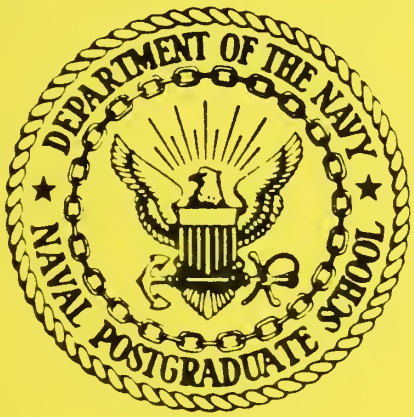
NPS55-85-015

NAVAL POSTGRADUATE SCHOOL

Monterey, California



AN APPROXIMATE SOLUTION TECHNIQUE FOR THE CONSTRAINED SEARCH PATH MOVING TARGET SEARCH PROBLEM

by

James N. Eagle
James R. Yee

October 1985

Prepared for:
Naval Postgraduate School
Monterey, CA 93943-5100

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. H. Shumaker
Superintendent

David A. Schrady
Provost

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPS55-85-015
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE *(and Subtitle)*: AN APPROXIMATE SOLUTION TECHNIQUE FOR THE CONSTRAINED SEARCH PATH MOVING TARGET SEARCH PROBLEM

5. TYPE OF REPORT & PERIOD COVERED: Technical

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s):

James N. Eagle
James R. Yee

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:

Naval Postgraduate School
Monterey, CA 93943-5100

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS:

Naval Postgraduate School
Monterey, CA 93943-5100

12. REPORT DATE: Jul 1985

13. NUMBER OF PAGES: 18

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*:

15. SECURITY CLASS. *(of this report)*: Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

search, moving target, constrained search path, nonlinear programming, convex simplex method

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A search is conducted for a target moving in discrete time among a finite number of cells according to a known Markov process. The searcher must choose one cell in which to search in each time period. The set of cells from which he can choose is a function of the cell chosen in the previous time period. The problem is to find a searcher path, i.e., a sequence of search cells, that minimizes the probability of not detecting the target in a fixed number of time periods. The problem is formulated as a nonlinear program and solved for a local optimum by a simple implementation of the convex simplex method.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

An Approximate Solution Technique for the Constrained Search Path Moving Target Search Problem

James N. Eagle
James R. Yee
Department of Operations Research
Naval Postgraduate School
Monterey, CA 93943

A discrete time search is conducted for a target moving among a finite set of cells $C = \{1,\ldots,N\}$. At the beginning of each time period one cell is searched. If cell i was searched in the previous time period, the current search cell must be selected from the set $C_i \subseteq C$. If the target is in the selected cell k, it is detected (i.e., found) with probability $q_k \in [0,1]$. If the target is not in the cell searched, it can not be detected during the current time period. After an unsuccessful search, a target in cell i moves to cell j with probability γ_{ij} for the next time period. The transition probability matrix $\Gamma = [\gamma_{ij}]$ and the initial distribution of the target over the search cells are known to the searcher. The objective of the searcher is to select a T-time period search path which minimizes the probability of nondetection.

1. Background

The path constrained search problem, described above, is a difficult one to solve efficiently. Trummel and Weisinger [1985] showed that the path constrained search problem with a stationary target is NP-complete. The moving target problem, which is a

generalization of the stationary target problem, is then also NP-complete.

Other than total enumeration of all search paths, the only optimal solution technique mentioned in the literature for the moving target constrained search problem has been the dynamic programming procedure of Eagle[1984a]. Although this method can solve problems much more quickly than total enumeration, it can require a large amount of computer storage as problem size increases.

It was the difficulty experienced with solving large problems optimally that motivated the development of good suboptimal solution procedures. The first such method proposed was a modified branch-and-bound method by Stewart [1979]. Stewart used a discrete version of a moving target search algorithm given by Brown [1980] to provide bounds for his procedure. However, Brown's algorithm does not necessarily give optimal solutions when search effort is discrete, so these "bounds" may result in an optimal branch of the enumeration tree being mistakenly fathomed. Nonetheless, Stewart's computational experience with 1-dimensional search problems indicates that the method can perform well.

Another approximate procedure was given by Eagle [1984b]. This dynamic programming method uses a moving or "rolling" time horizon that greatly reduces the computer storage requirements. It was used to approximately solve a small 2-dimensional problem (3 by 3 search grid) for 40 time periods. This procedure generalizes myopic search by selecting in each time period the next cell to be searched under the assumption that the search ends m time periods in the future.

For myopic search, m is 1. For small enough m, this procedure can be implemented on a microcomputer.

Reported here is a third suboptimal solution method which, like Brown's algorithm, is derived from a nonlinear programming formulation of the search problem. Unlike the problem addressed by Brown, however, this formulation (a) allows for path constraints to be specified for the searcher, (b) does not allow search effort in each time period to be infinitely divisible over the search cells, and (c) does not have a convex detection function. Consequently the objective function of the nonlinear program is not necessarily convex and the solutions obtained may be local rather than global optima. But like Brown's method, the structure of the problem allows a simple implementation of the nonlinear programming solution technique. When considered without path constraints, this procedure has similarities to discrete versions of both Brown's algorithm and those of Washburn [1980] and [1983].

2. Definitions

The movement of the searcher is described by a nonhomogeneous Markov process. Let $S_{ij}(t)$ be the probability that the searcher will search cell j in the time period t, given that cell i was searched in time period t-1. Then a search plan, $S = \{S(1),\ldots,S(T)\}$, is a sequence of T NxN stochastic matrices satisfying

$$\sum_{j \in C_i} S_{ij}(t) = 1, \qquad i = 1,\ldots,N;\ t = 1,\ldots,T.$$

A deterministic search plan is a $\mathbf{S}$ composed entirely of ones and zeros.

$p_{ij}(t)$ is the joint probability that, after the search and target transition in time period t, the searcher is in cell i and the target is in cell j and has not been detected by the first t searches.

$p(0) \in \mathbb{R}^{N \times N}$ is the initial joint searcher-target distribution and is assumed to be known by the searcher.

Note that $\Sigma_{i,j}\, p_{ij}(t)$ is the probability that the target has not been detected by the searches conducted in time periods 1 through t. Also, $p_{ij}(t)$ can be calculated recursively from $p_{ij}(t-1)$ by conditioning on the searcher cell and the target cell after the search and target transition in time period (t-1). Specifically,

$$p_{ij}(t) = (1-q_i \delta_{ij}) \sum_{k,l} p_{kl}(t-1)\, s_{ki}(t)\, \gamma_{i,j} , \qquad (1)$$

where δ_{ij} is 1 if i=j and 0 if i≠j.

3. The Search Problem as a Nonlinear Program (NLP)

We seek the solution of the following NLP

$$\min \sum_{i,j} p_{ij}(T) \qquad (2)$$

subject to

$$\sum_{j \in C_i} s_{ij}(t) = 1, \quad i = 1,\ldots,N;\ t = 1,\ldots,T \qquad (3)$$

$$s_{ij}(t) = 0, \quad i = 1,\ldots,N;\ j \notin C_i;\ t = 1,\ldots,T \qquad (4)$$

$$S(t) \geq 0, \quad t = 1,\ldots,T \qquad (5)$$

where $p_{ij}(T)$ is calculated recursively from (1). The decision variables are the NxN matrices $S(1),\ldots,S(T)$. To solve this problem, $p(0)$, $(q_1,\ldots,q_N)$, and Γ must be specified.

The four propositions which follow establish certain properties of the above nonlinear program. It is noted that this problem, while having a relatively complicated objective function, has constraints which are linear and highly structured.

Proposition 1: The minimum of nonlinear program (2)-(5) is achieved by a deterministic search plan.

Proof: Let S^* be an optimal search plan. (Such a plan exists since the objective function (2) is continuous in S and the set of feasible S defined by (3)-(5) is compact.) S^* defines a T-time period, N-cell Markov process defining probabilisticly an optimal search path. Let $\{\xi_1,\ldots,\xi_Z\}$ be the finite set of possible deterministic search paths generated by S^*. Each ξ_k has an associated $a_k \in \mathbb{R}^{N\times 1}$ where the j^{th} component of a_k is the probability of nondetection given the searcher follows ξ_k and the target starts in cell j. Also let

$$\eta_j = \sum_i p_{ij}(0)$$

be the probability that the target starts in cell j. Then the probability of nondetection given ξ_k is followed and an initial target distribution of $\eta\in\mathbb{R}^{1\times N}$ is the dot product ηa_k. Now if $P(\xi_k)$ is the probability of the searcher following path ξ_k when S^* is used, we have

$$\text{Prob}\{\text{nondetection} \mid \mathbf{S}^*, \eta\} = \sum_{k=1}^{Z} P(\xi_k)\ \eta a_k$$

$$\geq \eta a^*,$$

where

$$a^* = \text{argmin}_{a_k}\ (\eta a_k).$$

So the deterministic search path associated with a^* (and the deterministic search plan which generated it) is also optimal.

Proposition 2: $\mathbf{S}$ is a deterministic search plan if and only if $\mathbf{S}$ is an extreme point of the linear constraints (3)-(5).

Proof: If $\mathbf{S}$ is deterministic, then by definition all component matrices of $\mathbf{S}$ must be composed entirely of zeros and ones. Such an $\mathbf{S}$ can not be written as a <u>strict</u> convex combination of two other matrix series satisfying (3)-(5). So $\mathbf{S}$ is an extreme point. And if $\mathbf{S}$ is not deterministic, then some component matrix must contain a row vector with two or more components strictly between 0 and 1. This row vector can then be expressed as a strict convex combination of two other distinct and feasible row vectors. Thus $\mathbf{S}$ is not an extreme point.

Proposition 3: The objective function (2) is linear in $\mathbf{S}$ when constrained to any edge of the simplex formed by (3)-(5).

Proof: Let $\mathbf{S}'$ and $\mathbf{S}''$ be any two adjacent extreme point solutions of the constraints (3)-(5). Specifically, $\mathbf{S}'$ and $\mathbf{S}''$ are

identical stochastic (0,1) matrix series except in one row of one matrix. Let t be the time period where $\mathbf{S}'$ and $\mathbf{S}''$ differ, and for any $\lambda \in [0,1]$ let

$$\mathbf{S}^{\lambda} = \lambda\mathbf{S}' + (1-\lambda)\mathbf{S}''$$
$$= \mathbf{S}(1),\ldots,\mathbf{S}(t-1),\lambda\mathbf{S}'(t)+(1-\lambda)\mathbf{S}''(t),\mathbf{S}(t+1),\ldots,\mathbf{S}(T)$$

Conditioning on the searcher's cell and the target's cell before the search in time period t, we can write the objective function (2) evaluated at $\mathbf{S}^{\lambda}$ as

$$\sum_{i,j} \sum_{k,l} p_{ij}(t-1)\,[(1-q_k\delta_{kl})\,\mathbf{S}^{\lambda}_{ik}(t)\gamma_{jl}]\,Q_{kl}(t+1), \qquad (6)$$

where δ_{kl} is as in (1), and $Q_{kl}(t+1)$ is the probability that the target is undetected by searches in time periods $t+1,\ldots,T$ given the searcher is in cell k and target is in cell l at the end of time period t. Writing

$$\mathbf{S}^{\lambda}_{ik}(t) = \lambda\mathbf{S}'_{ik}(t) + (1-\lambda)\mathbf{S}''_{ik}(t),$$

and observing that $p_{ij}(t-1)$ and $Q_{kl}(t+1)$ are not functions of λ, shows (6) to be linear in λ.

Proposition 4: All basic feasible solutions to the nonlinear program (2)-(5) are nondegenerate.

Proof: The constraint matrix represented by (3) and (4) consists of NT linearly independent rows. So basic solutions will always have NT basic variables. We argue that these basic variables will always be positive. Each constraint consists of nonnegative variables summing to unity. So any feasible solution must involve at least one variable from each constraint. Since there are NT constraints and NT basic variables, any basic solution

must involve a single basic variable in each constraint. And furthermore, since the constraints sum to unity, the value of each basic variable must be 1.

4. Applying the Convex Simplex Method (CSM)

The CSM, which is a generalization of the simplex method, is applied to nonlinear programs with linear constraints. If the current feasible solution is an extreme point of the constraints, then the CSM determines the rate of change of the objective function along the edges radiating from that extreme point. The current solution then moves along an edge with the greatest initial rate of improvement. This movement continues until a local optimum is found along that edge or an adjacent extreme point is reached (whichever occurs first). If there is no edge along which the objective function can be improved, then the current extreme point is a Kuhn-Tucker point, and a local optimum has been found. The importance of Propoposition 4 is that it guarantees that an extreme point which does not satisfy the Kuhn-Tucker conditions can always be improved with a single iteration of the CSM.

The CSM is applicable to the nonlinear program (2)-(5) because the constraints are linear. Furthermore, its implementation is especially easy since from Proposition 3 the objective function is linear along edges of the simplex. Thus once movement along an edge is determined to reduce the objective function, this movement continues until an adjacent extreme point is reached. That is, no one-dimensional searches for local minima along extreme edges are required.

The CSM can thus be implemented for this problem by starting at any feasible extreme point, evaluating the objective function at all adjacent extreme points, and moving directly to the adjacent extreme point with the smallest objective function value.

The calculations required to evaluate the objective function (2) at any extreme point of the constraints are simplified when the searcher's starting cell is known with certainty (the usual case for most search applications). This occurs because the searcher's starting cell, together with the deterministic search plan associated with an extreme point, uniquely determine a search path $s = (s(1),\ldots,s(T))$, where $s(t)$ is the searcher's cell in time period t. And the probability of nondetection given s is

$$PND(s) = \eta \left(\prod_{t=1}^{T} \Gamma_{s(t)}\right)\mathbf{1} \qquad (7)$$

where η is the initial target distribution, $\mathbf{1}$ is a column vector of ones, and $\Gamma_{s(t)}$ is the target transition matrix Γ with row $s(t)$ multiplied by $(1-q_{s(t)})$.

It is also observed that, although a general search plan S is an element of $\mathbb{R}^{T\times N\times N}$, a deterministic search plan can be represented by a TxN integer matrix $\overline{S}$. Each element $\overline{S}(t,j)$ is the searcher's cell in time period t given the searcher's cell in time period t-1 was j. $\overline{S}$ represents a feasible search plan if and only if each $\overline{S}(t,j)$ is an element of C_j.

The CSM, specifically tailored for the search problem in (2)-(5), is the following:

1. Specify s(1) and an initial feasible $\overline{S}$.
2. Determine s = (s(1),...,s(T)) from s(1) and $\overline{S}$.
3. Calculate PND(s) from (7) and set PND_{min} = PND(s).
4. For t=1 to T,
   a. Vary $\overline{S}(t,s(t))$ over all $C_{s(t)}$ to generate candidate adjacent extreme points $\overline{S}'$.
   b. For each $\overline{S}'$, calculate the new search path s' and PND(s').
   c. If PND(s') < PND_{min}, save $\overline{S}'$ and s', and set PND_{min} = PND(s').
   d. Continue to next t.
5. If no improvement is achieved in PND_{min} after t=T, then STOP. Otherwise define $\overline{S}$ as $\overline{S}'$, s as s', and return to step 4.

Note that each main iteration of this algorithm (i.e.,step 4.) requires the calculation of PND for $\Sigma_i |C_i|$ search paths, where $|C_i|$ is the number of cells in C_i.

5. The Starting Solution

In this section, a procedure for generating an initial feasible $\overline{S}$ is described. A reasonable $\overline{S}$ would be one that corresponds to a myopic search path. A myopic search path is a path where the searcher moves from the cell searched in time period t to that accessible cell j with the largest $q_j \eta_j(t)$, where

$$\eta_j(t) = \Sigma_i \, p_{ij}(t)$$

$$= \text{Prob}\{\text{target in cell } j \text{ and has not been detected by searches in time periods } 1,\ldots,t\}.$$

If two or more accessible cells have identical maximal values of $q_j \eta_j(t)$, then the myopic policy, as used here, attempts to break the tie by selecting a cell with the minimum Euclidean distance (or some other reasonable norm) to a cell with the maximum value of $q_i\eta_i(t)$. If a tie still exists, it is broken randomly. When the target is distant, this tie breaking procedure attempts to generally move the searcher towards a favorable cell and, hopefully, into position for a future detection.

A procedure is still needed to complete the initial feasible $\overline{S}$. One procedure would be to assume that the target distribution is a stationary distribution generated by the target transition matrix Γ. (That is, the target distribution is $\eta \in \mathbb{R}^{1\times N}$ where $\eta = \eta\Gamma$, $\eta_i \geq 0$.) Then for each time period (i.e., each row of $\overline{S}$) the myopic policy is used to determine the next cell to be searched. This results in an $\overline{S}$ which, except for one element in each row, has identical rows. The element which is different

corresponds to the initial myopic search path. This starting solution was tested and, for the problems examined, appeared to work well.

6. Three Examples

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure 1. 25 cell search grid.

The target and searcher move among the 25 cells of Figure 1. In the next time period, the searcher has access to the cell just searched plus all adjacent cells. Cells are adjacent if they share a common side. Thus, for example, $C_1 = \{1,2,6\}$ and $C_3 = \{2,3,4,8\}$.

In each target transition, the target remains in the previously occupied cell j with probability .4 and moves to an adjacent cell with probability $.6/m_j$, where m_j is the number of cells adjacent to cell j.

In the first two examples, the searcher starts in cell 1 and the target in cell 13. Assume that detection is certain if the target's cell is searched. That is, $q_k = 1$, $k = 1,2,\ldots,25$.

A 10-time period problem was solved using FORTRAN 77 on an IBM 3033 mainframe computer. The following results were obtained:

	Search path	Prob. of Nondetection	CPU time (sec)
Myopic	6 7 12 13 14 13 18 19 14 9	.4977	.1
CSM	6 7 8 13 14 19 18 17 12 7	.4886	1.42

In addition, a total enumeration routine was written to find the optimal solutions. This procedure required 25 CPU minutes to examine the 1,225,623 possible 10-time period search paths, and showed the CSM solution to be one of 12 distinct optimal solutions.

In this example the myopic policy was close to optimal. Using the CSM resulted in only a slight improvement, while requiring considerably more CPU time. The next examples show these observations are not true in general.

In the second example, a "fast" target was considered. Specifically, the probability that the target in cell j remained in j was 0, and the probability that a transition occured to any of the m_j adjacent cells was $1/m_j$. All other problem parameters remained the same. This problem gaive the following results:

	Search path	Prob. of Nondetection	CPU time (sec)
Myopic	6 7 12 12 7 8 9 14 19 18	.4540	.1
CSM	6 7 7 8 9 14 19 18 17 12	.3868	1.73

Again, total enumeration showed the CSM solution to be optimal.

The third example examined the possible consequences of a moving mean target position. Here the target started in cell 21 and moved up or right one cell, each with probability .45. The target

remained in its current cell with probability .1. This movement up and right continued until either the top or right boundary was reached. Then with probability .9, the target moved one cell up (if currently in cell 25, 20 15, or 10) or one cell to the right (if in cell 1, 2, 3, or 4). Again, the target remained in the current cell with probability .1. When the target reached cell 5, it remained there forever and was assumed to have escaped. Except in cell 5, detection was certain if the target's cell was searched. In cell 5 the target could not be detected. For the 10-time period problem, the following results were obtained:

	Search path	Prob. of Nondetection	CPU time (sec)
Myopic	6 11 11 12 13 13 14 15 15 10	.2517	.1
CSM	6 11 11 12 13 14 15 15 10 10	.1531	1.09

Total enumeration again showed the CSM path to be optimal. In this final example, the stationary target distribution has the target in cell 5 (the trapping state) with certainty. Since q_5 is 0, the modified myopic procedure given in the previous section fails to give a unique starting $\overline{S}$. To find a starting solution for this example, it was assumed that q_5 was an arbitrarily small, positive number.

Finally it is noted that, like most nonlinear programming solution procedures, using a poor starting solution can result in a poor final solution. In the last example, setting $\overline{S}_{ij} = j$ (except for the one element of each row determined by the myopic path) resulted in a local optimal PND of .2379, a slight improvement over the myopic solution.

Acknowledgments

The authors are grateful to Mat Sagal, LT(jg) Turkish Navy for writing the FORTRAN code requred to solve the examples and generally assisting in all the computational work. Also the comments offered by Professors T.J. Stewart and R.E. Rosenthal were very helpful. This project was funded by the Foundation Research Program of the Naval Postgraduate School, Monterey, CA 93943.

References

BROWN, S.S. 1980. Optimal Search for a Moving Target in Discrete Time and Space. Opns. Res. **28**, 1275-1289.

EAGLE, J.N. 1984a. The Optimal Search for a Moving Target When the Search Path is Constrained. Opns. Res. **32**, 1107-1115.

EAGLE, J.N. 1984b. The Approximate Solution of a Simple Constrained Search Path Moving Target Problem Using Moving Horizon Policies. Naval Postgraduate School Technical Report NPS55-84-009.

STEWART, T.J. 1979. Search for a Moving Target When Searcher Motion is Restricted. Comput. & Ops. Res. **6**, 129-140.

STEWART. T.J. 1980. Experience with a Branch-and-Bound Algorithm for Constrained Searcher Motion. In Search Theory and Applications, K.B. Haley and L.D. Stone (eds.). Plenum Press, New York.

TRUMMEL, K.E. and WEISINGER, J.R. 1984. The Complexity of the Optimal Searcher Path Problem. Forthcoming in Opns. Res.

WASHBURN, A.R. 1980. On Search for a Moving Target. NRLQ, **27**, 315-322.

WASHBURN, A.R. 1983. Search for a Moving Target: The FAB Algorithm. Opns. Res. **31**, 739-751.

DISTRIBUTION LIST

	NO. OF COPIES
Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5100	2
Director of Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943-5100	1
Library, Code 55 Naval Postgraduate School Monterey, CA 93943-5100	1
Commander Submarine Development Squadron Twelve US Naval Submarine Base New London Groton, CT 06340	1
Dr. Mark Mangel Department of Mathematics University of California, Davis Davis, CA 95616	1
Dr. Al Roth Department of Economics University of Pittsburgh Pittsburgh, PA 15260	1
Center for Naval Analyses 2000 Beauregard Street Alexandria, VA 15260	1
Center for Naval Analyses Attn: Dr. Igor Mikolic-Torreira 2000 Beauregard Street Alexandria, VA 22311	1
Professor Loren K. Platzman School of Industrial and Systems Engineering Georgia Institute of Technology Atlanta, GA 30332	1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2