2F_{n-2} < P/t \leq F_n \), then \( N_t(P) = 2n - 3 \); If \( F_{n-1} < P/t \leq 2F_{n-2} \), then \( N_t(P) = 2n - 4 \).

A proof of this theorem and details of an optimal joint search algorithm are beyond the scope of this paper.

ACKNOWLEDGMENTS

I express my deep appreciation to Dr. Shmuel Winograd of IBM Thomas J. Watson Research Center and to Professor Leonid Levin of Boston University for their valuable comments. My special appreciation is to Professor Henryk Woźniakowski of Columbia University for his constructive comments and numerous suggestions.

REFERENCES


KROLAK, P., AND COOPER, L. (1963), An extension of Fibonaccian search to several variables, Comm. ACM 6, No. 10.

KUZOVKIN, A. I. (1968), Generalization of Fibonaccian search on multidimensional case, Èkonom. i Mat. Metody 4, No. 4.


VEROY, B. S. (1986b), Clustering algorithm for satellite communication network topology, submitted for publication, June.