



SRC TR 87-170

**TECHNICAL
RESEARCH
REPORT**

**Search for Randomly Moving
Targets:**

**I: Estimation
II: Search Path Planning**

by

I. Yan and G. L. Blankenship

SYSTEMS RESEARCH CENTER

UNIVERSITY OF MARYLAND

COLLEGE PARK, MARYLAND 20742

SEARCH FOR RANDOMLY MOVING TARGETS I: ESTIMATION¹

I. Yan² and G.L. Blankenship
Systems Research Center and
Electrical Engineering Department
University of Maryland, College Park, MD 20742

ABSTRACT

The detection search problem is the identification of search paths for a specified time interval $[0, T]$, so that the expected number of surviving targets at time T is minimised. The problem can be solved in real time only when the two major procedures: (1) estimation of target posterior distribution; and (2) evaluation of optimal controls (search path planning) based on this posterior target distribution can be done on line. The unnormalised target conditional pdf, $\tilde{p}(t, \zeta | z_0^t)$, satisfies a linear partial differential equation, too complex to be solved on line. Hence, a dual stochastic optimal control problem (in fact, a conventional LQG problem) by using a logarithmic transformation. A special case of the dual problem is studied in detail and a recursive formula is obtained for updating the target's conditional density. The search path planning problem is treated in Part II.

INTRODUCTION

Detection search involves determining searcher routes to maximise the probability of detecting the remaining (surviving) targets. The major tasks of a detection search are:

1. compute a prior distribution of target location;
2. obtain a good estimate of sensor capabilities;
3. determine a detection (mis-detection) function;
4. develop a search plan and estimate its success probability;
5. update the posterior target distribution from search feedback; and
6. evaluate search effectiveness.

Each is nontrivial and can only be solved by tools which compromise between theoretical optimality and operational feasibility. For example, there is no exact way of computing the prior target distribution, especially when targets are hostile. The prior target distribution could be an approximation reflecting a mixture of objective measurements (or calculations) and subjective information, based on experience or heuristics. Subjective information may be used to simplify the problem so as to generate a good search plan in a feasible amount of time.

In [6,7,8,9] detection search problems are treated as "resource allocation problems" in which the search effort is assumed to be infinitely divisible. However, for the "path planning problems" we assume that the searchers are discrete units constrained to follow continuous paths. This work is devoted to the study of procedures 4 and 5 which are called the "search estimation problem" and the "search control problem", treated, respectively, in Part I and Part II of this paper.³

1. SEARCH ESTIMATION PROBLEM CONDITIONAL PDF

Consider a detection search problem with target and searcher dynamics

$$d\zeta(t) = f(t, \zeta(t)) dt + G(t, \zeta(t)) dW(t) \quad (1)$$

$$dx(t) = u(t) dt. \quad (2)$$

where $\zeta(t) \in R^{2N}$, $x(t) \in R^{2M}$ are the target and searcher paths, respectively. N is the number of targets and M is the number of searchers. In equation (1), $f(t, \zeta) \in R^{2N}$ and $G(t, \zeta) \in R^{2N \times L}$ are a vector and a matrix. The initial value $\zeta(0)$ has a pdf $p_0(\zeta)$ which we assume is known. Searcher dynamics (2) are assumed to be deterministic, and the control vector $u(t)$ is the searcher velocity.

Note that we have no observation of the target processes. The only information available are the initial distribution and the results of searching up to time t . The following definitions are made for the single searcher - single target case:

1. detection function, $\psi(t, \zeta; z)$
the probability density function of detection

$$\psi(t, \zeta; z) dt = P\{\text{detecting target during } (t, t+dt) | \zeta(t) = \zeta, x(t) = z\} + O(dt) \quad (3)$$

2. mis-detection function, $\varphi(t, \zeta; z)$
the probability density function of mis-detection

$$1 - \varphi(t, \zeta; z) dt = \psi(t, \zeta; z) dt + O(dt) \quad (4)$$

3. unsuccessful event, x_0^t
the event of no detection when searching along track $x_0^t = \{x(r), 0 \leq r \leq t\}$.

4. conditional pdf, $p(t, \zeta | x_0^t)$
the conditional pdf of $\zeta(t) = \zeta$, conditioned on event x_0^t .

$$p(t, \zeta | x_0^t) d\zeta = P\{\zeta(t) \in [\zeta, \zeta + d\zeta] | x_0^t\} + O(d\zeta)$$

As shown in [2], $p(t, \zeta | x_0^t)$ will be the solution of a stochastic nonlinear integro-partial differential equation:

$$\frac{\partial p(t, \zeta | x_0^t)}{\partial t} = L_t^* p(t, \zeta | x_0^t) - p(t, \zeta | x_0^t) \{ \psi(t, \zeta; x) - \psi(t, x_0^t) \} \quad (5)$$

where:

1. The adjoint differential generator L_t^* is defined as :

$$L_t^*(\cdot) = - \sum_{i=1}^{2N} \frac{\partial (f_i(t, \zeta) \cdot)}{\partial \zeta_i} + \frac{1}{2} \sum_{i,j}^{2N, 2M} \frac{\partial^2 ([G' G']_{i,j} \cdot)}{\partial \zeta_i \partial \zeta_j}. \quad (6)$$

¹Supported in part by NSF grant CDR-85-00108 to the Systems Research Center.

²Current address: SFI, 7833 Walker Drive, Suite 308, Greenbelt, MD 20770.

³Part II will be published elsewhere.

2. The conditional forward detection function $\hat{\psi}(t; x_0^t)$ is defined as:

$$\hat{\psi}(t; x_0^t) = \int_{R^2} \psi(t, \xi; x) p(t, \xi | x_0^t) d\xi. \quad (7)$$

We define

$$\tilde{p}(t, \xi | x_0^t) \equiv \exp\left[-\int_0^t \hat{\psi}(\tau; x_0^\tau) d\tau\right] p(t, \xi | x_0^t), \quad (8)$$

$$p(t, \xi | x_0^t) \equiv \frac{\tilde{p}(t, \xi | x_0^t)}{\int \tilde{p}(t, \eta | x_0^t) d\eta}. \quad (9)$$

I.e., $p(t, \xi | x_0^t)$ and $\tilde{p}(t, \xi | x_0^t)$ are the normalised and unnormalised conditional pdf, respectively. After some manipulation, a simpler equation for $\tilde{p}(t, \xi | x_0^t)$ can be derived: (see [3][5] for details)

$$\frac{\partial \tilde{p}(t, \xi | x_0^t)}{\partial t} = L_t^* \tilde{p}(t, \xi | x_0^t) - \tilde{p}(t, \xi | x_0^t) \psi(t, \xi; x). \quad (10)$$

Equation (10) is an ordinary linear parabolic partial differential equation which generally must be solved numerically.

The components of the forward search equations, (5) and (10), involve a covariance matrix representing target diffusion, a drift vector for target motion, and a detection function characterising searcher detection capability.

2. DETECTION FUNCTIONS

The detection function we use is the "instantaneous detection function" defined by equation (3). Usually it will be homogeneous in time and space. This means that $\psi(t, \xi; x)$ only depends on the distance between targets and searchers. The most commonly used detection functions are listed as below:

Visual Detection ($\frac{1}{x^2}$ detection law)

Assume the searcher is in the sky looking down for targets on the sea, and define:

$$\psi(\xi, x) \equiv \frac{K x_3}{[(\xi_1 - x_1)^2 + (\xi_2 - x_2)^2 + x_3^2]^{\frac{3}{2}}} \quad (11)$$

where $K > 0$ (see Mangel [4] for details).

Radar Detection

In this case, the detection function will be:

$$\psi(\xi, x) \equiv -\frac{1}{S_t} \ln \left[\frac{1}{2} \operatorname{erfc} \left(a - \frac{b}{|\xi - x|^2} \right) \right] \quad (12)$$

where: S_t is radar scan time and erfc is the complementary error function (see [4] for details).

Cookie Cutter

In some cases, the detection function can be approximated by a discrete function such as:

$$\psi(\xi, x) = \begin{cases} \psi_0 & \text{if } |\xi - x|^2 \leq R \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

If $\psi_0 = \frac{1}{\Delta^2}$, we obtain the cookie cutter detection: the probability of detection is one if the distance between searcher and target is less than R and zero otherwise (see [4] for details).

Surface Detection

This is a two dimensional search. In this case the detection function is:

$$\psi(\xi, x) = k \exp \{ -\beta [(\xi_1 - x_1)^2 + (\xi_2 - x_2)^2] \} \quad (14)$$

where: k and β are parameters that depend on the environmental conditions (see [5] for details).

As we can see, detection functions generally depend on the inverse of the distances between searchers and targets.

3. DUALITY OF ESTIMATION AND CONTROL

In this section, a logarithmic transformation is applied to equation (10) to obtain a nonlinear partial differential equation which is the dynamic programming equation of an optimal control problem. Solving the conditional density is equivalent to solving the control problem. This was developed by Fleming and Mitter in [1]. To simplify the presentation, we shall assume

$$\begin{aligned} f(t, \xi) &= f(t) \\ G(t, \xi) &= G(t) \end{aligned} \quad (15)$$

which yields

$$\begin{aligned} \frac{\partial^2 [G(t) G'(t)]_{i,j}(\cdot)}{\partial \xi_i \partial \xi_j} &= [G(t) G'(t)]_{i,j} \frac{\partial^2 (\cdot)}{\partial \xi_i \partial \xi_j} \\ \frac{\partial f_i(t) (\cdot)}{\partial \xi_i} &= f_i(t) \frac{\partial (\cdot)}{\partial \xi_i}. \end{aligned}$$

The PDE satisfied by $\tilde{p}(t, \xi | x_0^t)$ becomes:

$$\left\{ \begin{aligned} \frac{\partial \tilde{p}(t, \xi | x_0^t)}{\partial t} &= \frac{1}{2} \sum_{i,j}^{2N, 2M} [G(t) G'(t)]_{i,j} \frac{\partial^2 \tilde{p}(t, \xi | x_0^t)}{\partial \xi_i \partial \xi_j} \\ &\quad - \sum_{i=1}^{2N} f_i(t) \frac{\partial \tilde{p}(t, \xi | x_0^t)}{\partial \xi_i} - \tilde{p}(t, \xi | x_0^t) \psi(t, \xi; x) \\ \tilde{p}(0, \xi | x_0^0) &= \tilde{p}_0(\xi) \end{aligned} \right\} \quad (16)$$

If we define the logarithmic transformation, Σ , by [2]:

$$\Sigma : \quad V(\tau, \eta; x_0^\tau) \equiv -\ln \tilde{p}(\tau, \eta | x_0^\tau) \quad (17)$$

then by substituting into equation (16), $V(\tau, \eta; x_0^\tau)$ will satisfy:

$$\left\{ \begin{aligned} \frac{\partial V(\tau, \eta; x_0^\tau)}{\partial \tau} &= L_\eta V(\tau, \eta; x_0^\tau) + \psi(\tau, \eta; x) \\ &\quad - \frac{1}{2} \left(\frac{\partial V}{\partial \eta} \right)' G G' \left(\frac{\partial V}{\partial \eta} \right) \\ V_0(\eta) &\equiv V(0, \eta; x_0^0) = -\ln \tilde{p}_0(\eta) \end{aligned} \right\} \quad (18)$$

which is the HJB equation of the following ("dual") stochastic optimal control problem: for $0 \leq \tau \leq t$

$$\left\{ \begin{aligned} d\eta(\tau) &= -f(\tau) d\tau + \bar{u}(\tau) d\tau + G(\tau) dW(\tau) \\ J(\bar{u}_0^t, t, \eta(0), x_0^t) &= E_{x_0^t} \left\{ \int_0^t L(\bar{u}(\tau), t - \tau, \eta(\tau); x(\tau)) d\tau \right. \\ &\quad \left. + V_0(\eta(t)) \right\} \\ L(\bar{u}(\tau), \tau, \eta(\tau); x(\tau)) &= \bar{u}'(\tau) N(\tau) \bar{u}(\tau) + \psi(\tau, \eta(\tau); x(\tau)) \\ N(\tau) &= \frac{1}{2} \{ G(\tau) G'(\tau) \}^{-1} \end{aligned} \right\} \quad (19)$$

where:

1. $f(t), G(t)$ are the same as in equation (15), and $V_0(\eta(0)) = -\ln \tilde{p}_0(\eta)$.
2. $E_{x_0^t}(\cdot)$ denotes the mathematical expectation conditioned on the event that searchers were following the track x_0^t .

Remark 1 The state $\eta(t)$ and control $\bar{u}(t)$ in the dual system (19) are "artificial objects" have nothing to do with any quantity in the real system (2). The only connection is the transformation between the minimum cost function $V(t, \xi; x_0^t)$ and the $\tilde{p}(t, \xi | x_0^t)$:

$$-\ln \tilde{p}(t, \xi | x_0^t) = V(t, \xi; x_0^t) \equiv \min_{\bar{u}_t^t} J(\bar{u}_0^t, t, \xi(t); x_0^t)$$

The dual system is a tool used to solve the PDE of $\tilde{p}(t, \xi | x_0^t)$. Some approximations and simplifications are made in next section, so that the dual problem can be solved analytically.

4. THE LQG FORMULATION

The dual stochastic optimal control problem, formulated in equation (19), can be readily solved ("online") by introducing an approximation based on the general property that detection functions depend on the inverse of the distance between target and searcher. That is, the misdetection function should be proportional to a function of the distance. In most cases, they are superlinear functions. This suggests approximation of the misdetection function by a quadratic function:

$$\varphi(t, \zeta; x) \cong \begin{cases} k \|\zeta - x\|^2 & \text{when } \|\zeta - x\|^2 \leq D \\ c & \text{when } \|\zeta - x\|^2 \geq D \end{cases} \quad (20)$$

$$\cong \varphi_q$$

for some $k > 0$.

Remark 2 The searcher detection capability is inversely proportional to the constant k .

The new evolution of $\tilde{p}(t, \zeta | x_0^t)$ is [10]:

$$\frac{\partial \tilde{p}(t, \zeta; x_0^t)}{\partial t} = L_\zeta^* \tilde{p}(t, \zeta | x_0^t) + \tilde{p}(t, \zeta | x_0^t) \varphi(t, \zeta; x) \quad (21)$$

which is just the result of replacing $\psi(t, \zeta; x)$ by $-\varphi(t, \zeta; x)$ in the original forward search equation. Approximating $\varphi(t, \zeta; x)$ by a quadratic function $k \|\zeta - x\|^2$, permits the dual optimal control problem to be formulated into a linear regulator problem. Hence, for $N = M = 1$, the dual dynamic programming equation will be (if $f(t, \zeta) = f(t)$ and $G(t, \zeta) = G(t)$):

$$\left\{ \begin{array}{l} \frac{\partial V(r, \eta; x_0^r)}{\partial r} = L_\eta V(r, \eta; x_0^r) - k \|\eta - x\|^2 \\ \quad - \frac{1}{2} \left(\frac{\partial V}{\partial \eta} \right)' G G' \left(\frac{\partial V}{\partial \eta} \right) \\ V_0(\eta) \equiv V(0, \eta; x_0^0) \\ \quad = -\ln \tilde{p}_0(\eta) \end{array} \right\} \quad (22)$$

And, the corresponding dual optimal control system is:

$$\left\{ \begin{array}{l} d\eta(r) = -f(r)dr + \bar{u}(r)dr + G(r)dW(r) \\ J(\bar{u}_0^t, t, \eta(0), x_0^t) = E_{x_0^t} \left\{ \int_0^t L(\bar{u}(r), t-r, \eta(r); x(r))dr \right. \\ \quad \left. + V_0(\eta(t)) \right\} \\ L(\bar{u}(r), r, \eta(r); x(r)) = \bar{u}'(r)N(r)\bar{u}(r) - k \|\eta(r) - x(r)\|^2 \\ N(r) = \frac{1}{2} \{ G(r) G'(r) \}^{-1} \end{array} \right\} \quad (23)$$

Remark 3 The dual problem in equation (23) is a linear regulator problem with state $\eta(t)$ minus a known function $x(t)$ appearing in the cost integrand.

5. SOLVING THE DUAL PROBLEM

An algorithm for solving the dual space homogeneous problem is proposed in this section. The time horizon $[0, T]$ is partitioned into small time intervals with length Δt . The dual problem is solved every Δt , and the conditional pdf $\tilde{p}(t, \zeta | x_0^t)$ is updated at every $n\Delta t$. Assume $N = M = 1$,

$$\begin{aligned} d\zeta(t) &= f(t) dt + G(t) dW(t) \in R^2 \\ dx(t) &= u(t) dt \in R^2 \end{aligned}$$

Let

$$N(t) \equiv \frac{1}{2} G(t)G'(t). \quad (24)$$

And, denote

$$t_n = n\Delta t$$

for simplicity. At any time instant $t_n \in [0, T]$, the path planner uses $\tilde{p}(t_n, \zeta | x_0^{t_n})$ as a prior distribution to make the remaining search strategy, for the interval $[t_n, T]$, adapted to $\tilde{p}(t_n, \zeta | x_0^{t_n})$.

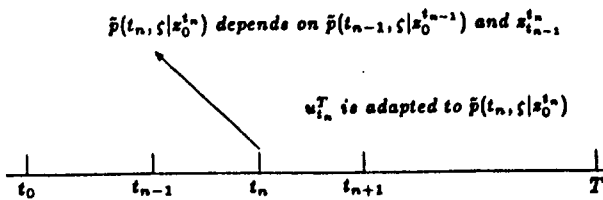


Figure 1: Time horizon partitions.

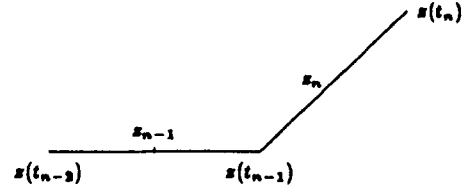


Figure 2: Piecewise linear search.

We assume that:

1. Searcher path is piecewise linear by applying a piecewise step function $u(t)$, with

$$u(t) = u_n, \quad \text{for } t_{n-1} < t \leq t_n$$

2. Also, we approximate the fact that

searchers move from $x(t_{n-1})$ to $x(t_n)$ during $[t_{n-1}, t_n]$.

by event of

searchers stay at the middle of the path for Δt amount of time

If we denote the middle of the path by x_n :

$$x_n \equiv \frac{1}{2} [x(t_{n-1}) + x(t_n)] \quad (25)$$

then

$$dx_n = 0 dr, \quad \text{for } t_{n-1} < r < t_n.$$

3. Define system state X_n to be the distance between the target and the searcher, and the linear transformation, Γ_n , by

$$\Gamma_n : X_n(r) \equiv [\eta(r) - x_n] \in R^2 \quad (26)$$

where

$$\begin{aligned} d\eta(r) &= -f(r)dr + \bar{u}(r)dr + G(r)dW(r), \\ dx_n &= 0 dr. \end{aligned}$$

Therefore, the state evolution can be expressed by

$$dX_n(r) = -f(r)dr + \bar{u}(r) dr + G(r)dW(r). \quad (27)$$

4. Approximate the misdetection function by a quadratic function of new state X_n

$$k \|\eta(r) - x(r)\|^2 \cong k \|\eta(r) - x_n\|^2 \equiv X_n' K X_n$$

with

$$K \equiv \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix}, \quad \text{and } X_n \in R^2.$$

After making assumptions 1 to 4, the dual of the estimation problem is equivalent to the following linear regulator problem P_n' , with path $x_{t_{n-1}}^{t_n}$ suspended (compared to problem of equation (23) in which $x_{t_{n-1}}^{t_n}$ appears explicitly):

$$P_n' : \left\{ \begin{array}{l} dX_n(r) = -f(r)dr + \bar{u}(r)dr + G(r)dW(r) \\ J_n = J_n(\bar{u}_{t_{n-1}}^{t_n}, t_n, X_n(t_{n-1})) \\ \quad = E \left\{ \int_{t_{n-1}}^{t_n} L(\bar{u}(r), t_n-r, X_n(r))dr \right. \\ \quad \quad \left. + V_n(t_{n-1}, X_n(t_n)) \right\} \\ L(\bar{u}(r), r, X_n(r)) = \bar{u}'(r)N(r)\bar{u}(r) - X_n'(r)KX_n(r) \end{array} \right\} \quad (28)$$

To solve problems $\dots P_{n-1}^*, P_n^*, P_{n+1}^*, \dots$ sequentially (i.e., the final condition of the minimum cost function of P_{n-1}^* is the initial condition for the minimum cost function of problem P_n^*), we introduce the following notation:

$$\Delta x_n \equiv x_n - x_{n-1} \in R^2 \quad (29)$$

$\Rightarrow$

$$\begin{aligned} X_n &= \begin{bmatrix} \eta - x_n \\ \eta - x_{n-1} \end{bmatrix} + \begin{bmatrix} x_{n-1} - x_n \\ 0 \end{bmatrix} \\ &= X_{n-1} - \Delta x_n \end{aligned} \quad (30)$$

$\Rightarrow$

$$\begin{aligned} V_n(t_{n-1}, X_n) &\quad (\text{initial condition of } V_n) \\ &= V_{n-1}(t_{n-1}, X_{n-1}) \quad (\text{final condition of } V_{n-1}) \\ &= V_{n-1}(t_{n-1}, X_n + \Delta x_n) \end{aligned}$$

Consequently, the estimation problem at time t_n becomes:

$$P_n'' : \begin{cases} dX_n(r) = -f(r)dr + \bar{u}(r)dr + G(r)dW(r) \\ J_n = J_n(\bar{u}_{n-1}^*, \Delta t, X_n(t_{n-1})) \\ = E \left\{ \int_{t_{n-1}}^{t_n} L(\bar{u}(r), t_n - r, X_n(r)) dr \right. \\ \left. + V_{n-1}(t_{n-1}, X_n(t_n) + \Delta x_n) \right\} \\ L(\bar{u}(r), r, X_n(r)) = \bar{u}'(r)N(r)\bar{u}(r) - X_n'(r)KX_n(r) \end{cases} \quad (31)$$

The problems $\dots P_{n-1}'', P_n'', P_{n+1}'', \dots$ are sequential. The final condition of the P_{n-1}'' minimum cost function gives the initial condition of the P_n'' minimum cost function with a simple linear adjustment term Δx_n .

Note that equation (31) is a linear regulator problem, which implies the solution of its dynamic programming equation has a quadratic form

$$V_n(r, X) = X' Y_n(r, X, \Delta x_n, Y_{n-1}) X \quad (32)$$

where, for $t_{n-1} \leq r \leq t_n$, $Y_n(r, X, \Delta x_n, Y_{n-1})$ satisfies the following matrix Riccati equation E_n :

$$E_n : \begin{cases} \frac{dY_n(r, X, \Delta x_n, Y_{n-1})}{dr} = K + f'(r) Y_n(r, X, \Delta x_n, Y_{n-1}) \\ + Y_n(r, X, \Delta x_n, Y_{n-1}) f(r) \\ + Y_n'(r, X, \Delta x_n, Y_{n-1}) N^{-1}(r) Y_n(r, X, \Delta x_n, Y_{n-1}) \\ \text{with initial condition } Y_n(t_{n-1}, X, \Delta x_n, Y_{n-1}) \\ \text{being determined by the next equation.} \end{cases} \quad (33)$$

$$\begin{aligned} X' Y_n(t_{n-1}, X, \Delta x_n, Y_{n-1}) X \\ = V_n(t_{n-1}, X) \quad (\text{initial condition of } V_n) \\ = V_{n-1}(t_{n-1}, X + \Delta x_n) \quad (\text{final condition of } V_{n-1}) \\ = [X + \Delta x_n]' Y_{n-1}(t_{n-1}, X + \Delta x_n, \Delta x_{n-1}, Y_{n-2}) [X + \Delta x_n] \end{aligned} \quad (34)$$

As we can see, $Y_n(t_n, X, \Delta x_n, Y_{n-1})$ depends on Y_{n-1} , and

$$Y_{n-1}(t_{n-1}, X + \Delta x_n, \Delta x_{n-1}, Y_{n-2})$$

depends on Y_{n-2} , and so on. Eventually, one obtains $Y_n(t_n, X, \Delta x_n, Y_{n-1})$ as a function of

$$[\Delta t, X, \Delta x_1, \Delta x_2, \dots, \Delta x_{n-1}, \Delta x_n, Y_0]$$

or $Y_n(t_n, X, \Delta x_n, Y_{n-1})$ depends on $[\Delta t, \eta, x_n, x_{n-1}, \dots, x_1, x_0, Y_0]$. To simplify the notation, we denote $Y_n(t, X, \Delta x_n, Y_{n-1})$ by $Y_n(t)$.

Remark 4 Note that N is a symmetrical matrix and K is a diagonal matrix with equal diagonal elements. The solution of the matrix Riccati equation

$$\frac{dY_n(r)}{dr} = K + f'(r) Y_n(r) + Y_n(r) f(r) + Y_n'(r) N^{-1}(r) Y_n(r) \quad (35)$$

will also be a diagonal matrix with equal diagonal elements, denoted by

$$Y_n(r) = \begin{bmatrix} y_n(r) & 0 \\ 0 & y_n(r) \end{bmatrix}, \quad (36)$$

and $y_n(r)$ is determined by evolution (35) and initial condition (34).

The conditional pdf of $\eta(t_n)$ can be obtained from the definition of the linear transformation $X_n(r) \equiv \eta(r) - x_n$, which yields

$$\begin{aligned} \tilde{p}(t_n, \eta(t_n) | x_0^0) &= \tilde{p}_n(t_n, \eta(t_n) - x_n) \\ &= \exp \{ -V_n(t_n, \eta(t_n) - x_n) \} \\ &= \exp \{ -[\eta(t_n) - x_n]' Y_n(t_n) [\eta(t_n) - x_n] \}. \end{aligned} \quad (37)$$

A SPECIAL CASE

Solution of the dual problem in the case of $f(t) = 0$ is studied explicitly and a recursive formula for updating $Y_n(r)$ is obtained. Assume $G(r) = I_2$ for simplicity, then

$$N = \frac{1}{2}(G'G) = \frac{1}{2}I_2.$$

The matrix Riccati equation for $f(t) = 0$ becomes

$$\frac{dY_n(r)}{dr} = K + Y_n'(r) N^{-1} Y_n(r)$$

which yields

$$Y_n(r) = \begin{bmatrix} y_n(r) & 0 \\ 0 & y_n(r) \end{bmatrix} = y_n(r) \cdot I_2,$$

and

$$y_n(r) = \sqrt{\frac{k}{2}} \tan(\sqrt{2k} r + A_n) \quad (38)$$

where, by equation (34), A_n is a constant to be determined by the initial condition $Y_n(t_{n-1})$, position X_n and Δx_n .

$$\begin{aligned} X_n' [Y_n(t_{n-1}) \cdot I_2] X_n \\ = [X_n + \Delta x_n]' [Y_{n-1}(t_{n-1}) \cdot I_2] [X_n + \Delta x_n]. \end{aligned} \quad (39)$$

By comparing (38) and (39), one finds that

$$A_n = \tan^{-1} \left[\sqrt{\frac{2}{k}} y_{n-1}(t_{n-1}) \frac{\|X_n + \Delta x_n\|^2}{\|X_n\|^2} \right] - \sqrt{2k} t_{n-1}. \quad (40)$$

If we substitute A_n into (38), then

$$\begin{aligned} y_n(t_n) &= \sqrt{\frac{k}{2}} \tan \left\{ \sqrt{2k} \Delta t \right. \\ &\quad \left. + \tan^{-1} \left[\sqrt{\frac{2}{k}} y_{n-1}(t_{n-1}) \frac{\|X_n + \Delta x_n\|^2}{\|X_n\|^2} \right] \right\}. \end{aligned} \quad (41)$$

Let's choose a set of points $\{Q_1, Q_2, Q_3, \dots, Q_R\}$ from the search domain to evaluate the $\tilde{p}(t_n, Q_r | x_0^0)$ for $1 \leq r \leq R$. Assume $Q_r \in R^2$, and $Y_n(t_n, r)$ denotes the $Y_n(t_n)$ matrix evaluated at point Q_r . The first matrix $Y_0(t_0, r)$ can be determined in the following steps:

1. Assume at time 0, the target has a known distribution evaluated at the given set of points (not necessarily Gaussian)

$$\tilde{p}(Q_r) \equiv \tilde{p}(0, Q_r | x_0^0), \quad \text{for } 1 \leq r \leq R$$

2. From definition (25), we have

$$\begin{aligned} z_0 &= \frac{1}{2} [x(t_{-1}) + x(t_0)] \\ &= \frac{1}{2} [x(0) + x(0)] = x(0) \end{aligned} \quad (42)$$

which is a known constant vector (the initial searcher locations).

3. Then, the planner may choose $Y_0(t_0, r)$ in such a way that

$$[Q_r - z_0]' Y_0(t_0, r) [Q_r - z_0] = -\ln \tilde{p}(0, Q_r | x_0^0)$$

$\Rightarrow$

$$Y_0(t_0, r) = \begin{bmatrix} y_0(t_0, r) & 0 \\ 0 & y_0(t_0, r) \end{bmatrix}$$

and

$$y_0(t_0, r) = \frac{-\ln \tilde{p}(0, Q_r | x_0^0)}{\|Q_r - z_0\|^2}$$

Hence, the $Y_0(t_0, r)$ depends on $[Q_r, z_0]$.

Theorem 1 The evolution of $y_n(t_n, r)$, at point Q_r , can be expressed as:

$$y_n(t_n, r) = \sqrt{\frac{k}{2}} \tan \left\{ \sqrt{2k} \Delta t + \tan^{-1} \left[\sqrt{\frac{2}{k}} y_{n-1}(t_{n-1}, r) \frac{\|Q_r - x_{n-1}\|^2}{\|Q_r - x_n\|^2} \right] \right\}, \quad (43)$$

and

$$\begin{aligned} \tilde{p}(t_n, Q_r | z_0^{t_n}) &= \tilde{P}_n(t_n, Q_r - x_n) \\ &= \exp \{ -y_n(t_n, r) \|Q_r - x_n\|^2 \}. \end{aligned} \quad (44)$$

Proof. [10].

Although the $\tilde{P}_n(t_n, X_n)$ has a Gaussian form (for a given X_n),

$$\begin{aligned} \tilde{P}_n(t_n, X_n) &= \exp \{ -V_n(t_n, X_n) \} \\ &= \exp \{ -X_n' Y_n(t_n) X_n \} \end{aligned} \quad (45)$$

it is not a Gaussian distribution because of the dependence of $Y_n(t_n)$ on X_n as shown in equation (43), where $Y_n(t_n, r)$ is a function of r . I.e., Since $X_n = Q_r - x_n$, $Y_n(t_n, r)$ is actually a function of

$$[n, \Delta t, Q_r, x_n, x_{n-1}, \dots, x_1, x_0, Y_0(t_0, r)].$$

Remark 5 The advantages of solving these cascaded dual linear regulator problems $\dots, P_{n-1}''', P_n''', P_{n+1}''', \dots$ are:

(1) Instead of solving the 2nd order linear partial differential equation

$$\frac{\partial \tilde{p}(t, Q_r | z_0^t)}{\partial t} = L_n^* \tilde{p}(t, Q_r | z_0^t) + \tilde{p}(t, Q_r | z_0^t) \varphi(t, Q_r; z)$$

numerically, we can solve $\tilde{p}(t_n, z | z_0^{t_n})$ analytically, given prior $\tilde{p}(t_{n-1}, z | z_0^{t_{n-1}})$, x_{n-1} and x_n . I.e., equation (43) is an analytic evolution of $Y_n(t_n, r)$ in terms of its previous value $Y_{n-1}(t_{n-1}, r)$ and the incremental searching event $Z_{t_{n-1}}^{t_n}$.

(2) The updating of $\tilde{p}(t_n, Q_r | z_0^{t_n})$ can be done on line.

One observation is made on the evolution of $y_n(t_n, r)$: If $Q_r = x_n$ which means we want to evaluate the $\tilde{p}(t_n, Q_r | z_0^{t_n})$ at the point covered by the searcher during $[t_{n-1}, t_n]$, then $X_n = 0$, and by using (43), we have $y_n(t_n, r) \rightarrow \infty$, which implies

$$\tilde{p}(t_n, Q_r | z_0^{t_n}) \rightarrow 0.$$

ANALYSIS OF THE SOLUTION

Theorem 2 For a detection search system with $f(t) = 0$, suppose we have two search tracks $Z_0^{t_n}$ and $\tilde{Z}_0^{t_n}$ defined by

$$\begin{aligned} Z_0^{t_n} &\equiv \{ Z_0^{t_{n-1}}, x_n \}, \\ \tilde{Z}_0^{t_n} &\equiv \{ Z_0^{t_{n-1}}, \tilde{x}_n \}, \end{aligned}$$

where path $Z_0^{t_n}$ and $\tilde{Z}_0^{t_n}$ are equal up to time t_{n-1} , but have the last search movement different ($x_n \neq \tilde{x}_n$). Assume

- (1) $\|Q_r - x_n\| < \|Q_r - \tilde{x}_n\|$, for some $1 \leq r \leq R$,
- (2) misdetection constant k is small enough (i.e., searcher's detection capability is strong enough),

such that

$$\frac{2 y_{n-1}^2(t_{n-1}, r) \|Q_r - x_{n-1}\|^4}{\|Q_r - x_n\|^2 \|Q_r - \tilde{x}_n\|^2} - k > 0, \quad (46)$$

then

$$\tilde{p}(t_n, Q_r | Z_0^{t_n}) < \tilde{p}(t_n, Q_r | \tilde{Z}_0^{t_n}).$$

Proof. [10].

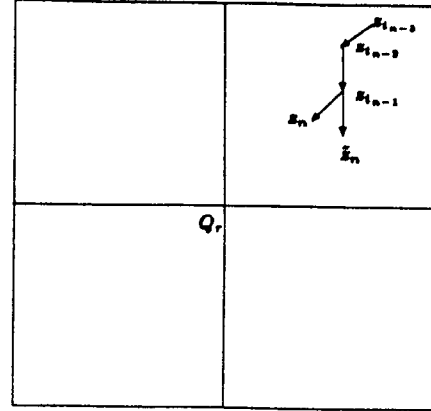


Figure 3: This graph represents a geometrical interpretation of $\|Q_r - \tilde{x}_n\|^2 > \|Q_r - x_n\|^2$.

Theorem 2 compares the densities of the same evaluation point, Q_r , conditioned on two different searching events $Z_0^{t_n}$ and $\tilde{Z}_0^{t_n}$. The only difference between these two events is

$$\|Q_r - \tilde{x}_n\|^2 > \|Q_r - x_n\|^2,$$

as shown in Figure 3. In other words, the searcher moves closer to Q_r in event $Z_0^{t_n}$ than in event $\tilde{Z}_0^{t_n}$ at time t_n . Then we can prove

$$\tilde{p}(t_n, Q_r | Z_0^{t_n}) < \tilde{p}(t_n, Q_r | \tilde{Z}_0^{t_n})$$

which agrees with intuition.

Theorem 3 For a detection search system with $f(t) = 0$, if

- (1) $\tilde{p}(t_{n-1}, Q_r | Z_0^{t_{n-1}}) = \tilde{p}(t_{n-1}, Q_r | \tilde{Z}_0^{t_{n-1}})$,
- (2) $\|Q_r - x_n\|^2 > \|Q_r - \tilde{x}_n\|^2$, and
- (3) misdetection constant k is small enough (or searcher's detection capability is strong enough), such that $k - \frac{2 y_{n-1}^2(t_{n-1}, r) \|Q_r - x_{n-1}\|^4}{\|Q_r - x_n\|^2 \|Q_r - \tilde{x}_n\|^2} < 0$,

and the event $Z_0^{t_n}$ is defined by:

$$Z_0^{t_n} \equiv \{ Z_0^{t_{n-1}}, x_n \},$$

then

$$\tilde{p}(t_n, Q_r | Z_0^{t_n}) > \tilde{p}(t_n, Q_r | \tilde{Z}_0^{t_n}).$$

Proof. [10].

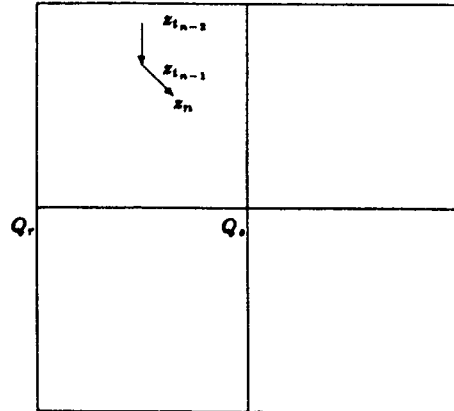


Figure 4: This graph represents a geometrical interpretation of $\|Q_r - z_n\|^2 > \|Q_s - z_n\|^2$.

Remark 6 Theorem 3 compares the conditional densities of two different evaluation points Q_r and Q_s , conditioned on the same event $Z_0^{t_n}$. Assumption (1) means

$$\bar{p}(t_{n-1}, Q_r | Z_0^{t_{n-1}}) = \bar{p}(t_{n-1}, Q_s | Z_0^{t_{n-1}}).$$

Assumption (2) means that the searcher moves closer to point Q_s than Q_r at time t_n as shown in Figure 4. Then we have the inequality

$$\bar{p}(t_n, Q_r | Z_0^{t_n}) > \bar{p}(t_n, Q_s | Z_0^{t_n}).$$

Remark 7 From Theorem 2 and Theorem 3, if we define density differences (see [10])

$$\begin{aligned} D(r, r, Z_0^{t_n}, \tilde{Z}_0^{t_n}) &\equiv -\ln \bar{p}(t_n, Q_r | Z_0^{t_n}) + \ln \bar{p}(t_n, Q_r | \tilde{Z}_0^{t_n}) \\ &= \Delta t \left[\|Q_r - \tilde{z}_n\|^2 - \|Q_r - z_n\|^2 \right] \\ &\quad \left[k - \frac{2 \nu_{n-1}^2(t_{n-1}, r) \|Q_r - z_{n-1}\|^4}{\|Q_r - z_n\|^2 \|Q_r - \tilde{z}_n\|^2} - k \right] \\ &\quad + \text{higher order terms.} \\ &> 0. \end{aligned} \quad (47)$$

$$\begin{aligned} D(r, s, Z_0^{t_n}, \tilde{Z}_0^{t_n}) &\equiv -\ln \bar{p}(t_n, Q_r | Z_0^{t_n}) + \ln \bar{p}(t_n, Q_s | \tilde{Z}_0^{t_n}) \\ &= \Delta t \left[\|Q_r - z_n\|^2 - \|Q_s - z_n\|^2 \right] \\ &\quad \left[k - \frac{2 \nu_{n-1}^2(t_{n-1}, r) \|Q_r - z_{n-1}\|^4}{\|Q_r - z_n\|^2 \|Q_s - z_n\|^2} \right] \\ &\quad + \text{higher order terms.} \\ &< 0. \end{aligned} \quad (48)$$

then

$$\begin{aligned} \Delta t \uparrow &\Rightarrow D(r, r, Z_0^{t_n}, \tilde{Z}_0^{t_n}) \uparrow. \\ \Delta t \uparrow &\Rightarrow |D(r, s, Z_0^{t_n}, \tilde{Z}_0^{t_n})| \uparrow. \end{aligned}$$

In other words, the longer the searcher stays at z_n or $\tilde{z}_n$ (i.e., as $\Delta t \uparrow$),

SIMULATION RESULTS

In this section, simulation results for a single target - single searcher Search Estimation Problem are illustrated. Different searcher paths are chosen for the experiments. Conditional densities corresponding to each path are calculated to investigate the influence of each path. Also, Δt and the misdetection constant k are varied to observe their roles in the calculation of $\bar{p}(t_n, Q_r | z_0^{t_n})$. The search domain is a square region as shown in Figure 5 with 9 evaluation points. Assume data are assigned as follows:

$$\begin{aligned} \tilde{z}(t_0) &= [0.5, 1.5], \\ z(t_0) &= [0.5, 1.1], \text{ and} \end{aligned}$$

$$\text{Covariance of } \tilde{z}(t_0) = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1/1.5 \end{bmatrix}$$

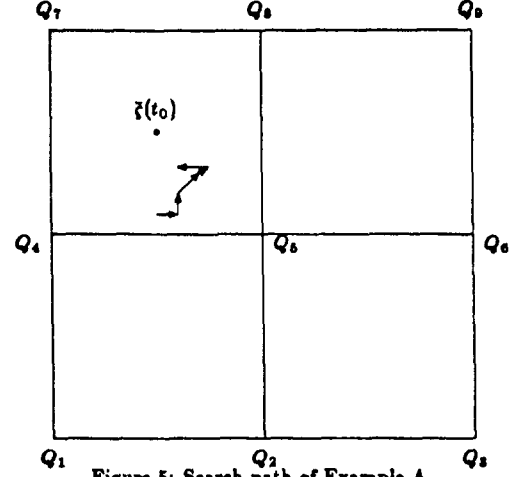


Figure 5: Search path of Example A.

Example A

If misdetection constant k , Δt and search path $Z_0^{t_n}$ are assigned as:

$$\begin{aligned} k &= 1.0 \\ \Delta t &= 0.05 \\ z(t_0) &= [0.5, 1.1] \\ z(t_1) &= [0.6, 1.1] \\ z(t_2) &= [0.6, 1.2] \\ z(t_3) &= [0.7, 1.3] \\ z(t_4) &= [0.75, 1.33] \\ z(t_5) &= [0.8, 1.33] \end{aligned}$$

then the values of unnormalised pdf

$$\begin{aligned} \bar{p}(t_n, Q_7 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_8 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_9 | z_0^{t_n}) \\ \bar{p}(t_n, Q_4 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_5 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_6 | z_0^{t_n}) \\ \bar{p}(t_n, Q_1 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_2 | z_0^{t_n}) & \quad \bar{p}(t_n, Q_3 | z_0^{t_n}) \end{aligned}$$

at each time instant are shown in Figure 7 for the time interval $[t_0, t_5]$. Note that for $0 < n \leq 5$, we have

$$\begin{aligned} \bar{p}(t_n, Q_7 | z_0^{t_n}) &> \bar{p}(t_n, Q_8 | z_0^{t_n}) \\ \bar{p}(t_n, Q_4 | z_0^{t_n}) &> \bar{p}(t_n, Q_5 | z_0^{t_n}) \\ \bar{p}(t_n, Q_1 | z_0^{t_n}) &> \bar{p}(t_n, Q_2 | z_0^{t_n}). \end{aligned}$$

These can be predicted by examining the search path in Figure 5.

Example B

If misdetection constant k , Δt and search path $Z_0^{t_n}$ are assigned as:

$$\begin{aligned} k &= 1.0 \\ \Delta t &= 0.05 \\ z(t_0) &= [0.5, 1.1] \\ z(t_1) &= [0.6, 1.1] \\ z(t_2) &= [0.45, 1.15] \\ z(t_3) &= [0.4, 1.15] \\ z(t_4) &= [0.35, 1.15] \\ z(t_5) &= [0.30, 1.13] \end{aligned}$$

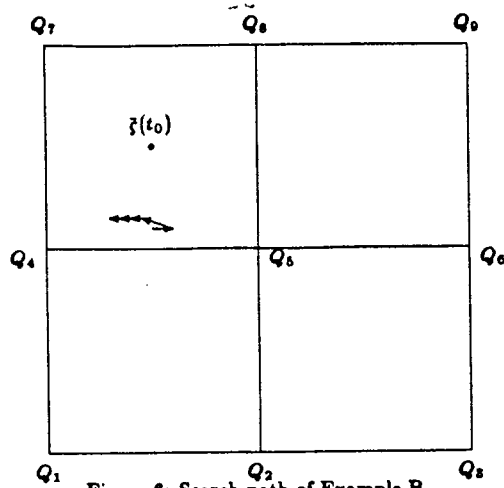


Figure 6: Search path of Example B.

Then the values of unnormalised conditional pdf

$$\begin{array}{ccc} \bar{p}(t_n, Q_7 | z_0^{t_n}) & \bar{p}(t_n, Q_8 | z_0^{t_n}) & \bar{p}(t_n, Q_9 | z_0^{t_n}) \\ \bar{p}(t_n, Q_4 | z_0^{t_n}) & \bar{p}(t_n, Q_5 | z_0^{t_n}) & \bar{p}(t_n, Q_6 | z_0^{t_n}) \\ \bar{p}(t_n, Q_1 | z_0^{t_n}) & \bar{p}(t_n, Q_2 | z_0^{t_n}) & \bar{p}(t_n, Q_3 | z_0^{t_n}) \end{array}$$

at each time instant are illustrated in Figure 8 for the time interval $[t_0, t_5]$. Note that for $0 < n \leq 2$, we have

$$\begin{array}{l} \bar{p}(t_n, Q_7 | z_0^{t_n}) > \bar{p}(t_n, Q_8 | z_0^{t_n}) \\ \bar{p}(t_n, Q_4 | z_0^{t_n}) > \bar{p}(t_n, Q_5 | z_0^{t_n}) \\ \bar{p}(t_n, Q_1 | z_0^{t_n}) > \bar{p}(t_n, Q_2 | z_0^{t_n}) \end{array}$$

However, for $3 \leq n \leq 5$, we have

$$\begin{array}{l} \bar{p}(t_n, Q_7 | z_0^{t_n}) < \bar{p}(t_n, Q_8 | z_0^{t_n}) \\ \bar{p}(t_n, Q_4 | z_0^{t_n}) < \bar{p}(t_n, Q_5 | z_0^{t_n}) \\ \bar{p}(t_n, Q_1 | z_0^{t_n}) < \bar{p}(t_n, Q_2 | z_0^{t_n}) \end{array}$$

These can also be predicted by just looking at the search path in Figure 6.

Remark 8 The plot of

$$\frac{\bar{p}(t_2, Q_4 | z_0^{t_2}) - \bar{p}(t_2, Q_5 | z_0^{t_2})}{\bar{p}(t_2, Q_4 | z_0^{t_2})}$$

versus different values of Δt and k are illustrated in Figures 9 and 10, respectively. The former curve is slightly super-linear, and the latter plot is linear.

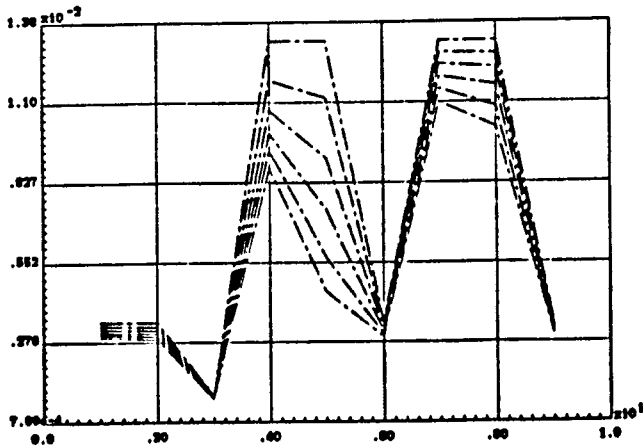


Figure 7: Example A: conditional density ($y = \bar{p}(t_n, Q_r | z_0^{t_n})$) versus evaluation points ($x = [Q_1, \dots, Q_9]$) for the time of $[t_0, t_5]$.

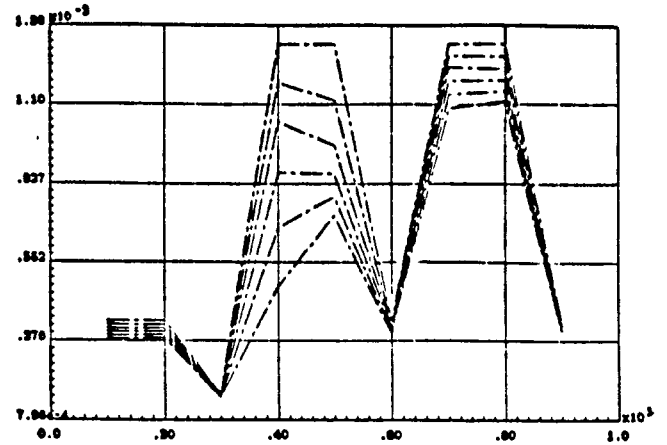


Figure 8: Example B: conditional density ($y = \bar{p}(t_n, Q_r | z_0^{t_n})$) versus evaluation points ($x = [Q_1, \dots, Q_9]$) for the time of $[t_0, t_5]$.

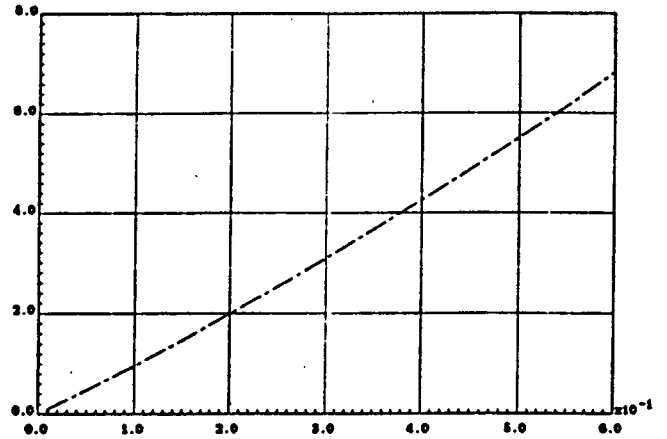


Figure 9: Conditional density difference ($y = \frac{100 |\bar{p}(t_2, Q_4) - \bar{p}(t_2, Q_5)|}{\bar{p}(t_2, Q_4)}$) versus ($x = \Delta t$).

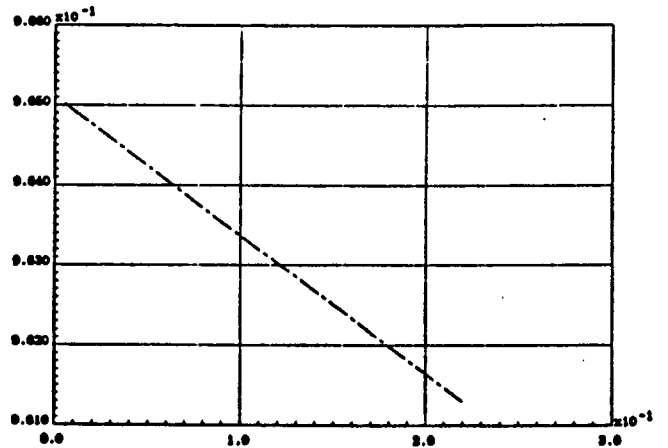


Figure 10: Conditional density difference ($y = \frac{100 |\bar{p}(t_2, Q_4) - \bar{p}(t_2, Q_5)|}{\bar{p}(t_2, Q_4)}$) versus ($x = k$).

References

- [1] W. H. Fleming and S. K. Mitter, "Optimal Control And Nonlinear Filtering For Nondegenerate Diffusion Processes," *Stochastics*, Vol. 8, pp. 63-77, 1982.
- [2] A. Ohsumi, "Optimal Searching For a Markovian Target And Relation To Optimal Stochastic Control," *Proceedings of MTNS, Stockholm*, 1985.
- [3] A. Ohsumi, "Stochastic Control With Searching A Randomly Moving Target," *Proceedings of The American Control Conference*, June, 1984.
- [4] Marc Mangel, "Search Theory," *Lecture Notes*, Springer-Verlag, Berlin, 1984.
- [5] Marc Mangel, "Search For A Randomly Moving Object," *SIAM, J. Appl. Math.*, Vol 40, No.2, April, 1981.
- [6] L. D. Stone and J. B. Kadane, "Optimal Whereabouts Search For A Moving Target," *Operations Research*, pp.1154-1166, 1981.
- [7] L. D. Stone, "Necessary And Sufficient Conditions For Optimal Search Plans For Moving Targets," *Math. of Oper. Res.*, 4, pp.431-440, 1979.
- [8] A. Washburn, "Search For A Moving Target : The FAB Algorithm," *Operations Research*, 31, pp.739-751, 1983.
- [9] J. L. Hibey, "Control-Theoretic Approach To Optimal Search For A Class Of Markovian Target," *Proc. 1982, American Control Conference*, Arlington, Virginia, pp. 705-709, June, 1982.
- [10] Ing Yan, "The Detection Search Problem For Systems of Randomly Moving Targets," *Ph.D. Dissertation*, University of Maryland, College Park, 1987.

SEARCH FOR RANDOMLY MOVING TARGETS II: SEARCH PATH PLANNING¹

I. Yan² and G.L. Blankenship
Systems Research Center and
Electrical Engineering Department
University of Maryland, College Park, MD 20742,

ABSTRACT

The *detection search problem* is the identification of search paths for a specified time interval $[0, T]$, so that the expected number of surviving targets at time T is minimised. The problem can be solved in real time only when the two major procedures: (1) estimation of target posterior distribution; and (2) evaluation of optimal controls (search path planning) based on this posterior target distribution can be done on line. The path planning problem is difficult since the state space is infinite dimensional. We introduce a discrete space-time model to which the Ordered Search Algorithm in AI Graph Search theory (with proper modifications) can be applied. The algorithm not only stops at the (an) optimal path but also expands far fewer nodes than an exhaustive search. The target estimation problem is treated in Part I.³

INTRODUCTION

Detection search involves determining searcher routes to maximise the probability of detecting the remaining (surviving) targets. The major tasks of a detection search are:

1. compute a prior distribution of target location;
2. obtain a good estimate of sensor capabilities;
3. determine a detection (mis-detection) function;
4. develop a search plan and estimate its success probability;
5. update the posterior target distribution from search feedback; and
6. evaluate search effectiveness.

Each is nontrivial and can only be solved by tools which compromise between theoretical optimality and operational feasibility. For example, there is no exact way of computing the prior target distribution, especially when targets are hostile. The prior target distribution could be an approximation reflecting a mixture of objective measurements (or calculations) and subjective information, based on experience or heuristics. Subjective information may be used to simplify the problem so as to generate a good search plan in a feasible amount of time.

In [1,2,3,4] detection search problems are treated as "resource allocation problems" in which the search effort is assumed to be infinitely divisible. However, for the "path planning problems" we assume that the searchers are discrete units constrained to follow continuous paths. This work is devoted to the study of procedures 4 and 5 which are called the "search estimation problem" and the "search control problem", treated, respectively, in Part I and Part II of this paper.

1. TARGET AND SEARCHER DYNAMICS

Consider a detection search problem with target and searcher dynamics

$$d\zeta(t) = f(t, \zeta(t)) dt + G(t, \zeta(t)) dW(t) \quad (1)$$

$$dx(t) = u(t) dt. \quad (2)$$

where $\zeta(t) \in R^{2N}$, $x(t) \in R^{2M}$ are the target and searcher paths, respectively. N is the number of targets and M is the number of searchers. In equation (1), $f(t, \zeta) \in R^{2N}$ and $G(t, \zeta) \in R^{2N \times L}$ are smooth functions of their arguments. The initial value $\zeta(0)$ has a pdf $p_0(\zeta)$ which we assume is known. Searcher dynamics (2) are assumed to be deterministic, and the

control vector $u(t)$ is the searcher velocity.

Note that we have no observation of the target processes. The only information available is the initial distribution and the results of searching up to time t . The following definitions are made for the single searcher - single target case which we shall treat here:

1. **detection function**, $\psi(t, \zeta; x)$
the probability density function of detection

$$\psi(t, \zeta; x) dt = P\{\text{detecting target during } (t, t+dt) | \zeta(t) = \zeta, x(t) = x\} + O(dt) \quad (3)$$

2. **mis-detection function**, $\varphi(t, \zeta; x)$
the probability density function of mis-detection

$$1 - \varphi(t, \zeta; x) dt = \psi(t, \zeta; x) dt + O(dt) \quad (4)$$

3. **unsuccessful event**, x_0^t
the event of no detection when searching along track $x_0^t = \{x(r), 0 \leq r \leq t\}$.

4. **conditional pdf**, $p(t, \zeta | x_0^t)$
the conditional pdf of $\zeta(t) = \zeta$, conditioned on event x_0^t .

$$p(t, \zeta | x_0^t) d\zeta = P\{\zeta(t) \in d\zeta | \zeta(t) = \zeta, x_0^t\} + O(d\zeta)$$

As shown in [5], $p(t, \zeta | x_0^t)$ will be the solution of a stochastic nonlinear integro-partial differential equation:

$$\frac{\partial p(t, \zeta | x_0^t)}{\partial t} = L_t^* p(t, \zeta | x_0^t) - p(t, \zeta | x_0^t) \{ \psi(t, \zeta; x) - \hat{\psi}(t, \zeta; x_0^t) \} \quad (5)$$

where:

1. The adjoint differential generator L_t^* is defined as :

$$L_t^*(\cdot) = - \sum_{i=1}^{2N} \frac{\partial (f_i(t, \zeta) \cdot)}{\partial \zeta_i} + \frac{1}{2} \sum_{i,j}^{2N, 2M} \frac{\partial^2 (G_i G_j |_{i,j} \cdot)}{\partial \zeta_i \partial \zeta_j}. \quad (6)$$

2. The conditional forward detection function $\hat{\psi}(t; x_0^t)$ is defined as:

$$\hat{\psi}(t; x_0^t) = \int_{R^2} \psi(t, \zeta; x) p(t, \zeta | x_0^t) d\zeta. \quad (7)$$

We define

$$\bar{p}(t, \zeta | x_0^t) \equiv \exp\left[-\int_0^t \hat{\psi}(r; x_0^r) dr\right] p(t, \zeta | x_0^t), \quad (8)$$

$$p(t, \zeta | x_0^t) \equiv \frac{\bar{p}(t, \zeta | x_0^t)}{\int \bar{p}(t, \eta | x_0^t) d\eta}. \quad (9)$$

That is, $p(t, \zeta | x_0^t)$ and $\bar{p}(t, \zeta | x_0^t)$ are the normalised and unnormalised conditional pdf, respectively. After some manipulation, a simpler equation for $\bar{p}(t, \zeta | x_0^t)$ can be derived: (see [6,7] for details)

$$\frac{\partial \bar{p}(t, \zeta | x_0^t)}{\partial t} = L_t^* \bar{p}(t, \zeta | x_0^t) - \bar{p}(t, \zeta | x_0^t) \psi(t, \zeta; x). \quad (10)$$

Equation (10) is an ordinary linear parabolic partial differential equation which generally must be solved numerically.

The components of the forward search equations, (5) and (10), involve a covariance matrix representing target diffusion, a drift vector for target motion, and a detection function characterising searcher detection capability.

In Part I of this work we reported on an approximate procedure for the

¹Supported in part by NSF grant CDR-85-00108 to the Systems Research Center.

²Current address: SEI, 7833 Walker Drive, Suite 308, Greenbelt, MD 20770.

³Part I will be published elsewhere.

9. COMPARISON WITH DYNAMIC PROGRAMMING

Dynamic Programming solves problems backward. In the context of Graph Search Theory, it traces the graph bottom-up. An optimal policy is accumulated from the bottom of a graph step by step. In other words, at each intermediate time, choose the best path initiated at all intermediate initial states, by using the principle of optimality, and repeat this process with the initial time backed up one step until the whole specified time horizon is considered. Each link cost is considered exactly once.

The link cost of the target search problem is the incremental return function $l_{ij}(\cdot)$. It is a function of the posterior target distributions which can only be calculated when all the nodes on the path connecting the starting node and the current node are expanded. However, Dynamic Programming needs to use every link cost exactly once. This implies we have to expand all the nodes above bottom level (on all the feasible paths) to make the tree ready for applying Dynamic Programming. Therefore, tracing the search tree backward by Dynamic Programming (bottom-up) is no better than an Exhaustive Search. That is why we chose a top-down algorithm, the Ordered Search Algorithm, to find an optimal path smartly, with the help of a qualified heuristic function to avoid expanding all the nodes.

10. SUMMARY

This paper presents efficient algorithms for solving the "Search Control Problem." The Search Control Problem is formulated into a simplified discrete space-time problem. Some simplifications, such as using Bayes Rule to update "cell" posterior distribution and using the Random Search Formula to estimate the detection probability in one cell, are introduced to simplify the computations. The Ordered Search Algorithm from AI graph search theory is used to generate the (an) optimal path without expanding all the feasible paths. An optimal heuristic function is constructed proves to be very efficient in finding the (an) optimal path. When this optimal heuristic function is applied in the Ordered Search Algorithm, we only expand about one ninth of the nodes expanded by an Exhaustive Search to generate the (an) optimal path.

By embedding the Dual Estimation Problem into the Optimal Ordered Search Algorithm to generate "accurate" posterior target distribution, this approach describes the real model more precisely. The whole algorithm is called Optimal Detection Search Algorithm. It not only updates target distribution at the beginning of each time interval to ensure the accuracy of the Search Estimation Problem, but also finds the (an) optimal path of the Search Control Problem based on this distribution and a simplified discrete space-time search model. The total procedure, we believe, can be done on line after speed improvement.

Acknowledgments: We would like to thank D.S. Nau for his comments on this work.

APPENDIX

ORDERED SEARCH ALGORITHM AND DEFINITIONS

The definitions and algorithms in this Appendix are based on [20] and [21]. They are standard in graph search theory.

$L(k) = (1 - w) G(k) + w H(k)$ with $w \in [0, 1]$.
 $\Gamma(k)$ $\equiv$ set of nodes generated by applying operation Γ to node k .
 S $\equiv$ set of nodes already expanded (CLOSED).
 $\tilde{S}$ $\equiv$ set of nodes yet to be expanded (OPEN).

ORDERED SEARCH ALGORITHM (BEST-FIRST SEARCH)

begin

1. $S = \emptyset$, $\tilde{S} = \{\text{start node}\}$.

2. while $\tilde{S} \neq \emptyset$ do

{ choose $n \in \tilde{S}$ $\ni$ n has the most merit,

if n is the goal, then exit with success,

else $\tilde{S} = \tilde{S} - \{n\}$

$S = S \cup \{n\}$

for each $m \in \Gamma(n)$ do

(a) if $m \notin S$ and $m \notin \tilde{S}$,

then $\tilde{S} = \tilde{S} \cup \{m\}$ in merit order, and point m to n .

(b) if $m \in \tilde{S}$ and $\text{merit}(m)$ is better than the old merit, then point m to n and use new merit to order elements in $\tilde{S}$.

(c) if $m \in S$ and merit is better than old merit, then
 $\tilde{S} = \tilde{S} \cup \{m\}$ with new merit, $S = S - \{m\}$, and point m to n .
 }
 3. exit with failure.
 end;

Definition 1 An order search strategy is complete if whenever there exists a solution to the problem, the strategy will find one.

Definition 2 An order search strategy is admissible if whenever it terminates, the strategy yields a minimum cost (maximum return) solution if one exists.

Definition 3 Let H_1 and H_2 be two heuristic functions, and that $H_2(k) < H_1(k) \leq H_p(k)$ for any node k . An admissible search strategy is said to be optimal if searching with H_2 expands all the nodes that searching with H_1 expands.

Definition 4 A heuristic function $H(k)$ is said to be monotonic or consistent if given any two nodes k and k' such that k' is a successor of k , we have $H(k) - H(k') \leq l(k, k')$, where $l(k, k')$ is the cost of going from k to k' .

Theorem 3 If $w \in [0, 0.5]$ and $H(k) \leq H_p(k)$, then the order search strategy is admissible for all δ -graphs containing a minimum solution.

Theorem 4 If $w \in [0, 0.5]$, $H_2(k) \leq H_1(k) \leq H_p(k)$ for any node k , and $H_1(k)$ and $H_2(k)$ are consistent, then the order search strategy is optimal for all δ -graph containing a minimum solution.

References

- [1] L. D. Stone and J. B. Kadane, "Optimal Whereabouts Search For A Moving Target," Operations Research, pp. 1154-1166, 1981.
- [2] L. D. Stone, "Necessary And Sufficient Conditions For Optimal Search Plans For Moving Targets," Math. of Oper. Res., 4, pp. 431-440, 1979.
- [3] A. Washburn, "Search For A Moving Target : The FAB Algorithm," Operations Research, 31, pp. 739-751, 1983.
- [4] J. L. Hibey, "Control-Theoretic Approach To Optimal Search For A Class Of Markovian Target," Proc. 1982, American Control Conference, Arlington, Virginia, pp. 705-709, June, 1982.
- [5] A. Ohsumi, "Optimal Searching For a Markovian Target And Relation To Optimal Stochastic Control," Proceedings of MTNS, Stockholm, 1985.
- [6] A. Ohsumi, "Stochastic Control With Searching A Randomly Moving Target," Proceedings of The American Control Conference, June, 1984.
- [7] Marc Mangel, "Search For A Randomly Moving Object," SIAM, J. Appl. Math., Vol 40, No.2, April, 1981.
- [8] Marc Mangel, "Search Theory," Lecture Notes, Springer-Verlag, Berlin, 1984.
- [9] B. O. Koopman, "Search and Screening," N.Y. : Pergamon Press, 1980.
- [10] L. D. Stone, "The Process Of Search Planning : Current Approaches And Continuing Problems," Operations Research, 31, pp. 207-233, 1983.
- [11] Ing Yan, "Estimation and Path Planning in Detection Search," Ph.D. Dissertation, University of Maryland, College Park, 1987.
- [12] N. J. Nilsson, "Principles Of Artificial Intelligence," Tioga Publishing Co., 1980.
- [13] A. Barr and E. A. Feigenbaum (editors), "The Handbook Of Artificial Intelligence," William Kaufmann, Inc., 1981.
- [14] I. Yan and G. L. Blankenship, "Search For Randomly Moving Targets I : Estimation," presented in 1987 Command and Control Research Symposium and submitted for Proceedings, Washington, DC, June, 1987.

(C3) For any two consecutive visiting cells, k_i and k_{i+1} , we have either $k_i = k_{i+1}$ or k_i and k_{i+1} are neighboring cells. This makes the cell visiting sequence feasible, since the searcher can not jump from one cell to a non-adjacent cell in zero time.

Let $F_{n+1}^s(k_n)$ denote the set of all feasible cell-visiting sequences starting from cell k_n at time t_n ,

$$F_{n+1}^s(k_n) \equiv \{ [k_i]_{i=n+1}^s | t(i, k_i) = \Delta t, \text{ and (C1), (C2) and (C3) are satisfied} \}.$$

Therefore, D_{i+1}^{t+1} is simplified to

$$D_{i+1}^{t+1} = P(t_i, k_i) d(\Delta t).$$

At time $t_n \in [0, T]$, the prior target distribution is $P(t_n, k_n)$ which is computed from the dual estimation problem (treated in Part I). However, to predict the $D_{n+1}^{t+1}, D_{n+2}^{t+1}, \dots, D_{n+s}^{t+1}$ functions, the posterior target distributions $P(t_i, k_i)$ (for $i > n$) must be evaluated from the given prior target distribution and trial control very quickly. As suggested in [10], Bayes' Rule is applied to approximate the posterior distribution

$$P(t_{i+1}, k) = \frac{P(t_i, k) [1 - d(t(i, k))]}{\sum_{j=1}^K P(t_i, j) [1 - d(t(i, j))]} \quad (13)$$

Although only the visiting cell posterior distributions $\{P(t_i, k_i)\}_{i=n}^s$ appear in D_{i+1}^{t+1} , all the $P(t_i, k)$, for $1 \leq k \leq K$, are used in (13) to update the next visiting posterior distribution $P(t_{i+1}, k_{i+1})$. Hence, one evaluation of the cost function requires the calculation of all the target distributions $\{P(t_i, k)\}_{k=1}^K$ for each given trial control sequence. This is expensive, but the simplifications introduced here reduce the complexity of the search operations model.

4. EFFORT ALLOCATION VERSUS PATH PLANNING

Washburn [3] proposed an FAB algorithm, the Forward And the Backward algorithm, for search operation. However, the FAB algorithm is designed for continuous search "effort allocation" problems (for which it can generate an optimal solution when feasible effort allocation has certain properties), but not for search "path planning" problems (for which it can only generate a critical solution and a lower bound of the optimal solution).

Path planning is a very constrained special case of effort allocation. An individual searcher cannot be "partitioned" to visit different cells at the same time - searchers are discrete units which may not be subdivided. Furthermore, they must follow continuous search tracks and cannot jump from one place to another in zero time. Therefore, most of the nice properties associated with continuous search effort formulations has, such as the convexity of feasible effort distribution set, do not hold in search path planning for discrete searchers (see [3] for details).

If the FAB algorithm is applied for path planning in the case considered here, we can only obtain a critical solution. The corresponding estimate of the lower bound of the optimal cost is too loose to be useful. Examples are given in [3], p. 750.

5. CONDITIONAL RETURN FUNCTION

We define a return function $R(P(t_n, \cdot), k_n, k_{n+1}^s)$ for the time interval $[t_n, t_s]$ by

$$R(P(t_n, \cdot), k_n, k_{n+1}^s) = \text{probability of detection during } [t_n, t_s].$$

Also, let

$$\begin{aligned} \bar{D}_{i+1}^{t+1} &\equiv 1 - D_{i+1}^{t+1}, \\ \bar{D}_{i+1}^{t+1} &\equiv \prod_{l=0}^{j-1} \bar{D}_{i+l}^{t+1}, \end{aligned}$$

then

$$\begin{aligned} R(P(t_n, \cdot), k_n, k_{n+1}^s) &= D_{n+1}^{t+1} + \bar{D}_{n+1}^{t+1} D_{n+2}^{t+1} + \bar{D}_{n+1}^{t+1} \bar{D}_{n+2}^{t+1} D_{n+3}^{t+1} + \dots + \bar{D}_{n+s-1}^{t+1} D_{n+s}^{t+1} \\ &= \sum_{i=n}^{s-1} [D_{i+1}^{t+1} | \text{conditioned on } \bar{D}_{i+1}^{t+1}] \\ &= P(t_n, k_n) d(\Delta t) + [1 - P(t_n, k_n) d(\Delta t)] P(t_{n+1}, k_{n+1}) d(\Delta t) \\ &\quad + \dots + \left\{ \prod_{i=n}^{s-1} [1 - P(t_i, k_i) d(\Delta t)] \right\} \cdot P(t_s, k_s) d(\Delta t). \end{aligned}$$

The optimal detection search problem is to maximise the return function $R(P(t_n, \cdot), k_n, k_{n+1}^s)$ over all feasible paths. The search operation model can be expressed as:

Objective :

$$\begin{aligned} \max_{k_{n+1}^s \in F_{n+1}^s(k_n)} R(P(t_n, \cdot), k_n, k_{n+1}^s) \\ = \max_{k_{n+1}^s \in F_{n+1}^s(k_n)} \{ P(t_n, k_n) d(\Delta t) \\ + \sum_{i=n+1}^{s-1} \prod_{j=n}^{i-1} [1 - P(t_j, k_j) d(\Delta t)] P(t_i, k_i) d(\Delta t) \} \end{aligned} \quad (14)$$

where:

- (1) $k_{n+1}^s = \{k_{n+1}, \dots, k_s\} \in F_{n+1}^s(k_n)$, with $F_{n+1}^s(k_n) \equiv \{[k_i]_{i=n+1}^s | [k_i]_{i=n+1}^s \text{ a feasible cell visiting sequence for time interval } [t_{n+1}, t_s], \text{ starting from cell } k_n \text{ at time } t_n\}$.
- (2) $t(i, k) = \begin{cases} \Delta t & k = k_i, \\ 0 & \text{otherwise.} \end{cases}$
- (3) $P(t_{i+1}, k) = \frac{P(t_i, k) [1 - d(t(i, k))]}{\sum_{j=1}^K P(t_i, j) [1 - d(t(i, j))]}$ for $i = n, n+1, \dots, s, j = 1, 2, \dots, K$, with $\{P(t_n, k)\}_{k=1}^K$ given as prior distributions.
- (4) $d(t(i, k)) = 1 - \exp(-\frac{WV}{A} t(i, k))$, with W, V and A as given constants.

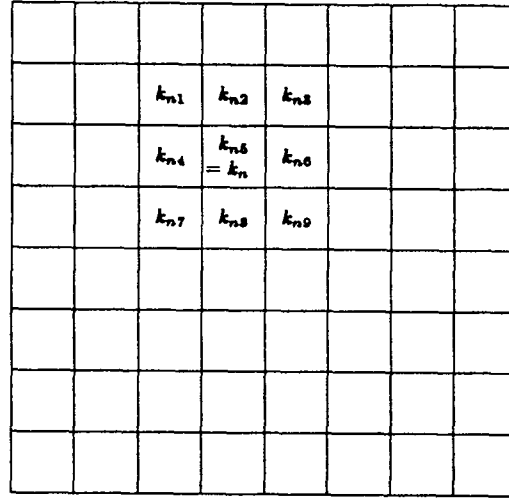


Figure 3: Neighbor cells of cell k_n .

We define:

$$\begin{aligned} N(k_n) &\equiv \text{set of neighbor cells of cell } k_n \\ &= \{k_{nj}, j = 1, 2, \dots, 9, \text{ with } k_{n5} = k_n\}. \end{aligned}$$

Since there are 9 elements in set $N(k_n)$, there should be, at most, 9^{s-n} feasible cell-visiting sequences in set $F_{n+1}^s(k_n)$ given k_n as a starting cell at time t_n (cells at boundary have fewer neighbors). Therefore, the search space $F_{n+1}^s(k_n)$ is a finite set. Let $k_{ni}, k_{nij}, \dots$ denote the nodes reached from node k_n through path $\{k_n, k_{ni}, k_{nij}, k_{nijl}, \dots\}$ at time instants $\{t_n, t_{n+1}, t_{n+2}, t_{n+s}, \dots\}$ respectively. Note that the return function (probability of detection) for time interval $[t_n, t_s]$ has a summation form:

$$R(P(t_n, \cdot), k_n, k_{n+1}^s) = \sum_{i=n}^{s-1} [D_{i+1}^{t+1} | \text{conditioned on } \bar{D}_{i+1}^{t+1}].$$

Then, at time t_n , for $\forall n \leq m \leq s$, the estimated return function for a path starting from node k_n traversing a partial cell-visiting sequence $\{k_n, k_{n+1}, \dots, k_m\}$ is defined by $L(k_m)$, called the conditional estimate of the real return function for time interval $[t_n, t_s]$, constrained to going through partial path $\{k_n, k_{n+1}, \dots, k_m\}$.

$$\begin{aligned} L(k_m) &\equiv \text{the estimate of } R(P(t_n, \cdot), k_n, k_{n+1}^s) \text{ when search begins in cell } k_n \text{ and goes through } \{k_n, \dots, k_m\}. \\ &= [\text{the known return by searching along } \{k_n, \dots, k_m\}] \\ &\quad + [\text{the estimated return for time interval } [t_{m+1}, t_s]] \\ &\equiv G(k_m) + H(k_m). \end{aligned}$$

Assume $\{k_n, k_{ni}, k_{nij}, \dots\}$ is a feasible cell visiting sequence, then the incremental return functions of visiting cell $k_n, k_{ni}, k_{nij}, \dots$ are denoted by $l(k_n), l(k_{ni}), l(k_{nij}), \dots$ respectively.

$$\begin{aligned} l(k_n) &\equiv D_{n+1}^{t+1} = P(t_n, k_n) d(\Delta t) \\ l(k_{ni})(k_{ni}) &\equiv \bar{D}_{n+1}^{t+1} D_{n+2}^{t+1} \\ &= [1 - P(t_n, k_n) d(\Delta t)] P(t_{n+1}, k_{ni}) d(\Delta t) \end{aligned}$$

$$l_{(k_n, k_{n+1})}(k_{n+2}) \equiv D_{t_n}^{t_{n+2}} D_{t_{n+1}}^{t_{n+2}} \\ = [1 - P(t_n, k_n)d(\Delta t)] [1 - P(t_{n+1}, k_{n+1})d(\Delta t)] \\ P(t_{n+2}, k_{n+2})d(\Delta t)$$

Then the heuristic function, $H(k_n)$, of node k_n can be expressed in terms of the subsequent incremental return functions and the heuristic functions of k_n 's successors.

$$\begin{aligned} H(k_n) &= \max_{k_{n+1} \in N(k_n)} \{ l_{(k_n, k_{n+1})}(k_{n+1}) + H(k_{n+1}) \} \\ &= \max_{k_{n+1} \in N(k_n)} \{ l_{(k_n, k_{n+1})}(k_{n+1}) + H(k_{n+1}) \} \\ &= \max_{k_{n+1} \in N(k_n)} \{ l_{(k_n, k_{n+1})}(k_{n+1}) \\ &\quad + \max_{k_{n+2} \in N(k_{n+1})} [l_{(k_{n+1}, k_{n+2})}(k_{n+2}) + H(k_{n+2})] \} \\ &= \max_{k_{n+1} \in N(k_n)} \max_{k_{n+2} \in N(k_{n+1})} \{ l_{(k_n, k_{n+1})}(k_{n+1}) + l_{(k_{n+1}, k_{n+2})}(k_{n+2}) + H(k_{n+2}) \} \end{aligned} \quad (15)$$

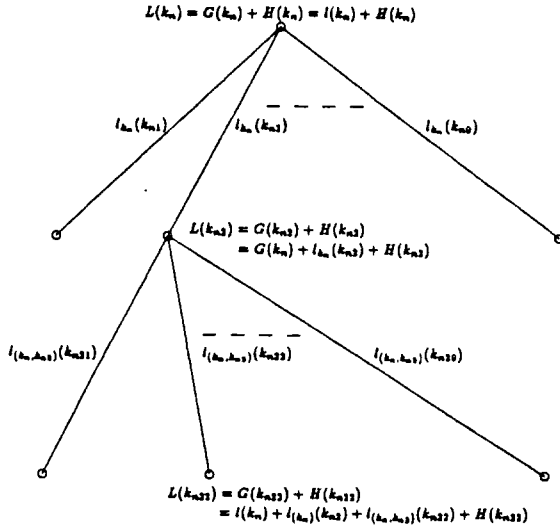


Figure 4: $H(k_n)$ is a function of the subsequent incremental return functions and the heuristic functions of k_n 's successors.

6. OPTIMAL DETECTION SEARCH

In this section, an Optimal Ordered Search (Best-First Search) strategy is introduced which can find the optimal cell-visiting sequence for problem described by model (14) whenever the algorithm terminates.

Define the perfect return function, $L_p(k_n)$, starting from cell k_n being the maximum achievable detection probability among all feasible searching sequences $\{k_{n+1}^* | k_{n+1}^* \in F_{n+1}^*(k_n)\}$:

$$\begin{aligned} L_p(k_n) &\equiv \max_{k_{n+1}^* \in F_{n+1}^*(k_n)} R(P(t_n, \cdot), k_n, k_{n+1}^*) \\ &\equiv G(k_n) + H_p(k_n) \\ &= l(k_n) + \max_{k_{n+1} \in N(k_n)} \{ l_{(k_n, k_{n+1})}(k_{n+1}) + H_p(k_{n+1}) \} \\ &= l(k_n) + \max_{k_{n+1}^* \in F_{n+1}^*(k_n)} \left\{ \sum_{i=0}^{s-n-1} l_{(k_n, k_{n+1}, \dots, k_{n+i})}(k_{n+i+1}) \right\} \end{aligned}$$

where: $H_p(k_{n+i})$ is the perfect heuristic function which gives the exact maximum achievable return for time interval $[t_{n+i+1}, t_s]$ when cell k_{n+i} is visited at time t_{n+i} . Its ground values will be $H_p(k_s) = 0$ for any cell k_s visited at time t_s . Node k_n is said to be expanded, if the associated posterior distributions, $P(t_{n+1}, k)$, for $\forall k = 1, 2, 3, \dots, K$, are computed by Bayes' Rule. To compute the perfect return function, $L_p(k_n)$, by an Exhaustive Search, we need to expand a large number of nodes; in fact, all the nodes from level n to level $s-1$,

$$\sum_{i=n}^{s-1} |\text{number of nodes at level } i| = \sum_{i=n}^{s-1} 9^{s-i},$$

in order to tell which path will yield the maximum return. Fortunately, using graph techniques developed in Artificial Intelligence research, this is not the only way to find the optimal solution. Instead of expanding all the nodes, we can expand nodes according to a "merit order" which is called the Ordered Search Algorithm (see the Appendix for details). If the heuristic function $H(\cdot)$ chosen has certain properties and the Ordered Search Algo-

rithm is applied, we do not have to expand all the nodes to obtain the optimal cell-visiting sequence.

Theorem 1 An Ordered Search Algorithm is optimal (see Appendix for definition) for any δ -graph, if

1. the heuristic function satisfies upper bound condition, i.e., at any node k we have

$$H(k) \geq H_p(k),$$

2. $L(k) \equiv (1 - \omega)G(k) + \omega H(k)$ with $\omega \in [0.5, 1)$, and

3. the heuristic function is consistent (monotonic)

$$H(k_m) - H(k_{m+1}) \geq l_{(k_m, k_{m+1})}(k_{m+1}).$$

Proof: See [11].

The Ordered Search Algorithm terminates whenever it detects the (a) goal node if exists (uniquely). The (a) goal node of the target search problem is a node at level s (a node to visit at time t_s), with its backward pointers up to the starting node k_n giving the (a) maximum return.

Theorem 2 The goal node of the target search problem when applying the Ordered Search Strategy is the first s -level node picked from set $\tilde{S}$ (OPEN, defined in Appendix) for expansion. In another words, the backward pointers, associated with the first s -level node picked from set $\tilde{S}$ for expansion, back all its way to the starting node k_n gives the (an) optimal path. The path gives the highest return in maximizing the probability of detection.

Proof: See [11].

Although the optimal ordered search algorithm can generate the optimal cell-visiting sequence for problems modeled by (14), this model does not describe the real problem completely. It uses: (1) the random search formula to estimate the detection probability in one cell, and (2) Bayes' Rule to update posterior distribution fast. The model (14) is only an approximation for the real problem. We must update this approximate model from time to time by solving the dual estimation problem (treated in Part I) to provide the posterior distribution (conditional density) more accurately. Therefore, a combination of the optimal ordered search and solution of the dual estimation problem is proposed for improved performance.

Optimal Detection Search Algorithm :

1. $n = 0$, set k_0 = the cell occupied by the searcher is in at time t_0 ,
2. Compute $P(t_0, k)$ for $k = 1, 2, \dots, K$, from the initial target distribution,
3. Choose a heuristic function $H(\cdot)$ which satisfies all optimality conditions,
4. Use the ordered search strategy (detailed in the Appendix) to generate the optimal cell-visiting sequence $\{k_{n+1}^*, k_{n+2}^*, \dots, k_s^*\}$,
5. Set $k_{n+1} = k_{n+1}^*$, and perform search in cell k_{n+1} for time interval $[t_{n+1}, t_{n+2}]$,
6. If $n + 1 = s$, then stop.
7. Solve the dual estimation problem for the conditional density function $P(t_{n+1}, k)$, $\forall 1 \leq k \leq K$, to update the prior distribution for the next time interval.
8. Set $n = n + 1$ and go to step 4.

The dual estimation problem is detailed in Part I of [11] and [14].

7. HEURISTIC FUNCTION CANDIDATES

Some heuristic functions are described in this section to show that such functions $H(\cdot)$ exist which permit the evaluation function $L(\cdot)$ to satisfy admissibility and optimality conditions (defined in Appendix). To simplify the notation, we suppress the dependency of the incremental return function $l_{(k_n, k_{n+1}, \dots, k_{m-1})}(k_m)$ on the past cell-visiting sequence $(k_n, k_{n+1}, \dots, k_{m-1})$

$$l(k_m) = l_{(k_n, k_{n+1}, \dots, k_{m-1})}(k_m).$$

(1) Admissibility

For all $m, n \leq m \leq s$, let

$$\begin{aligned} L(k_m) &= G(k_m) + H(k_m) \\ G(k_m) &= \sum_{i=1}^m l(k_i) \\ H(k_m) &= \max_{(k_{m+1}, k_{m+2}, k_{m+3})} [l(k_{m+1}) + l(k_{m+2}) + H_g(k_{m+3})] \end{aligned}$$

and

$$H_g(k_{m+3}) \equiv l(k_{m+3}) + 1.$$

Then

$$\begin{cases} H(k_m) > 1 > H_p(k_m) & \text{(upper bound condition)} \\ w = 0.5 \in [0.5, 1) \end{cases}$$

which implies that the admissibility conditions hold.

(2) Optimality

By definition, the perfect heuristic is:

$$H_p(k_m) \equiv \max_{(k_{m+1})} \sum_{i=m+1}^{\infty} l(k_i) \equiv \sum_{i=m+1}^{\infty} l(k_i^*).$$

In order to define an $H(k_m)$ which satisfies the upper bound condition, we compute an upper bound for all k_{m+i} , $\forall s-m \geq i \geq 1$. From the equality

$$l_{m+i+1} = p(t_{m+i+1}, k_{m+i+1}) \left[\frac{1}{p(t_{m+i+1}, k_{m+i+1})} - d(\Delta t) \right] * l_{m+i}, \quad (16)$$

it suffices to find an upper bound for $p(t_{m+i+1}, k_{m+i+1})$ and a lower bound of $p(t_{m+i}, k_{m+i})$ to give an upper bound for l_{m+i+1} in terms of the upper bound of l_{m+i} . The upper and lower bounds of $p(t_{m+i}, k_{m+i})$ can be calculated by using Bayes' Rule. After some manipulation, we obtain

1. for $k \neq k_{m+i}$,

$$\begin{aligned} p(t_{m+i+1}, k) - p(t_{m+i}, k) &= \frac{p(t_{m+i}, k_{m+i}) d(\Delta t)}{[1 - p(t_{m+i}, k_{m+i}) d(\Delta t)]} p(t_{m+i}, k) \\ &\equiv \Delta \bar{p}_{m+i+1} > 0 \end{aligned}$$

2. for $k = k_{m+i}$,

$$\begin{aligned} p(t_{m+i+1}, k_{m+i}) - p(t_{m+i}, k_{m+i}) \\ = - \frac{p(t_{m+i}, k_{m+i}) * d(\Delta t)}{[1 - p(t_{m+i}, k_{m+i}) * d(\Delta t)]} * [1 - p(t_{m+i}, k_{m+i})] \\ < 0 \end{aligned}$$

Thus, for $k \neq k_{m+i}$ and $i \geq 1$

$$\begin{aligned} p(t_{m+i+1}, k) - p(t_{m+i}, k) &\leq \frac{\bar{p}_{m+i} * d(\Delta t)}{1 - \bar{p}_{m+i} d(\Delta t)} \\ &\equiv \Delta \bar{p}_{m+i+1} \end{aligned}$$

where $\bar{p}_{m+i} \equiv \max_{(k=1,2,\dots,K)} p(t_{m+i}, k)$. If we define

$$\bar{p}_m \equiv \bar{p}_m, \quad (17)$$

$$\Delta \bar{p}_{m+1} \equiv \frac{d(\Delta t) * \bar{p}_m}{[1 - p(t_m, k_m) * d(\Delta t)]} * p(t_m, k_m) \quad (18)$$

then, for $i \geq 1$,

$$\begin{aligned} \bar{p}_{m+i} &\leq \bar{p}_{m+i-1} + \Delta \bar{p}_{m+i} \\ \bar{p}_{m+i} &\equiv \min \{ \bar{p}_{m+i-1} + \Delta \bar{p}_{m+i}, 1 \} \\ \Delta \bar{p}_{m+i+1} &\equiv \frac{\bar{p}_{m+i}^2 * d(\Delta t)}{[1 - \bar{p}_{m+i} * d(\Delta t)]} \\ &< \frac{\bar{p}_{m+i}^2 * d(\Delta t)}{[1 - \bar{p}_{m+i} * d(\Delta t)]} \\ &\equiv \Delta \bar{p}_{m+i+1}. \end{aligned} \quad (19) \quad (20)$$

All $\bar{p}_{m+i}$ and $\Delta \bar{p}_{m+i}$, $s-m \geq i \geq 1$, can be calculated by using definitions (19) and (20) alternatively.

For $k = k_{m+i}$ and $i \geq 1$,

$$p(t_{m+i+1}, k_{m+i}) \geq [1 - \frac{d(\Delta t)}{1 - p(t_{m+i}, k_{m+i}) d(\Delta t)}] p(t_{m+i}, k_{m+i}).$$

If we define

$$L_m \equiv \min_{(k=1,2,\dots,K)} p(t_m, k),$$

$$\Delta \bar{p}_{m+i+1} \equiv \frac{d(\Delta t)}{[1 - \bar{p}_{m+i} d(\Delta t)]}, \quad (21)$$

$$\bar{p}_m = L_m \quad (22)$$

then

$$\begin{aligned} L_{m+i+1} &\equiv [1 - \frac{d(\Delta t)}{1 - \bar{p}_{m+i} d(\Delta t)}] * L_{m+i} \\ &\geq [1 - \frac{d(\Delta t)}{1 - \bar{p}_{m+i} d(\Delta t)}] * \bar{p}_{m+i} \text{ (because } d(\Delta t) < 1) \\ &\equiv [1 - \Delta \bar{p}_{m+i+1}] * \bar{p}_{m+i} \end{aligned}$$

$$\bar{p}_{m+i+1} \equiv \begin{cases} (1 - \Delta \bar{p}_{m+i+1}) \bar{p}_{m+i} & \text{if } 1 - \Delta \bar{p}_{m+i+1} > 0 \\ \bar{p}_{m+i} & \text{otherwise} \end{cases} \quad (23)$$

All $\bar{p}_{m+i}$ and $\Delta \bar{p}_{m+i}$, $s-m \geq i \geq 1$, can be calculated by using (21) and (23) alternatively. Since the upper ($\bar{p}_{m+i}$) and lower ($\bar{p}_{m+i}$) bounds of $p(t_{m+i}, k)$ are obtained from these derivations, let us define the heuristic function:

$$H(k_m) \equiv \sum_{i=m+1}^{\infty} l_i, \quad (24)$$

where

$$l_{m+1} \equiv \bar{p}_{m+1} \left[\frac{1}{p(t_m, k_m)} - d(\Delta t) \right] * l(k_m), \quad (25)$$

for $i \geq 1$

$$l_{m+i+1} \equiv \bar{p}_{m+i+1} \left[\frac{1}{\bar{p}_{m+i}} - d(\Delta t) \right] * l_{m+i}. \quad (26)$$

The heuristic function defined by (24) is an optimal heuristic function (proved in [11]).

8. SIMULATION RESULTS

In this section, simulations are reported which show the performance of the Optimal Detection Search Algorithm with the optimal heuristic function defined by (24). The search domain is partitioned into 100 cells. Assume $\Delta t = 1$ and $d(\Delta t) = 0.2591818$. Initial target distributions are assumed known and as given in Table 1. For instance, the initial probability of target being in cell 38 is located at row 30 and column 8 in the Table.

A PASCAL program was written to implement the Optimal Detection Search Algorithm (O.D.S.A.). If the second node of the merit-ordered "OPEN" list is also a goal node when program terminates, then it is either a second optimal path (if it has less merit than the first goal node), or another optimal path (if it has the same merit as the first goal node). Table 2 lists the (an) optimal path backtracked from the first detected goal node, and the path backtracked from the second node in the "OPEN" list (if it is also a goal node when program terminates). Different time horizons are also considered as shown in Table 2. Probability of detection is the probability of detecting the target when searching along the (an) optimal path generated by the program. Execution time is measured on a Sun 3/160 system for finding the (an) optimal path. The effort spent to find the (an) optimal path is compared with those spent by an Exhaustive Search (E.S.) as shown in Table 3.

Table 1: The initial target distribution.

cell	1	2	3	4	5	6	7	8	9	10
00	0.000	0.01	0.01	0.005	0.005	0.005	0.005	0.01	0.03	0.005
10	0.01	0.01	0.03	0.005	0.01	0.005	0.01	0.01	0.01	0.01
20	0.005	0.01	0.01	0.005	0.005	0.005	0.01	0.03	0.005	0.01
30	0.005	0.005	0.005	0.005	0.03	0.01	0.01	0.01	0.02	0.005
40	0.005	0.005	0.03	0.03	0.01	0.03	0.03	0.03	0.005	0.005
50	0.01	0.01	0.01	0.005	0.03	0.03	0.01	0.005	0.005	0.005
60	0.01	0.01	0.005	0.005	0.01	0.01	0.005	0.01	0.03	0.005
70	0.005	0.01	0.005	0.005	0.03	0.005	0.03	0.03	0.01	0.005
80	0.005	0.005	0.005	0.005	0.03	0.01	0.03	0.005	0.01	0.005
90	0.005	0.005	0.005	0.005	0.005	0.01	0.005	0.005	0.005	0.005

Table 2: The optimal path(s).

starting cell	time horizon	the (an) optimal path	probability of detection	execution time (sec.)
00	3	00, 00, 17.	0.0100000	0.1
00	4	00, 00, 20.	0.0100000	0.1
00	4	00, 47, 00, 00.	0.0333333	1.0
00	4	00, 00, 20, 17.	0.0200000	0.005
00	5	00, 47, 00, 00, 00.	0.0300000	10.0
00	5	00, 47, 00, 00, 00.	0.0300000	0.005
00	5	00, 47, 00, 00, 00, 00.	0.0300000	0.005
00	5	00, 47, 00, 00, 00, 00.	0.0300000	0.005

Table 3: Comparison of Effort Spent.

s	path found	number of nodes expended by O.D.S.A.	number of nodes generated by O.D.S.A.	number of nodes expended by E.S.	number of nodes generated by E.S.
1	optimal	8	10	10	10
4	optimal	103	104	91	91
5	optimal	103	103	710	710
5	optimal	221	213	710	695

Remark 1 As we can observe from Table 2, the optimal path of a longer time horizon is not an extension of that of a shorter time horizon.

optimal path of $s = 4$

$= 00, 47, 00, 00.$

$\neq 00, 00, 17, \dots$

$= \{ \text{optimal path of } s = 3 \}, \dots$

9. COMPARISON WITH DYNAMIC PROGRAMMING

Dynamic Programming solves problems backward. In the context of Graph Search Theory, it traces the graph bottom-up. An optimal policy is accumulated from the bottom of a graph step by step. In other words, at each intermediate time, choose the best path initiated at all intermediate initial states, by using the principle of optimality, and repeat this process with the initial time backed up one step until the whole specified time horizon is considered. Each link cost is considered exactly once.

The link cost of the target search problem is the incremental return function $l_{ij}(\cdot)$. It is a function of the posterior target distributions which can only be calculated when all the nodes on the path connecting the starting node and the current node are expanded. However, Dynamic Programming needs to use every link cost exactly once. This implies we have to expand all the nodes above bottom level (on all the feasible paths) to make the tree ready for applying Dynamic Programming. Therefore, tracing the search tree backward by Dynamic Programming (bottom-up) is no better than an Exhaustive Search. That is why we chose a top-down algorithm, the Ordered Search Algorithm, to find an optimal path smartly, with the help of a qualified heuristic function to avoid expanding all the nodes.

10. SUMMARY

This paper presents efficient algorithms for solving the "Search Control Problem." The Search Control Problem is formulated into a simplified discrete space-time problem. Some simplifications, such as using Bayes Rule to update "cell" posterior distribution and using the Random Search Formula to estimate the detection probability in one cell, are introduced to simplify the computations. The Ordered Search Algorithm from AI graph search theory is used to generate the (an) optimal path without expanding all the feasible paths. An optimal heuristic function is constructed proves to be very efficient in finding the (an) optimal path. When this optimal heuristic function is applied in the Ordered Search Algorithm, we only expand about one ninth of the nodes expanded by an Exhaustive Search to generate the (an) optimal path.

By embedding the Dual Estimation Problem into the Optimal Ordered Search Algorithm to generate "accurate" posterior target distribution, this approach describes the real model more precisely. The whole algorithm is called Optimal Detection Search Algorithm. It not only updates target distribution at the beginning of each time interval to ensure the accuracy of the Search Estimation Problem, but also finds the (an) optimal path of the Search Control Problem based on this distribution and a simplified discrete space-time search model. The total procedure, we believe, can be done on line after speed improvement.

Acknowledgments: We would like to thank D.S. Nau for his comments on this work.

APPENDIX

ORDERED SEARCH ALGORITHM AND DEFINITIONS

The definitions and algorithms in this Appendix are based on [20] and [21]. They are standard in graph search theory.

$L(k) = (1 - w) G(k) + w H(k)$ with $w \in [0, 1]$.
 $\Gamma(k) \equiv$ set of nodes generated by applying operation Γ to node k .
 $S \equiv$ set of nodes already expanded (CLOSED).
 $\tilde{S} \equiv$ set of nodes yet to be expanded (OPEN).

ORDERED SEARCH ALGORITHM (BEST-FIRST SEARCH)

begin

1. $S = \emptyset$, $\tilde{S} = \{ \text{start node} \}$.

2. while $\tilde{S} \neq \emptyset$ do

 { choose $n \in \tilde{S} \ni n$ has the most merit,

 if n is the goal, then exit with success,

 else $\tilde{S} = \tilde{S} - \{n\}$

$S = S \cup \{n\}$

 for each $m \in \Gamma(n)$ do

 (a) if $m \notin S$ and $m \notin \tilde{S}$,

 then $\tilde{S} = \tilde{S} \cup \{m\}$ in merit order, and point m to n .

 (b) if $m \in \tilde{S}$ and $\text{merit}(m)$ is better than the old merit, then point m to n and use new merit to order elements in $\tilde{S}$.

(c) if $m \in S$ and merit is better than old merit, then
 $\tilde{S} = \tilde{S} \cup \{m\}$ with new merit, $S = S - \{m\}$, and point m to n .

3. exit with failure.
end;

Definition 1 An order search strategy is complete if whenever there exists a solution to the problem, the strategy will find one.

Definition 2 An order search strategy is admissible if whenever it terminates, the strategy yields a minimum cost (maximum return) solution if one exists.

Definition 3 Let H_1 and H_2 be two heuristic functions, and that $H_2(k) < H_1(k) \leq H_p(k)$ for any node k . An admissible search strategy is said to be optimal if searching with H_2 expands all the nodes that searching with H_1 expands.

Definition 4 A heuristic function $H(k)$ is said to be monotonic or consistent if given any two nodes k and k' such that k' is a successor of k , we have $H(k) - H(k') \leq l(k, k')$, where $l(k, k')$ is the cost of going from k to k' .

Theorem 3 If $w \in [0, 0.5]$ and $H(k) \leq H_p(k)$, then the order search strategy is admissible for all δ -graphs containing a minimum solution.

Theorem 4 If $w \in [0, 0.5]$, $H_2(k) \leq H_1(k) \leq H_p(k)$ for any node k , and $H_1(k)$ and $H_2(k)$ are consistent, then the order search strategy is optimal for all δ -graph containing a minimum solution.

References

- [1] L. D. Stone and J. B. Kadane, "Optimal Whereabouts Search For A Moving Target," Operations Research, pp. 1154-1166, 1981.
- [2] L. D. Stone, "Necessary And Sufficient Conditions For Optimal Search Plans For Moving Targets," Math. of Oper. Res., 4, pp. 431-440, 1979.
- [3] A. Washburn, "Search For A Moving Target : The FAB Algorithm," Operations Research, 31, pp. 739-751, 1983.
- [4] J. L. Hibey, "Control-Theoretic Approach To Optimal Search For A Class Of Markovian Target," Proc. 1982, American Control Conference, Arlington, Virginia, pp. 705-709, June, 1982.
- [5] A. Ohsumi, "Optimal Searching For a Markovian Target And Relation To Optimal Stochastic Control," Proceedings of MTNS, Stockholm, 1985.
- [6] A. Ohsumi, "Stochastic Control With Searching A Randomly Moving Target," Proceedings of The American Control Conference, June, 1984.
- [7] Marc Mangel, "Search For A Randomly Moving Object," SIAM, J. Appl. Math., Vol 40, No.2, April, 1981.
- [8] Marc Mangel, "Search Theory," Lecture Notes, Spring-Verlag, Berlin, 1984.
- [9] B. O. Koopman, "Search and Screening," N.Y. : Pergamon Press, 1980.
- [10] L. D. Stone, "The Process Of Search Planning : Current Approaches And Continuing Problems," Operations Research, 31, pp. 207-233, 1983.
- [11] Ing Yan, "Estimation and Path Planning in Detection Search," Ph.D. Dissertation, University of Maryland, College Park, 1987.
- [12] N. J. Nilsson, "Principles Of Artificial Intelligence," Tioga Publishing Co., 1980.
- [13] A. Barr and E. A. Feigenbaum (editors), "The Handbook Of Artificial Intelligence," William Kaufmann, Inc., 1981.
- [14] I. Yan and G. L. Blankenship, "Search For Randomly Moving Targets I : Estimation," presented in 1987 Command and Control Research Symposium and submitted for Proceedings, Washington, DC, June, 1987.