JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 60 (1995) 357–365

Dynamic programming and the backpacker's linear search problem

Alan Washburn

Operations Research Department, Naval Postgraduate School, Monterey, CA 93943-5000, United States

Received 22 October 1993; revised 3 March 1994

Abstract

A backpacker approaches a road with a marker on it, desirous of finding the marker but having only a rough idea of where it is located. It is well known among backpackers that it is best to aim either right or left of the marker, since otherwise it will not be clear which way to turn upon reaching the road. The problem of deciding exactly where to aim can be formalized as a modification of the Linear Search Problem. This paper does so, and also discusses dynamic programming as a solution method.

Keywords: Search; Dynamic programming

1. Introduction

The Linear Search Problem (LSP), as posed by Bellman [4] and Beck [2], is one where a searcher attempts to locate a random point X in one dimension, starting from a given point and moving at constant speed until the searcher's location first coincides with X at some time T. The problem is to find the searcher path that minimizes $E(T)$, the mean time to find X. Here we consider a modification of the problem where the starting point can be chosen by the searcher. Since the motivation for this is most easily stated in terms of a backpacker approaching a road with a marker on it, we refer to the problem as the backpacker's LSP, or BLSP. Certain search and rescue problems could be modeled as BLSPs, or see [1] for additional applications. The essential assumptions are that:

(1) An object is located on a line according to a known probability distribution.

(2) The searcher starts at such a long distance from the line that his time of arrival on it is essentially independent of the point of intersection; that is, the time and place at which search starts on the line are not related.

(3) Searching is at unit speed and in some sense expensive per unit time, but there are no fuel constraints nor any special time by which the target must be found. These assumptions justify use of $E(T)$ as the quantity to be minimized.

SSDI 0377-0427(94)00038-3

Assumption 3 is important. If minimizing $E(T)$ were replaced by the equally reasonable objective of maximizing the probability of detecting the target by some fixed time, then the optimal track for the searcher would involve at most one turning point instead of the infinite sequence typical of the LSP.

As stated so far, the BLSP is equivalent to Balkhi's "generalized" linear search problem [1], the subject of Sections 2 and 3 following. In Section 4 we will introduce the additional feature that the backpacker may choose to put down his heavy pack while he searches for X. In that form, the BLSP is being considered here for the first time.

2. Analysis

Define a search plan (SP) to be a sequence (x_n) for which $\cdots \leqslant x_4 \leqslant x_2 \leqslant x_0 \leqslant x_1 \leqslant x_3 \leqslant \cdots$, the understanding being that the searcher proceeds at unit speed first from x_0 to x_1 and then to the other points in the order of the index on x, a zig-zag path moving initially to the right. In principle one should also consider search plans that move initially to the left, but we will not do so in order to avoid tedious notational details. This definition is the same as Beck's [2], except that here x_0 is not required to be 0. The search time T is the first time when the searcher's location coincides with X, the location of the target.

Whether an optimal SP exists or not depends on the nature of the distribution of the random variable X. We will assume throughout that X has a bounded density $f(\,)$ with associated cumulative distribution $F(\,)$, that $0 < F(x) < 1$ for all x, and that $E(|X|) < \infty$. In this case an optimal SP exists if x_0 is fixed, as shown in [2, 5]. Let $K(x_0)$ be the minimum expected time to find the target if the searcher is required to start at x_0. Then certainly $|K(x) - K(y)| \leqslant |x - y|$, since the searcher could always move from one point to the other as a first step. Therefore $K(\,)$ is continuous. Also $K(x) \geqslant E(|X - x|)$, since the searcher must cover at least the distance $|X - x|$. Since $|X - x| \geqslant |x| - |X|$, it follows that $\lim_{x \to \infty} K(x) = \lim_{x \to -\infty} K(x) = \infty$, and therefore that the continuous function $K(\,)$ must have a minimum. Thus an optimal SP exists, and the question becomes one of computation. Balkhi [1] has shown the existence of optimal search plans under more general assumptions about $F(\,)$.

The sequence $(x_n;\, n \geqslant 0)$ partitions the real line into intervals, one of which must contain X. Given the interval, there is a simple relation between T and X. If $x_4 < X \leqslant x_2$, for example, then $T = (x_1 - x_0) + (x_1 - x_2) + (x_3 - x_2) + (x_3 - X)$. By accounting for all intervals in which X might lie and taking expectations, it can be shown (as in [8] or [3]) that $E(T) = E(|X - x_0|) - 2x_0F(x_0) + 2\Delta(x)$, where

$$\Delta(x) \equiv \sum_{i=1}^{\infty} x_i H_i, \tag{1}$$

and where

$$H_i = \begin{cases} 1 + F(x_{i-1}) - F(x_i) & \text{if } i \text{ is odd, or} \\ F(x_{i-1}) - F(x_i) - 1 & \text{if } i \text{ is even.} \end{cases}$$

An optimal SP must have the property that $(\mathrm{d}/\mathrm{d}x_i)E(T) = 0$. Since $(\mathrm{d}/\mathrm{d}x_0)E(T) = 2(x_1 - x_0)f(x_0) - 1$, it follows that, for an optimal SP, $x_1 = x_0 + 0.5/f(x_0)$. For $i > 0$, equating

$(\mathrm{d}/\mathrm{d}x_i)E(T)$ to 0 results in the equation $x_{i+1} = x_i - H_i/f(x_i)$. Letting $H_0 \equiv -0.5$, these equations can be written as

$$x_{i+1} = x_i - H_i/f(x_i), \quad i \geqslant 0. \tag{2}$$

As in the case of LSP, first-order optimality conditions determine x except for x_0, since (2) can be used to solve for the x_i sequentially once x_0 is known. Some choice of x_0 must produce an optimal SP. Other choices may produce a sequence that is not an SP (numbers do not diverge from x_0) or which is simply not optimal. Also in the LSP it holds that the optimal SP is the minimal SP that satisfies the analog of (2); larger starting points produce a sequence that diverges too quickly to infinity [3, 8].

3. Solution in the case of the unit normal

Suppose that $F(x) = \Phi(x)$, the standard normal distribution, and let $y_i = (-1)^{i+1}x_i$ for $i \geqslant 0$. Since $\Phi(-x) = 1 - \Phi(x)$ for all x, $H_i = (-1)^i(2 - \Phi(y_i) - \Phi(y_{i-1}))$ for $i > 0$, and therefore (2) becomes

$$y_{i+1} = -y_i + (2 - \Phi(y_i) - \Phi(y_{i-1}))/\phi(y_i), \quad i > 0, \tag{3}$$

with

$$y_1 = -y_0 + 0.5/\phi(y_0) \tag{4}$$

where $\phi(x) \equiv (\mathrm{d}/\mathrm{d}x)\Phi(x)$ is the density function of X.

Eqs. (3) are identical to those of the LSP [8]. The requirement that y_0 be 0 in the LSP is replaced by (4) in the BLSP. Experimentation with these equations using the erf(x, 'high') function of MATLAB [6] produces the SP $(y_i) = (1.571183469417, 2.7352, 3.7259, 4.6046, 5.4040, 6.1435, 6.8357, \ldots)$, with associated $E(T) = 2.1630$. y_0 is given to more decimal places than the other turning points because the sequence produced by (3) and (4) will change if even the last digit of y_0 is altered. Smaller starting points result in the sequence eventually starting to decrease, and larger starting points result in an early increase to MATLAB's "infinity". Rousseeuw [8] reports that $E(T)$ is 2.903665 when the searcher starts at the origin, so the benefit of being able to start at a favorable point in the BLSP is substantial. The probability of finding the target in the interval $[-y_0, y_1]$ is $\Phi(2.7352) - \Phi(-1.5712) = 0.94$, so a searcher using this SP will hardly ever need to turn at y_1.

4. Accounting for the pack

A backpacker has his choice of two methods for getting his pack to the marker at X. Method A is to simply carry his pack until he gets to the marker. Method B is to remove the pack upon first encountering the road, find the marker, return to the pack, and then carry the pack to the known location of the marker. Assuming that the distance traveled is (say) K times as painful when carrying a pack as without one, which method should be used if the object is to minimize total expected pain, and how does the choice of method influence the starting point on the road?

Let P be total pain from first contact with the road until the pack is first located at the marker. Then $P = KT$ for method A, so the minimal value of $E(P)$ is $2.1630K$, and the best search plan for method A is as outlined above. With method B, $P = T + (K + 1)|X - x_0|$, since the distance from X to the pack needs to be traversed twice after X is located, once with the pack and once without. Therefore $E(P) = (K + 2)E(|X - x_0|) - 2x_0F(x_0) + 2\Delta(x)$ for method B, assuming as before that the optimal SP proceeds first to the right. With $F = \Phi$, the first-order conditions for optimality still include (3), but equation $(\mathrm{d}/\mathrm{d}x_0)E(P) = 0$ now leads to the equation

$$y_1 = ((K + 1)(1 - \Phi(x_0)) - 0.5K)/\phi(x_0) - y_0. \tag{5}$$

For each $K \geqslant 1$, the optimal SP can be found as before by experimenting with y_0. For $K = (2, 2.706, 3)$, the best values of y_0 are (0.152632, 0.099808, 0.086843), and the minimal expected pain for method B is (5.2854, 5.8530, 6.0887). When $K = 2.706$ the two methods have the same expected pain so our backpacker should use method B in preference to method A if and only if carrying the pack is more than 2.706 times as painful as not carrying it.

Actually, the best search plan may be neither of these two methods. It may be optimal for the searcher to carry the pack for several nonadjacent intervals of time, putting it down at the end of each interval. It may even be that an optimal search plan does not exist, unlikely as that may seem given known results for the closely related LSP. The option of proving that one exists in order to justify the application of first-order conditions to a problem with no convenient analytic form and many more variables than the LSP is not attractive. In the next section we will shown how dynamic programming can be applied instead.

5. Dynamic programming

In this section we will solve a discrete version of the problem described at the end of Section 4, where the backpacker has the option of carrying the pack at some times but not at others. The marker will be located on the road in accordance with a normal distribution with mean 0 and standard deviation σ. The reason for considering $\sigma \neq 1$ is that the searcher will be restricted in what follows to making turns or changing the status of the backpack only at integer points on the line. By making $\sigma \gg 1$ we can presumably diminish the effect of this integrality assumption.

Let $b(M)$ be a bound on the average amount of travel required to find the marker, given that the marker is known to be outside of the interval $[-M, M]$ and that the searcher is at M. The bound used in all of the calculations reported below is developed in the Appendix. Consider then the finite action space $-M, \dots, M$, where M is some positive integer. The searcher first chooses an integer in this space, then moves back and forth within the action space until either the marker is found or it has been established that the marker is not in $[-M, M]$. In the latter case the searcher must return to the backpack, carry it to the nearest endpoint, and then suffer an additional pain that is $Kb(M)$, on the average, before finally getting the backpack to the marker. All of these restrictions on the searcher's motion have the effect of making $E(P)$ as calculated in this section be an upper bound on what the expected pain would be without the restrictions.

We observe that, at all integer times at which an action is possible, the searcher will have cleared a certain interval $[i, j]$, with the searcher being at one end of the interval and the backpack located somewhere within it. Any point that the searcher has visited has been "cleared". The set of cleared points is always an interval regardless of the searcher's path; this is the basic observation that makes a (reasonably simple) dynamic program possible. The "state" of the situation at decision times is (i, j, k), where j is the location of the searcher, k is the integer location of the backpack, and i is the location of the other end of the cleared interval (we permit $i > j$, in which case the cleared interval is $[j, i]$ and $j \leqslant k \leqslant i$). When the searcher reverses course, he must decide what state to visit next. That decision amounts to selecting i' and k', each at least as far away from j as the corresponding unprimed quantity. In fact, we insist that i' be strictly farther from j than i than i is, since a strategy that reversed course at i and then j and then i again could be improved. If the latter point is not obvious, then take the requirement to be yet another restriction on the searcher's motion. The requirement forces the cleared set to strictly increase with each reversal, and will also permit a dynamic programming recursion that can be solved in one pass.

Let $H(i, j, k)$ be the minimal expected additional pain required to find the target, given that the target is known not to be in the interval $[i, j]$ or $[j, i]$, whichever is appropriate, and that there will be a course reversal at j, the current location of the searcher. Take $H(-M, M, k)$ to be $(K+1)(M-k) + kb(M)$, for $-M \leqslant k \leqslant M$. This assignment corresponds to the idea that the searcher returns to the backpack, brings it to M, and then carries out the search described in the Appendix. Since $H(i, j, k) = H(-i, -j, -k)$ by symmetry, $H(M, -M, -k)$ is also defined. Let $Q(i, j)$ be the a priori probability that X is not in the cleared interval. Then, assuming for clarity that $i \leqslant j$, the desired recursive formula for $H(i, j, k)$ can be obtained by accounting for the two possibilities that X is either in the interval $[i', i]$, in which case the search will terminate, or outside the interval $[i', j]$, in which case the next state will be (j, i', k'):

$$H(i,j,k) = \min_{k' \leqslant k,\, i' < i} \left\{ \int_{i'}^{i} p(u) f(u)\, \mathrm{d}u + [(j - i' + (K-1)(k-k')) + H(j, i', k')] \frac{Q(i',j)}{Q(i,j)} \right\}, \tag{6}$$

where $f(u) = \phi(u/\sigma)/[\sigma Q(i,j)]$ is the density of X, given that X is not in $[i, j]$, and $p(u)$ is the pain in getting from j to u and possibly returning to k' to retrieve the pack:

$$p(u) = \begin{cases} (j-k) + K(k-k') + (2+K)(k'-u) & \text{if } u \leqslant k', \\ (j-k) + K(k-u) & \text{if } u \geqslant k'. \end{cases} \tag{7}$$

Letting $h(i,j,k) \equiv H(i,j,k)Q(i,j)$, (6) can be expressed somewhat more compactly as

$$h(i,j,k) = \min_{k' \leqslant k,\, i' < i} \left\{ \int_{i'}^{i} p(u)[\phi(u/\sigma)/\sigma]\, \mathrm{d}u + Q(i',j)[j - i + (K-1)(k-k')] + h(j,i',k') \right\}. \tag{8}$$

Notice that the h-evaluations on the right-hand side of (8) have arguments such that $|i' - j| > |i - j|$. Thus the computations of h can be organized so that the size of the cleared set is reduced in stages from $2M$ to $2M-1$, etc., until finally $h(i, i, i)$ is computed for $-M \leqslant i \leqslant M$ in the final stage where the size of the cleared set is 0. In the final stage (and only in the final stage) the direction of movement cannot be obtained by comparing the first two arguments of h. In an unsymmetric situation both directions should be tried. The computations reported below simply utilize (8), which amounts to requiring that the searcher first travel to the left.

The integrals involved in (8) can be simplified by observing that

$$\int_a^b u[\phi(u/\sigma)/\sigma]\,\mathrm{d}u = E(a) - E(b) \tag{9}$$

where $E(x) = \sigma\phi(x/\sigma)$.

6. Results of the dynamic program

$M = 20$ and $\sigma = 5$ were used in all cases, with small (1 or 2 seconds on the Naval Postgraduate School mainframe) associated computation times.

The pain factor K was first set to 1 to confirm the computations reported in Section 3. The smallest total pain was 2.1641σ, achieved by always carrying the pack and making turns at 1.6σ, -2.0σ, 4.0σ, and -4.0σ, followed by the procedure described in the Appendix if necessary. These numbers are close enough to those of Section 3 to be consistent with the idea that the FORTRAN implementation was in fact written correctly, as well as to confirm that not too much has been given away in making restrictions on the searcher's path.

The pain factor K was next set to 2.706, the "swing factor" of Section 4. The smallest total pain was 5.1357σ, with the optimal turn schedule being $(-0.4\sigma, -1.6\sigma, 3\sigma, -4\sigma, 4\sigma)$ and the backpack schedule being $(-0.4\sigma, -0.4\sigma, 1.4\sigma, -2.8\sigma, 4\sigma)$. Thus the backpacker should go to -0.4σ, leave the backpack there while he goes left to -1.6σ, pick it up and carry it to 1.4σ on his way to 3σ, etc. To be more precise, a backpacker who follows that schedule is guaranteed that the expected pain will not exceed 5.1357σ, that being the smallest value possible with the restrictions that have been imposed. Comparison of these results with those of Section 4 reveals that neither method A nor method B is optimal when $K = 2.706$; the optimal way of searching is more complicated than either of them.

The fact that the best starting point is negative when $K = 2.706$ is somewhat surprising upon first acquaintance, since the searcher is constrained to move left initially. Sufficiently surprising, in fact, to have caused the author to spend considerable time trying to find the bug in the FORTRAN program that produced it. But in fact a searcher who is constrained to first go left has a dilemma. He can aim to the right of the marker, keep the pack on, and hope that his travel will be short because he will first travel through the heart of the normal distribution. Alternatively, he can aim left, take the pack off, explore a large part of the left side of the normal distribution in relative comfort, and then be reasonably sure that the next time he puts the pack on will be the last. The first alternative is best when $K = 1$, but the second turns out to be superior when $K = 2.706$ (or presumably for any larger value).

Fig. 1 shows graphs of $h(i, i, i)$ for $K = 1$, $K = 2$, and $K = 3$. For $K = 1$, the left minimum is slightly higher than the right minimum only because the searcher is constrained to move at least one unit to the left. Starting at -8 (the -1.6σ point), for example, the searcher grudgingly moves to -9 before beginning his favorite tactic of searching to the right. Without the constraint, the left half would be symmetric to the right. When $K = 3$ however, the asymmetry of the left and right halves is not caused by the constraint of moving initially to the left, but rather by different tactics

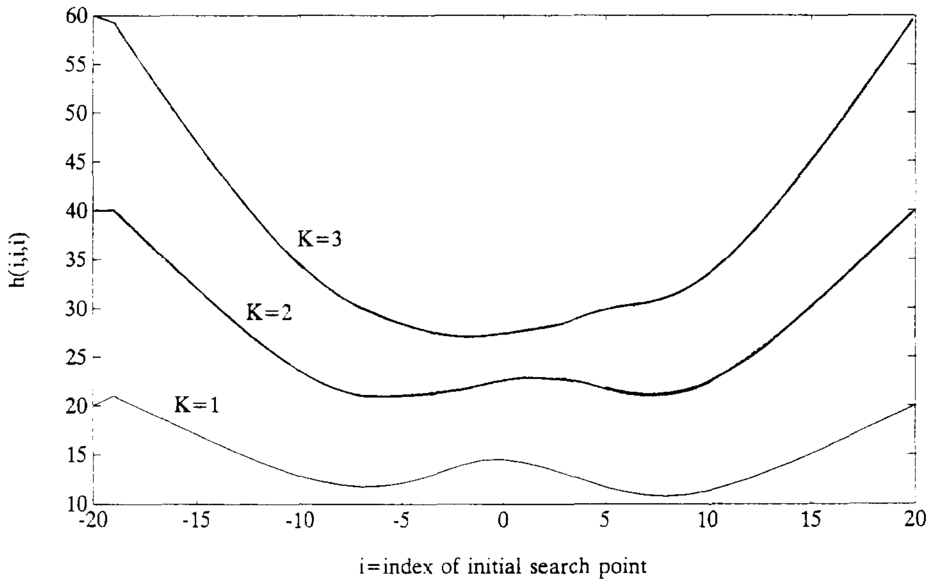

Fig. 1.

for minimization as described in the previous paragraph. When K is smaller than a number that is about 2, the best tactic is to initially retain the pack and move towards the center. When K is larger than that number, the best tactic is to initially put down the pack and move away from the center. The left minimum for $K = 2$ corresponds to the latter tactic, and the right minimum to the former. The left and right minima are nearly equal when $K = 2$.

7. Summary

Dynamic programming is a reasonable competitor to analytic methods for problems such as the LSP and the BLSP. It achieves less accuracy per unit of computational effort, but requires less theorem proving, computes an approximation that is also a bound, and handles multiple local optima with ease. It is adaptable to variations such as backpacks as long as the basic assumption that permits a simple state representation is not disturbed. That assumption in the case of the LSP is that the backpacker cannot possibly pass by the marker without recognizing it, hence the interval of "cleared" points.

It is interesting that dynamic programming could not solve the original "variation" proposed by Bellman [4], at least not as simply. In that variation the searcher only recognizes the marker with some probability $p < 1$, so the state would have to include an interval of points searched once containing an interval of points searched twice containing.... The "curse of dimensionality" would set in, since each additional variable in the state representation has a multiplicative effect on computation time. Solution of that problem by dynamic programming will have to be deferred to a time when computers are bigger and faster.

Appendix

A bound on the travel required in the LSP

A marker was originally located according to a standard normal distribution, but is now known to be outside the interval $[-M, M]$. A searcher is located at M, and must travel at unit speed until he finds the marker at some time T.

Theorem. *For $M > 1$, there is a method of searching for which $E(T) < b(M)$, where*

$$b(M) \equiv 0.5\,\frac{\left(M + \sqrt{M^2 + (8/\pi)}\right)\left(1 + \sqrt{M/(M-1)}\right)}{1 - \exp(-\frac{1}{2}M)} - M.$$

Proof. Let $x_i = \sqrt{M(M+i)}$ for $i \geqslant 0$, an increasing sequence for which $x_0 = M$. The searcher proceeds to $x_1, -x_2, x_3, -x_4, \ldots$, following the SP (x_i). Let $G(x) = 1 - \Phi(x)$, let $c = 0.5/G(M)$, and let $F(x) = c\Phi(x)$ for $x \leqslant -M$, $F(x) = 0.5$ for $|x| \leqslant M$, and $F(x) = 1 - F(-x)$ for $x > M$. $F(x)$ is the CDF of the location of the marker, given that it is not in the interval $[-M, M]$. According to Eq. (1), for this plan

$$E(T) = E(|X - M|) - M + 2\sum_{i=1}^{\infty} x_i(2 - F(x_i) - F(x_{i-1})) \tag{A.1}$$

where the symmetry of F has been utilized in writing the sum. The expected value $E(|X-M|)$ when computed according to the distribution F is $\phi(M)/G(M)$. Since $1 - F(x_i) = F(-x_i) = c\Phi(-x_i) = G(x_i)/(2G(M))$, (A.1) can be written as

$$E(T) = \phi(M)/G(M) - M + \frac{1}{G(M)}\sum_{i=1}^{\infty} x_i(G(x_i) + G(x_{i-1})). \tag{A.2}$$

The ratio $R(x) \equiv G(x)/\phi(x)$ is known as Mill's ratio. It is known [7] that

$$1/(x + \sqrt{x^2 + (8/\pi)}) < R(x) < 1/x \quad \text{for } x > 0. \tag{A.3}$$

Letting S be the sum in (A.2), we therefore have

$$\begin{aligned} S &\leqslant \sum_{i=1}^{\infty} (\phi(x_i) + (x_i/x_{i-1})\phi(x_{i-1})) \\ &= \frac{1}{\sqrt{2\pi}}\sum_{i=1}^{\infty}\left(\exp(-\tfrac{1}{2}M(M+i)) + \sqrt{\frac{M+i}{M+i-1}}\exp(-\tfrac{1}{2}M(M+i-1))\right). \end{aligned} \tag{A.4}$$

Let $z = \exp(-\frac{1}{2}M)$ and observe that $\sqrt{(M+i)/(M+i-1)} \leqslant \sqrt{M/(M-1)}$. Then, from (A.4), after factoring out $\exp(-\frac{1}{2}M^2)$ from each of the exponentials,

$$S \leqslant \phi(M)\sum_{i=1}^{\infty} z^i\left(1 + \sqrt{\frac{M}{M-1}}\exp(\tfrac{1}{2}M)\right). \tag{A.5}$$

The sum in (A.5) is a geometric series, which sums in closed form to give

$$E(T) \leqslant \frac{\phi(M)}{G(M)}\left(1 + \frac{z + \sqrt{M/(M-1)}}{1-z}\right) - M. \tag{A.6}$$

The conclusion follows upon applying the left-hand side of (A.3) to the ratio $\phi(M)/G(M)$. □

It is clear that $E(T) \geqslant M$ no matter what policy the searcher follows, since the marker is located to the left of $-M$ half the time, and in that case the interval $[-M, M]$ of length $2M$ will have to be traversed. Thus $b(M)$ is necessarily larger than M. It is not difficult to show that $\lim_{x \to \infty} b(M) - M = 0.5$, so the bound is sharp when M is large.

References

[1] Z. Balkhi, The generalized linear search problem, existence of optimal search paths, *J. Oper. Res. Soc. Japan* **30** (1987) 399–421.
[2] A. Beck, On the linear search problem, *Israel J. Math.* **2** (1964) 221–228.
[3] A. Beck and M. Beck, Son of the linear search problem, *Israel J. Math.* **48** (1984) 109–122.
[4] R. Bellman, Research problem no. 63-9, *SIAM Rev.* **5** (1963) 274.
[5] W. Franck, An optimal search problem, *SIAM Rev.* 7 (1965) 503–512.
[6] Mathworks, Inc., MATLAB users guide, Natick, MA, 1990.
[7] H.O. Pollack, A remark on "elementary inequalities for Mills' ratio", by Yusaku Komatu, *Reports of Statistical Application Research, Union of Japanese Scientists and Engineers* **4** (1956) 40.
[8] P.J. Rousseeuw, Optimal search paths for random variables, *J. Comput. Appl. Math.* **9** (1983) 279–286.