

PATH CONSTRAINED SEARCH PROBLEM WITH REWARD CRITERION

Ryūsuke Hohzaki Kōji Iida
National Defense Academy

(Received August 24, 1993; Revised March 24, 1994)

Abstract A target is moving on a finite number of cells in discrete time. Knowing the probabilities of the target's path selection, a searcher is searching for the target in this search space with constraints that he can move from cell i to one of the adjacent cells. He gains a value $V(t)$ on the detection of the target at time t but expends cost $c_0(i, t)$ for the search in cell i at t . In this paper, we propose a method to find an optimal path for the searcher, which maximizes the expected reward defined as the expected value minus the expected cost. We use a branch and bound procedure with an upper bound estimation given by solving the problem relaxed in continuous search effort.

1. Introduction

The path constrained search problem in discrete time, space and search effort is NP-complete even in the relatively simple problem of maximizing the detection probability, as proved by Trummel and Weisinger[5]. Stewart[4] proposed a method to solve this problem using the branch and bound method. However, his estimation of the lower bound is not necessarily optimal because the algorithm of Brown[1] applied by Stewart does not necessarily give an optimal solution to the problem in discrete search efforts. Eagle and Yee[2] proposed first a method for the optimal solution using the precise estimation of the lower bound.

The purpose of this paper is to extend the problem to the one with a reward criterion, where a searcher gets a value on the detection of the target and loses cost by distributing search efforts. The reward is one of important criteria, by which we can deal with practical OR problems from the view point of the cost-performance. As an example of this search situation, we can take the search-and-rescue work by an airplane (the searcher) on the sea. At the first stage of the search, the possibilities about the targets' movement are estimated. The purpose of the search-and-rescue is to find living lost persons (the targets) as early as possible and therefore, the value of the targets is considered to depend only on the time of detection and to become lower as time elapses. The place to which the searcher can move next time is limited by his mobility. The searcher consumes valuable oil or valuable time by the search operation. We are interesting in optimizing such a search-and-rescue operation in terms of the cost-performance. To this problem, we make use of the branch and bound procedure and propose a method to give an optimal solution, in which an upper bound of the objective value is estimated by solving a relaxed problem with continuous effort.

We formulate a mathematical model of this problem in the next section. In Section 3, the branch and bound method and the estimation of the upper bound of the expected reward are proposed. In Section 4, we examine the sensitivity of the optimal solution to parameters in the problem and the computational time by some numerical examples.

2. Modeling and Formulation of the Problem

In this section, we present assumptions of our model and formulate the model in an integer programming problem. Assumptions of our search problem are as follows:

- (1) A target moves in a search space consisting of a finite number of search cells $K = \{1, \dots, K\}$ in discrete time $T = \{1, \dots, T\}$. A path ω of the target is defined as a sequence of cells $\{\omega(t), t = 1, 2, \dots, T\}$, where $\omega(t)$ is the cell which the target visits at time t . The target selects a path probabilistically from a finite set Ω of possible paths at the beginning of the search, and moves along the path after that. The probability $\pi(\omega)$ with which the target selects path ω is assumed to be known to the searcher, where $\sum_{\omega \in \Omega} \pi(\omega) = 1$ holds.
- (2) A searcher makes efforts to detect the target. The searcher starts from cell s at the beginning $t = 0$. He moves in the search space K but cells to which he can move from cell i are bound within the adjacent cells $I(i)$. He is obliged to look the cell where he is.
- (3) The conditional probability of detecting the target given that the target is in cell $i \in K$ and the searcher looks in the cell is $p_i = 1 - \exp(-\alpha_i)$ where $\alpha_i \geq 0$. Event of the detection or the non-detection of the target at time $t + 1$ given no detection until t is independent of the past looks.
- (4) The searcher gains value $V(t)$ on the detection of the target at time t but expends cost $c_0(i, t)$ per look in cell i at time t . $V(t)$ is assumed to be a non-increasing function of t and not to depend on the cell at which the detection is taken place.
- (5) The searcher selects his path optimally so as to maximize the expected reward, which is defined as the expected value minus the expected cost over the whole time T .

Let $\varphi(i, \tau)$ be a 0-1 variable such that $\varphi(i, \tau) = 1$ if the searcher is in cell i at time τ and $\varphi(i, \tau) = 0$ otherwise. $\varphi(i, \tau)$ is also interpreted as the variable representing the allocation of search efforts. For the allocation of search efforts $\{\varphi(i, \tau)\}$, the total amount of cost $C(\tau, \varphi)$ and the detection probability $P^\tau(\varphi)$ during time period $[1, \tau]$ are given by the next equations.

$$(2.1) \quad C(\tau, \varphi) = \sum_{\xi=1}^{\tau} \sum_{i=1}^K c_0(i, \xi) \varphi(i, \xi),$$

$$(2.2) \quad P^\tau(\varphi) = 1 - \sum_{\omega \in \Omega} \pi(\omega) \exp \left(- \sum_{\xi=1}^{\tau} \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right).$$

The searcher can detect the target at time τ with probability $P^\tau(\varphi) - P^{\tau-1}(\varphi)$. If the detection occurs at the time, he gains value $V(\tau)$ but has expended cost $C(\tau, \varphi)$ so far. If he can not detect the target until terminating time T , he just loses cost $C(T, \varphi)$. Therefore, the expected reward over $[1, T]$ is given by

$$(2.3) \quad \begin{aligned} R^T(\varphi) &= \sum_{\tau=1}^T (V(\tau) - C(\tau, \varphi)) (P^\tau(\varphi) - P^{\tau-1}(\varphi)) - C(T, \varphi) (1 - P^T(\varphi)) \\ &= V(T) P^T(\varphi) + \sum_{\tau=1}^{T-1} (\Delta C(\tau, \varphi) - \Delta V(\tau)) P^\tau(\varphi) - C(T, \varphi) \end{aligned}$$

$$\begin{aligned}
&= V(T) \left\{ 1 - \sum_{\omega} \pi(\omega) \exp\left(-\sum_{\tau=1}^T \alpha_{\omega(\tau)} \varphi(\omega(\tau), \tau)\right) \right\} \\
&\quad + \sum_{\tau=1}^{T-1} (\Delta C(\tau, \varphi) - \Delta V(\tau)) \left\{ 1 - \sum_{\omega} \pi(\omega) \exp\left(-\sum_{\xi=1}^{\tau} \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi)\right) \right\} \\
&\quad - C(T, \varphi),
\end{aligned}$$

where

$$(2.4) \quad P^0(\varphi) = 0,$$

$$(2.5) \quad \Delta C(\tau, \varphi) = C(\tau + 1, \varphi) - C(\tau, \varphi) = \sum_{i=1}^K c_0(i, \tau + 1) \varphi(i, \tau + 1),$$

$$(2.6) \quad \Delta V(\tau) = V(\tau + 1) - V(\tau).$$

Consequently, this path constrained search problem is formulated as an integer programming problem as follows:

$$(2.7) \quad \max_{\varphi} \quad R^T(\varphi)$$

subject to:

$$(2.8) \quad \varphi(s, 0) = 1,$$

$$(2.9) \quad \sum_{i=1}^K \varphi(i, \tau) = 1, \quad \tau = 0, \dots, T,$$

$$(2.10) \quad \varphi(i, \tau) \leq \sum_{j \in I(i)} \varphi(j, \tau + 1), \quad i = 1, \dots, K, \quad \tau = 0, \dots, T - 1,$$

$$(2.11) \quad \varphi(i, \tau) = \{0, 1\}, \quad i = 1, \dots, K, \quad \tau = 1, \dots, T.$$

Equations (2.8) and (2.9) imply that the searcher starts from cell s at time 0 and that he is obliged to look in a certain cell at each time. Search effort at time 0 is tentative and has no effect on the detection of the target. If the searcher looks in cell i at time τ , namely, $\varphi(i, \tau) = 1$, the searcher must go to one of its adjacent cells $I(i)$ at the next time $\tau + 1$, which is represented by Constraint (2.10).

3. A Branch and Bound Method and a Relaxed Problem

We adopt such a branch and bound method as proposed by Stewart[4]. In order to manipulate the method enough efficiently, we use an optimal solution of a relaxed problem as the upper bound of the objective function. The relaxed problem is generated from the original problem by ignoring Eqs.(2.10) and (2.11).

3.1 A branch and bound method

A whole set of feasible paths of the searcher is represented by an enumeration tree as illustrated in Fig.1. The tree has the starting cell s at time $\tau = 0$ as the root and spreads downwards as time passes. A node $\{j, t\}$ denoting a combination of cell j and time t branches to a set of adjacent nodes denoting combinations of $I(j)$ and time $t + 1$. The branching is conducted until time T and the whole tree represents all options of searcher's paths.

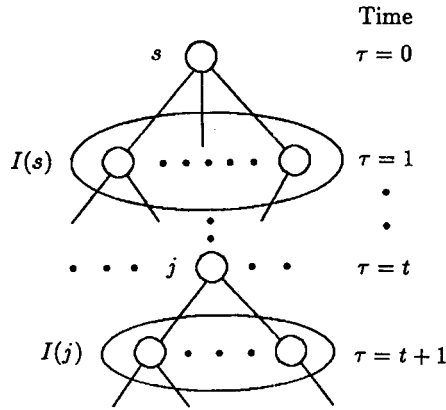


Figure 1. Enumeration tree.

A sequential chain of nodes from time 1 to time t on this tree corresponds to a path of the searcher during $[1, t]$ and therefore determines uniquely the value of $\{\varphi(i, \tau), i \in K, \tau = 1, \dots, t\}$. We can evaluate the posterior probability $\Lambda_t(\omega, \varphi)$ that the target takes path ω given no detection until t , which means the probability that the target starts at time $t + 1$ along path ω .

$$(3.1) \quad \Lambda_t(\omega, \varphi) = \frac{\pi(\omega) \exp(-\sum_{\tau=1}^t \alpha_{\omega(\tau)} \varphi(\omega(\tau), \tau))}{1 - P^t(\varphi)}.$$

Here, we partition the whole search in two stages. The first stage is the search during $[1, t]$ and the second stage is the search during $[t + 1, T]$. Furthermore, we generalize the definitions (2.1) ~ (2.4) to these partial stages of the search. For the search during $[a, b]$, we suppose that the target selects path $\omega \in \Omega$ by the probability $\Lambda_{a-1}(\omega, \varphi)$ at the initial search time a and the searcher uses an effort distribution $\{\varphi(j, \tau), j \in K, \tau \in [a, b]\}$. The cumulative cost $C_a(\tau, \varphi)$, the detection probability $P_a^r(\varphi)$ during $[a, \tau]$, where $a \leq \tau \leq b$, and the expected reward R_a^b during $[a, b]$ are evaluated as follows.

$$(3.2) \quad C_a(\tau, \varphi) = \sum_{\xi=a}^{\tau} \sum_{i=1}^K c_0(i, \xi) \varphi(i, \xi),$$

$$(3.3) \quad P_a^r(\varphi) = 1 - \sum_{\omega \in \Omega} \Lambda_{a-1}(\omega, \varphi) \exp\left(-\sum_{\xi=a}^{\tau} \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi)\right),$$

$$(3.4) \quad R_a^b(\varphi) = \sum_{\tau=a}^b (V(\tau) - C_a(\tau, \varphi))(P_a^r(\varphi) - P_a^{r-1}(\varphi)) - C_a(b, \varphi)(1 - P_a^b(\varphi)) \\ = V(b)P_a^b(\varphi) + \sum_{\tau=a}^{b-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))P_a^r(\varphi) - C_a(b, \varphi),$$

where

$$P_a^{a-1}(\varphi) = 0, \\ \Lambda_0(\omega, \varphi) = \pi(\omega).$$

By a certain chain of nodes from time 1 to time t on the above enumeration tree, we can calculate the detection probability $P_1^t(\varphi)$ and the expected reward $R_1^t(\varphi)$. The expected reward in two stages of the search during $[1, t]$ and during $[t + 1, T]$ have the next relation.

Theorem 1. (Partitioning the expected reward)

$$(3.5) \quad R_1^T(\varphi) = R_1^t(\varphi) + (1 - P_1^t(\varphi))R_{t+1}^T(\varphi).$$

Proof. By Eqs.(3.1) ~ (3.3), we have the following equations.

$$P_{t+1}^T(\varphi) = 1 - \sum_{\omega \in \Omega} \Lambda_t(\omega, \varphi) \exp \left(- \sum_{\xi=t+1}^T \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right) = \frac{P_1^T(\varphi) - P_1^t(\varphi)}{1 - P_1^t(\varphi)},$$

$$C_{t+1}(T, \varphi) = C_1(T, \varphi) - C_1(t, \varphi).$$

The right-hand side of Eq.(3.5) is transformed by substituting the above and Eq.(3.4) into Eq.(3.5) as follows.

$$\begin{aligned} & R_1^t(\varphi) + (1 - P_1^t(\varphi))R_{t+1}^T(\varphi) \\ &= V(t)P_1^t(\varphi) + \sum_{\tau=1}^{t-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))P_1^\tau(\varphi) - C_1(t, \varphi) \\ & \quad + (1 - P_1^t(\varphi)) \cdot \left\{ V(T)P_{t+1}^T(\varphi) + \sum_{\tau=t+1}^{T-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))P_{t+1}^\tau(\varphi) - C_{t+1}(T, \varphi) \right\} \\ &= V(t)P_1^t(\varphi) + \sum_{\tau=1}^{t-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))P_1^\tau(\varphi) - C_1(t, \varphi) \\ & \quad + V(T)(P_1^T(\varphi) - P_1^t(\varphi)) + \sum_{\tau=t+1}^{T-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))(P_1^\tau(\varphi) - P_1^t(\varphi)) \\ & \quad - (1 - P_1^t(\varphi))(C_1(T, \varphi) - C_1(t, \varphi)) \\ &= V(T)P_1^T(\varphi) + \sum_{\tau=1}^{T-1} (\Delta C(\tau, \varphi) - \Delta V(\tau))P_1^\tau(\varphi) - C_1(T, \varphi) = R_1^T(\varphi). \end{aligned}$$

Q.E.D.

From the above theorem, if we estimate an upper bound $\tilde{R}_{t+1}^T(\varphi)$ of $R_{t+1}^T(\varphi)$, an upper bound $\tilde{R}_1^T(\varphi)$ of the expected reward $R_1^T(\varphi)$ over the whole time $[1, T]$ can be obtained by

$$(3.6) \quad \tilde{R}_1^T(\varphi) = R_1^t(\varphi) + (1 - P_1^t(\varphi))\tilde{R}_{t+1}^T(\varphi).$$

An estimation $\tilde{R}_{t+1}^T(\varphi)$ will be discussed in the next section. Using the method of Stewart[4], we can itemize the branch and bound procedure to give an optimal solution as follows. The maximum value of the objective function on the way of the procedure is stored in \tilde{R} .

- (Step1) Let $t = 0, \sigma(t) = s, P_1^t(\varphi) = 0, R_1^t(\varphi) = 0, \Lambda_t(\omega, \varphi) = \pi(\omega)$, and $\tilde{R} = -\infty$. Set $K[0, \sigma(0)] = I(s)$.
- (Step2) If $\tilde{R} = -\infty$, go to Step5. Otherwise, estimate $\tilde{R}_{t+1}^T(\varphi)$ by $\Lambda_t(\omega, \varphi)$ and $\tilde{R}_1^T(\varphi)$ by Eq.(3.6). If $\tilde{R}_1^T(\varphi) > \tilde{R}$, go to Step5. Otherwise, go to Step3 because a further forward branching does not bring any better solutions.
- (Step3) If $t = 0$, terminate. The current stored value is optimal. If $t \neq 0$, go to Step4.
- (Step4) Delete $\sigma(t)$ from $K[t-1, \sigma(t-1)]$ and let $t = t-1$. If $K[t, \sigma(t)] = \emptyset$, go to Step3. Otherwise, go to Step5.
- (Step5) Select a cell $\sigma(t+1)$ from $K[t, \sigma(t)]$ and let $t = t+1$. If $t < T$, let $K[t, \sigma(t)] = I(\sigma(t))$ and go to Step2. If $t = T$, go to Step6.

(Step6) If $\tilde{R} > R_1^T(\varphi)$, the current chain of nodes during $[1, T]$ is not an optimal searcher's path and go to Step4. Otherwise, set $\tilde{R} = R_1^T(\varphi)$ because the current chain gives the current best path and go to Step4.

3.2 Estimation of the upper bound of $R_{t+1}^T(\varphi)$

The upper bound $\tilde{R}_{t+1}^T(\varphi)$ could be estimated by solving a relaxed problem. The relaxed problem is defined by deleting Constraints (2.10) and (2.11) and adding the continuity of $\{\varphi(i, \tau)\}$.

$$(3.7) \quad \max_{\varphi} R_{t+1}^T(\varphi)$$

subject to:

$$(3.8) \quad \sum_{i=1}^K \varphi(i, \tau) = 1, \quad \tau = t+1, \dots, T,$$

$$(3.9) \quad \varphi(i, \tau) \geq 0, \quad i = 1, \dots, K, \quad \tau = t+1, \dots, T.$$

Iida and Hozaki[3] found a necessary condition for the optimality in the case that an equality of (3.8) is replaced by an inequality. By following their way, we can obtain a necessary condition of the optimality for the above relaxed problem. A symbol Ψ denotes a whole family of sets of search efforts allocations satisfying Eqs.(3.8) and (3.9).

Theorem 2. If $\{\varphi^*(i, \tau), i \in K, \tau \in [t+1, T]\} \in \Psi$ is optimal for the problem defined by (3.7) \sim (3.9), there exists some $\{\lambda(\tau), \tau = t+1, \dots, T\}$ satisfying

$$(3.10) \quad \varphi^*(i, \tau) > 0 \implies A_{i\tau}^{t+1 T}(\varphi^*) = \lambda(\tau),$$

$$(3.11) \quad \varphi^*(i, \tau) = 0 \implies A_{i\tau}^{t+1 T}(\varphi^*) \leq \lambda(\tau),$$

where

$$(3.12) \quad A_{i\tau}^{t+1 T}(\varphi) = \sum_{\omega} \Lambda_t(\omega, \varphi) \alpha_i \left\{ V(T) \exp \left(- \sum_{\xi=t+1}^T \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right) + \sum_{\zeta=\tau}^{T-1} (\Delta C(\zeta, \varphi) - \Delta V(\zeta)) \exp \left(- \sum_{\xi=t+1}^{\zeta} \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right) \right\} \delta_{i\omega(\tau)} - c_0(i, \tau) (1 - P_{t+1}^{\tau-1}(\varphi))$$

and $\delta_{..}$ is the Kronecker's delta.

$A_{i\tau}^{t+1 T}(\varphi)$ can be transformed as follows.

$$(3.13) \quad A_{i\tau}^{t+1 T}(\varphi) = B_{i\tau} \cdot \exp(-\alpha_i \varphi(i, \tau)) - c_0(i, \tau) (1 - P_{t+1}^{\tau-1}(\varphi)),$$

$$(3.14) \quad B_{i\tau} = \alpha_i \sum_{\omega} \Lambda_t(\omega, \varphi) \left\{ V(T) \exp \left(- \sum_{\substack{\xi=t+1 \\ \xi \neq \tau}}^T \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right) + \sum_{\zeta=\tau}^{T-1} (\Delta C(\zeta, \varphi) - \Delta V(\zeta)) \exp \left(- \sum_{\substack{\xi=t+1 \\ \xi \neq \tau}}^{\zeta} \alpha_{\omega(\xi)} \varphi(\omega(\xi), \xi) \right) \right\} \delta_{i\omega(\tau)}.$$

If $\{\varphi(j, \zeta), j \in K, \zeta \in [t+1, T], \zeta \neq \tau\}$ is kept constant, $A_{i\tau}^{t+1 T}(\varphi)$ could be considered as a function of $\varphi(i, \tau)$ and be denoted by $\rho_{i\tau}(\varphi(i, \tau))$ simply. It is obvious by Eq.(3.13) that $\rho_{i\tau}(\varphi(i, \tau))$ is strictly decreasing and the next corollary is established.

Corollary 1. For every $\tau \in [t+1, T]$, there exists a finite $\lambda(\tau)$ such as

$$(3.15) \quad \sum_{i=1}^K [\rho_{i\tau}^{-1}(\lambda(\tau))]^+ = 1.$$

Then an optimal solution $\varphi^*(i, \tau)$ is given by

$$(3.16) \quad \varphi^*(i, \tau) = [\rho_{i\tau}^{-1}(\lambda(\tau))]^+.$$

$[z]^+$ implies 0 if $z < 0$ and z if $z \geq 0$. $\rho_{i\tau}^{-1}(\cdot)$ is the inverse function of $\rho_{i\tau}(\cdot)$.

Proof. If $\varphi^*(i, \tau) > 0$, then we have $0 < \varphi^*(i, \tau) = \rho_{i\tau}^{-1}(\lambda(\tau))$ from Eq.(3.10). If $\varphi^*(i, \tau) = 0$, then we have $0 = \varphi^*(i, \tau) \geq \rho_{i\tau}^{-1}(\lambda(\tau))$ from Eq.(3.11) and the strictly decreasing property of $\rho_{i\tau}(\cdot)$. In both cases, $\varphi^*(i, \tau)$ can be expressed in the form of (3.16). Now let

$$(3.17) \quad \bar{\lambda} = \max_i \rho_{i\tau}(0) = \max_i \{B_{i\tau} - c_0(i, \tau) (1 - P_{i+1}^{\tau-1}(\varphi))\},$$

$$(3.18) \quad \underline{\lambda} = \max_i \{-c_0(i, \tau) (1 - P_{i+1}^{\tau-1}(\varphi))\}.$$

Because $\lim_{z \rightarrow \infty} \rho_{i\tau}(z) = -c_0(i, \tau) (1 - P_{i+1}^{\tau-1}(\varphi))$, for $\bar{\lambda}$ and $\underline{\lambda}$,

$$(3.19) \quad \sum_i [\rho_{i\tau}^{-1}(\bar{\lambda})]^+ = 0 < 1 < \sum_i [\rho_{i\tau}^{-1}(\underline{\lambda})]^+ = \infty.$$

Hence, by the continuity of $\rho_{i\tau}^{-1}(\cdot)$, there exists $\lambda(\tau)$ so that Eq.(3.15) is satisfied for $\bar{\lambda} > \lambda(\tau) > \underline{\lambda}$.

Q.E.D.

Considering that $B_{i\tau}$ is calculated by a search effort allocation $\{\varphi(i, \zeta), i \in K, \zeta \in [t+1, T], \zeta \neq \tau\}$, we can obtain $\rho_{i\tau}^{-1}(\cdot)$ in an analytical form. Consequently, if an optimal effort allocation $\{\varphi(i, \zeta), i \in K, \zeta \in [t+1, T], \zeta \neq \tau\}$ is provided for each time except for τ , an optimal search effort $\{\varphi(i, \tau), i \in K\}$ at time τ can be also evaluated in the following procedure.

First obtain $\lambda(\tau)$ satisfying Eq.(3.15) by some search method, e.g., a binary search among an interval $[\underline{\lambda}, \bar{\lambda}]$, and determine the optimal effort $\{\varphi(i, \tau), i \in K\}$ by the next formula.

$$(3.20) \quad \varphi(i, \tau) = \left[\frac{1}{\alpha_i} \log \frac{B_{i\tau}}{\lambda(\tau) + c_0(i, \tau) (1 - P_{i+1}^{\tau-1}(\varphi))} \right]^+.$$

We suppose that an optimal effort allocation for all time except for τ is given. Practically, we start from an arbitrary allocation of search efforts and make it converge to an optimal allocation asymptotically by iterating the above procedure for each $\tau = t+1, \dots, T$. This convergence is attainable as shown by Iida and Hohzaki[3]. This kind of asymptotical numerical methods are named as the FAB algorithm which Brown[1] used first and Washburn[6] investigated in studies on the maximizing problem of the detection probability.

4. Numerical Examples

We analyze several numerical examples in order to see the sensitivity of the optimal search path to parameters in this search model. Furthermore, the performance of the upper bound estimation proposed in Section 3 is evaluated from a computational point of view.

4.1 Sensitivity of the optimal search path

Consider the target and the searcher moving on one-dimensional search cells $K = \{1, 2, \dots, 7\}$ in discrete time $T = \{1, 2, \dots, 7\}$. There are three paths to be selected by the target as illustrated in Fig.2. Paths 1 and 2 cross each other at time 4 and separate at time 6. Path 3 is far away from other two paths after time $t = 1$.

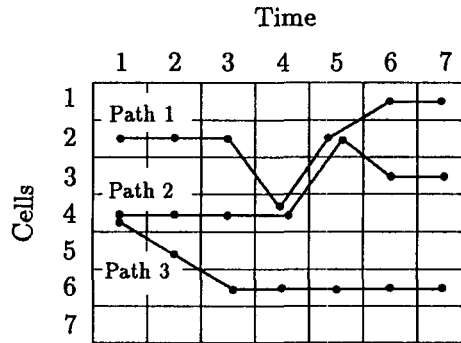


Figure 2. Paths of the target.

Now the searcher starts from Cell 7 at time $t = 0$ and thereafter moves from a cell to one of its neighborhood cells. Namely,

$$(4.1) \quad I(1) = \{1, 2\}, I(2) = \{1, 2, 3\}, \dots, I(k) = \{k-1, k, k+1\}, \dots, I(7) = \{6, 7\}.$$

Note that the searcher can not be in both of Cell 4 at time 4 and Cell 2 at time 5.

Common parameters used in the following examples are

$$\alpha_i = 0.2, \quad c_0(i, t) = 1.0, \quad i \in K, \quad t \in T.$$

(1) Sensitivity to the probability of the path selection

A value $V(t)$ is assumed to remain constant.

$$V(t) = 20, \quad t \in T.$$

Consider three cases with different probabilities of path selection of the target.

Case1: $\pi(1) = \pi(2) = 0.4, \pi(3) = 0.2$

Case2: $\pi(1) = \pi(2) = 0.3, \pi(3) = 0.4$

Case3: $\pi(1) = \pi(2) = 0.25, \pi(3) = 0.5$

The optimal search paths and the optimal rewards are presented in Table 1.

Table 1. Optimal search paths for Case 1, 2 and 3

| Case \ t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Reward |
|----------|---|---|---|---|---|---|---|--------|
| Case1 | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 0.766 |
| Case2 | 6 | 5 | 4 | 4 | 5 | 6 | 6 | 0.611 |
| Case3 | 6 | 5 | 6 | 6 | 6 | 6 | 6 | 1.061 |

Since parameters $\alpha_i, c_0(i, t)$ and $V(t)$ remain constant for $i \in K, t \in T$ and the number of looks is constrained by Eq.(2.9), the searcher wants to look into the cell where the probability of the existence of the target is high and the high reward is expected. Hence, the searcher should chose a path covering $\{time, cell\} = \{5, 2\}$ or $\{4, 4\}$ such as path

$\{\sigma(1), \sigma(2), \dots, \sigma(7)\} = \{6, 5, 4, 3, 2, 1, 1\}$ or $\{6, 5, 4, 4, 3, 3, 3\}$ in Case 1. However, the latter path is not optimal because the existence probability of the target on Path 2 becomes low at times 6 and 7 after the first looks at times 3 and 4. Actually, the movement along this path results in the reward 0.496.

Path 3 becomes more attractive for the searcher in the order of Cases 1,2,3 because the existence probability of the target on Path 3 increases in the same order. This attractiveness has much effect on the optimal path as shown in Table 1.

(2) Sensitivity to the target's value

Consider Case 4 where the probability of the path selection is the same as Case 1 but the value $V(t)$ decreases as time passes.

$$V(t) = 20 - 3(t - 4), \quad t = 1, 2, \dots, 7.$$

In this case, an optimal search path and the optimal reward are presented in Table 2.

Table 2. An optimal search path for Case 4

| Case \ t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Reward |
|------------|---|---|---|---|---|---|---|--------|
| Case4 | 6 | 5 | 4 | 4 | 3 | 3 | 3 | 0.281 |

The searcher can expect that the detection probability is higher for the optimal path of Case 1 than the path in the above table. However, he should select the latter path since the early detection is preferable from the view of the expected value.

4.2 Performance of the upper bound estimation

The total number of nodes generated on the enumeration tree in Fig.1 approximately grows larger by the order of the T -th power of K . The branch and bound method using the upper bound estimation (3.6) aims to decrease the number of nodes to be searched in the calculation process and to reduce its computational time in consequence. It takes some additional computational time to estimate the upper bound. Computational time for solving the problem is determined from the balance between the time-decreasing by saving the number of searched nodes and the time-expenditure by the estimation. We need to evaluate the effectiveness of the proposed estimation on computational time. In the next example, computational time is examined for various terminating times.

Let the number of cells K be 5 and the number of target's paths Ω be 4. Other parameters are kept constant.

$$\alpha_i = 0.2, \quad V(t) = 20, \quad c_0(i, t) = 1.0, \quad i \in K, \quad t \in T.$$

Four paths are generated randomly as follows. The target randomly selects one cell from K cells for each time and gains a sequential list of selected cells, namely, a path. This process is repeated four times to get 4 paths. Each path is selected by the target with an equal probability, namely $\pi(\omega) = 0.25$. The starting cell of the searcher is selected randomly too. The movement of the searcher is restricted to the neighborhood cells as stated before in Eq.(4.1). In this example, terminating time T varies from 2 through 12. Ten problems are generated for each terminating time and solved to measure CPU-time. Results are mentioned as the mean value. A mainframe computer M-680H(OS:VOS3) and a computer language FORTRAN 77 are used in this calculation.

Let N be the number of practical searched nodes on the enumeration tree by the solution method and A be the total number of nodes on the tree. Table 3 shows CPU-time(sec.) and the ratios N/A . For comparison, we tabulate data for the total enumerating method

which calculates rewards for the whole options of paths on the enumeration tree and takes a maximum one. The above ratio N/A is always one for the total enumerating method and omitted in the table. Circles in Fig.3-A and Fig.3-B denote logarithms of CPU-time and of the ratios, respectively, to the base two. The axis of abscissa indicates terminating time T . In Fig.3-A, results for the total enumerating method are shown by square symbols to compare with the performance of the upper bound estimation.

Table 3. CPU-time for varying terminating time

| T | The bound estimation | | The enumerating method |
|----|-----------------------|-----------------------|------------------------|
| | CPU-time(sec.) | N/A | CPU-time(sec.) |
| 2 | 4.74×10^{-4} | 7.37×10^{-1} | 2.20×10^{-4} |
| 4 | 9.96×10^{-3} | 3.79×10^{-1} | 3.10×10^{-3} |
| 6 | 8.12×10^{-2} | 1.47×10^{-1} | 3.11×10^{-2} |
| 8 | 1.57×10^{-1} | 5.37×10^{-2} | 3.37×10^{-1} |
| 10 | 5.47×10^{-1} | 1.97×10^{-2} | $3.05 \times 10^{+0}$ |
| 12 | $1.05 \times 10^{+0}$ | 2.52×10^{-3} | $3.57 \times 10^{+1}$ |

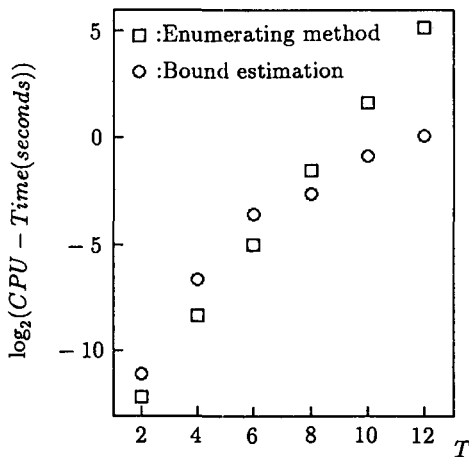


Figure 3-A. CPU-Time.

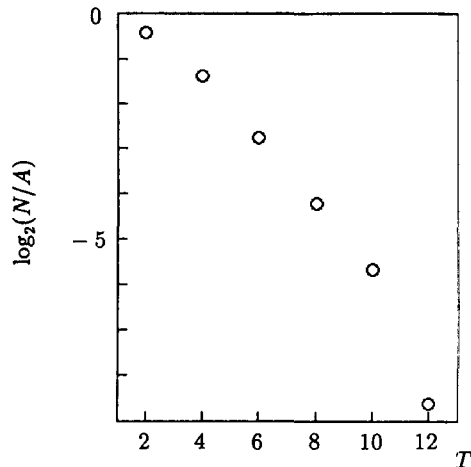


Figure 3-B. The ratio of the number of searched nodes.

In the total enumerating method, its procedure mainly consists of the calculation of $R_1^T(\varphi)$ as many times as the number of terminal nodes on the enumeration tree and the comparison among them. Therefore, CPU-time is estimated to be proportional to the number of the terminal nodes. The number of the terminal nodes increases approximately by the T -th power of K . That is the reason why squares in Fig.3-A increase linearly on the logarithmic scale. On the other hand, the increasing of results for the proposed upper bound estimation has a concave form. Therefore, we may say that the estimation is superior to the total enumerating method for the terminating time of 8 or more in spite of the inferiority for the time of 6 or less. The concave form is caused from the decreasing of the ratios N/A as shown in Fig.3-B.

5. Concluding Remarks

This paper deals with a path constrained search problem with a reward criterion. This problem is clearly NP-complete in discrete time, search space and search effort. We proposed

a method to give an optimal solution using the branch and bound procedure. The method accompanies the upper bound estimation of the expected reward which produces a good effect on computational time of solving problems especially in the case of larger terminating time of the search.

We think that search problems with many searchers or many targets may be located on the extension of this study.

Acknowledgments

The authors would like to express their gratitude to the anonymous referees for careful reading of this paper and helpful comments.

References

- [1] Brown, S.S. : Optimal Search for a Moving Target in Discrete Time and Space. *Operations Research*, Vol.28, No.6 (1980), 1275–1286.
- [2] Eagle, J.N. and Yee, J.R. : An Optimal Branch-and-Bound Procedure for the Constrained Path – Moving Target Search Problem. *Operations Research*, Vol.38, No.1 (1990), 110–114.
- [3] Iida, K. and Hohzaki, R. : The Optimal Search Plan for a Moving Target Minimizing the Expected Risk. *Journal of the Operations Research Society of Japan*, Vol.31, No.3 (1988), 294–320.
- [4] Stewart, T.J. : Experience with a Branch-and-Bound Algorithm for Constrained Searcher Motion. *Search Theory and Applications*, 247–253, Plenum Press, New York, 1980.
- [5] Trummel, K.E. and Weisinger, J.R. : The Complexity of the Optimal Searcher Path Problem. *Operations Research*, Vol.34, No.2 (1986), 324–327.
- [6] Washburn, A.R. : Search for a Moving Target: The FAB Algorithm. *Operations Research*, Vol.31, No.4 (1983), 739–751.

Ryusuke HOHZAKI:
Department of Applied Physics,
National Defense Academy,
1-10-20 Hashirimizu, Yokosuka,
Kanagawa 239, Japan
E-mail: hozaki@JPNND.ABITNET