

구역분할 알고리즘을 이용한 다수 탐색플랫폼의 구역할당 방법

안우선^{*,1),2)} · 조윤철¹⁾ · 이찬선³⁾

¹⁾ 해군 전력분석시험평가단

²⁾ (주)심네트 기술경영분석연구소

³⁾ 국방과학연구소 분석평가실

A Methodology for Partitioning a Search Area to Allocate Multiple Platforms

Woosun An^{*,1),2)} · Yunchol Cho¹⁾ · Chansun Lee³⁾

¹⁾ Test & Evaluation Group, Naval Force Analysis, Korea

²⁾ Institute for Technology & Management Analysis, SIMNET Inc., Korea

³⁾ Analysis and Evaluation Center, Agency for Defense Development, Korea

(Received 14 September 2017 / Revised 3 November 2017 / Accepted 23 February 2018)

ABSTRACT

In this paper, we consider a problem of partitioning a search area into smaller rectangular regions, so that multiple platforms can conduct search operations independently without requiring unnecessary coordination among themselves. The search area consists of cells where each cell has some prior information regarding the probability of target existence. The detection probability in particular cell is evaluated by multiplying the observation probability of the platform and the target existence probability in that cell. The total detection probability within the search area is defined as the cumulative detection probability for each cell. However, since this search area partitioning problem is NP-Hard, we decompose the problem into three sequential phases to solve this computationally intractable problem. Additionally, we discuss a special case of this problem, which can provide an optimal analytic solution. We also examine the performance of the proposed approach by comparing our results with the optimal analytic solution.

Key Words : Search Problem(탐색문제), Search Area Partiton(탐색구역분할), Anti-Submarine Warfare(대잠작전), Rectangular Partition(직사각형분할)

1. 서론

잠수함은 수중에서 은밀하게 공격대상 표적에 접근한 후 공격하여 손상을 줄 수 있는 함정으로 정보수집, 기뢰부설, 특수부대 침투 등 다양한 작전을 수행하는데 용이하며, 필요시 기습공격을 통해 고가치 수

* Corresponding author, E-mail: woosunan@gmail.com
Copyright © The Korea Institute of Military Science and Technology

상 또는 육상표적에 막대한 피해를 줄 수 있는 비대칭 핵심전력이다. 현재 북한은 우리군에 비해 절대 우위의 잠수함 전력을 유지하고 있으며, SLBM 탑재 잠수함 건조 및 구형 잠수함의 개조를 통해 잠항시간을 늘리고 자체소음을 감소하는 등 전력 최신화를 꾸준히 추진하고 있다. 북한이 개량한 잠수함은 피탐이 어렵기 때문에, 한반도 전 해역에서 작전수행이 가능하며, 천안함 폭침을 통해 알려진 바와 같이 그 위협은 심각한 수준에 이른다. 또한, 2015년 목함지뢰 사태 발생 시 50여척의 북한 잠수함이 일시에 기지를 이탈하여 수중 위협을 전개한 상황에서 보듯이 북한은 우리군의 대응능력을 지속적으로 시험하고 있는 실정이다. 이에 따라 우리군도 대잠작전을 가장 중요하게 인식하고 작전수행 능력 향상에 매진하고 있으나, 보유하고 있는 현존 대잠전력으로 효과적인 작전수행을 하는 것은 매우 제한적임을 인식하고 있다.

따라서 이러한 환경에서는 제한된 가용전력의 효율적인 운용을 통한 효과적인 작전수행이 반드시 필요하다. 대잠정보를 지속적으로 수집하고 전파해야 하며, 이를 바탕으로 대잠전력의 효율적 배치를 통해 최단 시간에 위협을 탐지하고, 격멸하는 일련의 절차를 통합하고 체계화하는 것 또한 매우 중요하다. 그러나 다수의 플랫폼이 일정한 구역에서 동시에 대잠작전을 수행해야 하는 경우, 탐지장비의 간섭을 피하기 위해 지속적으로 플랫폼의 위치를 조정하는 것은 작전 효율성을 저하시키는 요인이 되므로, 각 플랫폼의 기동능력, 탐지성능을 고려하여 사전에 구역을 할당할 후 작전을 수행해야 할 것이다.

본 논문에서는 다수의 플랫폼을 이용하여 탐색작전을 수행하는 경우에, 전체 탐색구역을 직사각형 모양으로 분할한 후 각 플랫폼에 할당하는 문제를 다루며, 구성은 다음과 같다. 1장의 서론에 이어, 2장에서 기존 문헌을 고찰한다. 3장에서는 탐색구역 할당문제에 대한 해법을 제안하며, 구역할당 문제를 완화하여 해석적 해를 얻을 수 있는 특수한 경우를 소개한다. 또한 시뮬레이션을 통해 제안한 방법의 성능을 분석하여, 해석적 해와 비교하고 4장의 결론으로 본 논문을 마무리한다.

2. 기존 문헌 고찰

단일 플랫폼이 탐색구역에서 대잠작전을 수행중인

경우, Eagle^[1], Martins^[2], Lau^[3]은 플랫폼의 최적의 탐색경로를 찾는 문제를 연구하였다. 탐색구역은 단일 탐지표적이 존재할 확률을 포함하는 셀들로 구성되며, 탐색자와 탐지표적은 현재 위치하고 있는 셀에서 이웃한 셀들로만 이동이 가능한 것으로 가정하고, Branch and Bound 알고리즘을 통해 최적 탐색경로를 찾았다. 그러나 셀의 수가 증가함에 따라, 계산 복잡도는 지수함수적으로 증가하기 때문에, 최적의 해를 얻는 것이 매우 어려운 단점이 있다. Santos^[4], An^[5]은 단일 탐지표적에 대해 다수 탐색자가 최적 이동경로를 찾는 휴리스틱 알고리즘을 제안하였다. 또한, An^[6]은 단일 탐지표적에 대한 다수 플랫폼의 탐색작전 수행 시 탐지확률이 높은 셀을 중심으로 부분 탐색구역을 생성한 후, 각 플랫폼에 할당하는 방법을 제안하였다.

본 논문은 An^[6]이 연구한 구역할당 문제에 전체 탐색구역을 구성하는 모든 셀이 직사각형으로 분할된 탐색구역에 포함되어야 한다는 제약식을 추가한 확장된 문제를 다룬다. 직사각형 분할은, NP-Hard 문제로, 탐색구역을 구성하는 셀과 플랫폼의 수가 증가함에 따라 최적해를 구하는 것은 매우 어려운 것으로 알려져 있으며^[7], 이러한 문제를 해결하기 위해, 본 논문에서는 탐색구역 생성, 탐색구역 조정, 탐색구역 배치의 3단계 절차로 구성되는 휴리스틱 알고리즘을 제안한다. 또한 구역할당 문제를 완화한 후, Lagrange 방법을 이용하여 해석적 해를 손쉽게 얻을 수 있는 특수한 경우를 설명한다. 그와 같이 유도된 해는 탐색구역할당 문제의 upper-bound를 제공하므로, 본 논문에서 제안한 방법의 성능을 판단하는 중요한 기준이 된다.

3. 본 론

3.1 탐색구역 할당문제

이 절에서는 다수의 플랫폼을 이용하여 단일 탐지표적을 탐색하는 경우 전체 탐색구역을 직사각형으로 분할하여 각 플랫폼에 할당하는 구역할당 문제를 소개한다. 전체 탐색구역은 탐지표적이 존재할 확률을 내재한 셀들로 구성되며, 구역분할 시 탐지표적은 이웃한 셀로 이동하지 않는 것으로 가정한다.

Table 1에서 정의한 것처럼, 전체 탐색구역은 가로와 세로 $M \times N$ 개로 구성된 셀(g)들의 집합인 G 로 정의한다. 각각의 셀은 탐지표적이 존재할 확률을 가지며 ($PP(g)$, $g \in G$)로 정의한다. 그러한 경우 단일 셀에서

Table 1. Summary of notation

내용	기호
Number of horizontal cells of whole search area	M
Number of vertical cells of whole search area	N
Set of cells of whole search area $G = \{1, 2, \dots, G \}, G = MN$	G
Set of platforms, $S = \{1, 2, \dots, s, \dots, S \}$	S
Set of cells belonging to a search region assigned to platform $s \in S$	A_s
Target existence probability in cell $g \in G$	$PP(g)$
Search width of platform $s \in S$	w_s
Search velocity of platform $s \in S$	v_s
Search time of platform $s \in S$	τ_s
Probability for platform $s \in S$ to detect a target in search region A_s	$PD(A_s)$
Area of single cell	b
Set of search regions that are randomly generated $Q = \{1, 2, \dots, Q \}, Q = S $	Q
Set of cells belonging to search region $q \in Q$	A_q
Set of cells belonging to search region $q \in Q$ assigned to platform $s \in S$	A_{sq}
If platform $s \in S$ is assigned to search region $q \in Q, f_{sq} = 1$, otherwise $f_{sq} = 0$	f_{sq}

플랫폼 s 가 탐지표적을 탐지할 수 있는 확률($PD(g)$)은 플랫폼의 탐지능력과 $PP(g)$ 의 곱으로 나타낼 수 있으며, 다수의 셀로 구성된 탐색구역(A_s)에서 탐지표적을 탐지할 확률($PD(A_s)$)은 탐색구역을 구성하는 각 셀에 대한 탐지확률의 합으로 식 (1)과 같이 나타낼 수 있다.

$$PD(A_s) = \sum_{g \in A_s} PP(g) \left(1 - \exp\left(\frac{-w_s v_s \tau_s}{b|A_s|}\right) \right) \quad (1)$$

식 (1)에서 플랫폼의 탐지능력은 플랫폼의 탐색폭 (w_s), 탐색속력(v_s), 탐색시간(τ_s)에 비례하고, 탐색구역의 크기에 반비례하는 함수인 Koopman^[8] 공식을 적용한다. 탐색구역의 크기는 단일 셀의 넓이(b)와 탐색구역을 구성하는 셀들의 수($|A_s|$)의 곱으로 나타낸다. 본

논문에서 다루는 탐색구역 할당문제는 최적의 분할구역($A_s^*, \forall s \in S$)을 찾는 비선형 정수계획법 문제로 식 (2)와 같이 모델링 한다.

$$\begin{aligned} \max \sum_{s \in S} PD(A_s) & \quad (2) \\ \text{s.t. } A_s \cap A_k &= \emptyset, \forall s, k \in S; (s \neq k) & (a) \\ A_s &= \text{rectangular shape}, \forall s \in S & (b) \\ \sum_{s \in S} |A_s| &= MN & (c) \\ |A_s| &\geq 1, |A_s| = \text{integer}, \forall s \in S & (d) \end{aligned}$$

식 (2)의 목적함수는 각 플랫폼에 대한 $PD(A_s)$ 의 합이며, 전체 탐색구역을 분할하여 각 플랫폼에 할당하는 방법은 제약식에 포함된다. 제약식 2(a)는 플랫폼에 할당된 각각의 탐색구역은 서로 중첩되지 않아야 한다는 조건이며, 식 2(b)는 플랫폼에 할당된 탐색구역은 직사각형 모양으로 제한하는 조건이다. 또한, 식 2(c)는 각각의 탐색구역을 구성하는 셀들의 총합은 전체 탐색구역을 구성하는 셀들의 수와 동일해야 한다는 조건이고, 마지막으로 식 2(d)는 플랫폼에 할당한 탐색구역은 적어도 하나의 셀을 포함해야 한다는 조건이다. 그러나 식 (2)는 NP-Hard 문제로, 전체 탐색구역을 구성하는 셀과 플랫폼의 수가 증가하면 탐색구역을 분할하여 플랫폼을 배치할 수 있는 경우가 지수 함수적으로 증가하기 때문에, 최적의 해($A_s^*, \forall s \in S$)를 구하는 것은 매우 어려운 일이다. 따라서, 본 논문에서는 An^[6]이 제안한 방법을 확장하여, 다수의 플랫폼을 분할된 탐색구역에 할당하는 휴리스틱 알고리즘을 제안한다.

3.2 구역할당 알고리즘 소개

본 논문에서 제안하는 구역할당 알고리즘은 탐색구역 생성, 탐색구역 조정, 탐색구역 배치 단계로 구성되는 휴리스틱 방법이다. 탐색구역 생성 및 조정 단계에서 식 (2)의 제약식을 만족하도록 전체 탐색구역을 플랫폼의 수만큼 임의로 분할하여, 탐색구역($A_q, q \in Q$)을 생성한다. 탐색 구역이 주어진 경우 식 (2)는 각각의 플랫폼을 탐색구역에 할당하는 일대일 배치(one-to-one assignment)^[9] 문제가 되며, 탐색구역 배치 단계에서 일대일 배치 알고리즘을 이용하여 각 플랫폼의 탐지확률의 합이 최대가 되도록 플랫폼을 탐색구역에 할당한다. 구역할당 알고리즘은 이러한 3단계 절차의 반복 수행을 통해, 매번 산출되는 탐지확률을 이전에

산출된 탐지확률과 비교하여, 점차 향상된 탐지확률을 얻는 방법이다.

3.2.1 탐색구역 생성

탐색구역 생성 단계는 $M \times N$ 개의 셀들로 구성된 전체 탐색구역에서 플랫폼의 수만큼 임의의 단일 셀들을 선택한 후, 직사각형 모양을 유지하고, 확장하면서 임의의 구역을 생성하는 단계이다. Fig. 1은 단일 셀의 넓이가 b 이고, 가로와 세로를 구성하는 셀의 수가 12×12 인 전체 탐색구역에서 7개의 플랫폼에 대한 탐색구역을 생성하는 경우를 나타낸다. 탐색구역을 생성하기 위해 먼저 7개의 단일 셀이 전체 탐색구역에서 임의로 선택되며, 각 셀에 상·하, 좌·우 방향으로 확장 가능한 4개의 셀들이 (Expandable cells) 설정된다. 만약 ③구역과 같이 확장 가능한 셀이 전체 탐색구역 범위를 벗어나거나 또는 ①구역과 ②구역같이 다른 구역에 포함되는 경우 그 셀을 확장 가능한 셀들로부터 제거하여, 그 방향으로 탐색구역이 확장되는 것을 차단한다.

초기 설정이 완료된 후, 처음 선택된 셀들로부터 임의의 셀을 선택하고, 연속적으로 확장 가능한 셀로부터 임의의 셀을 선택한다. Fig. 2는 ①구역 생성의 예로 탐색구역은 확장 가능한 셀이 선택된 방향으로 직사각형 모양을 유지하도록 확장되며, 확장 가능한 셀

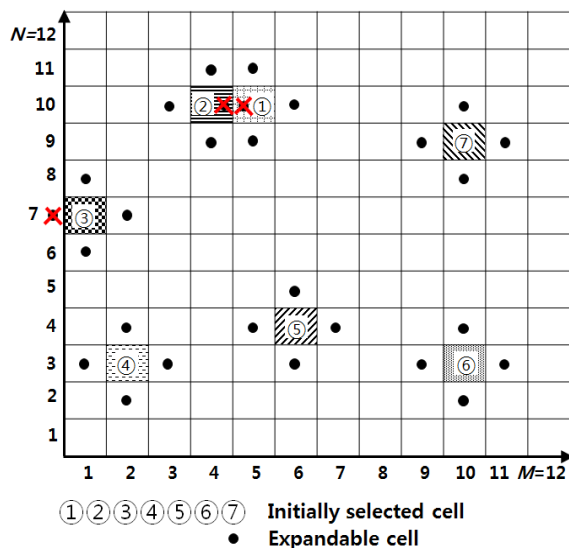


Fig. 1. Initial setup of search regions and expandable cells

은 확장된 탐색구역으로 재설정된다. 만약 재설정된 탐색구역의 확장 가능한 셀이 다른 탐색구역에 포함되거나 또는 전체 탐색구역 범위를 벗어난다면, 해당 셀을 확장 가능한 셀들에서 제거하여, 그 방향으로 탐

Table 2. Algorithm of generating rectangular regions

```

function GenDetArea()
for s = 1: |S| % Initialize
% Randomly select cells
Q(s,:) = [randint(M), randint(N)]
% Set horizontal and vertical ranges for
each search region(Left, Right, Down, Up)
R(s) = [Q(s,1)-1, Q(s,1), Q(s,2)-1, Q(s,2)]
% Save index of expandable cells
E(s,1,:)=[Q(s,1)-1,Q(s,2)] % Left
E(s,2,:)=[Q(s,1)+1,Q(s,2)] % Right
E(s,3,:)=[Q(s,1),Q(s,2)-1] % Down
E(s,4,:)=[Q(s,1),Q(s,2)+1] % Up
end for
while E ≠ empty
s = randint(|S|) %Randomly select a search region
j = randint(|E(s)|) %Randomly select an expandable cell
if E(s,j,1)<1 or E(s,j,1)>M or E(s,j,2)<1 or E(s,j,2)>N
E(s,j) = [] %Remove expandable cell
end if
for h = 1:|S|
if (R(h,1)<E(s,j,1)≤R(h,2)) and...
(R(h,3)<E(s,j,2)≤R(h,4))
E(s,j) = []
end if
end for
% Update index of expandable cells
% Update horizontal/vertical range of the search region
if E(s,j) ≠ empty
if j==1
E[s,1,1] = E[s,1,1] - 1
R(s,1) = R(s,1) - 1
elseif j==2
E[s,2,1] = E[s,2,1] + 1
R(s,2) = R(s,2) + 1
elseif j==3
E[s,3,2] = E[s,3,2] - 1
R(s,3) = R(s,3) - 1
elseif j==4
E[s,4,2] = E[s,4,2] + 1
R(s,4) = R(s,4) + 1
end if
end if
end while
return R
    
```

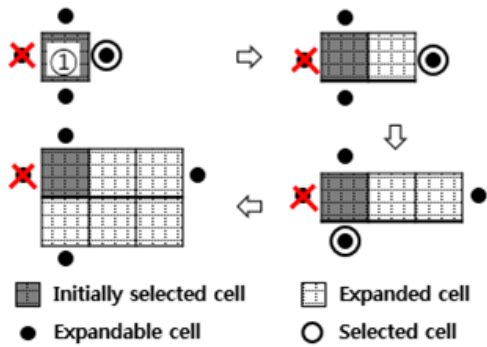


Fig. 2. Search region growth Algorithm for maintaining rectangular shape

색구역이 확장되는 것을 차단한다. 해당 절차는 각 구역에 설정된 확장 가능한 셀이 존재하지 않을 때까지 반복 수행되며, 세부 절차는 Table 2와 같다. 그러나 이러한 방법은 구역에 할당되지 않은 셀들(미할당구역)이 존재한 채로 종료될 수 있으며, 이러한 경우 생성된 구역의 범위를 조정하여 미할당구역을 제거해야 한다. 미할당구역을 제거하는 방법은 다음 절에서 자세히 소개한다.

3.2.2 탐색구역 조정

탐색구역 조정단계는, 탐색구역 생성 단계에서 구역에 할당되지 않은 셀들(미할당구역)을 인접해 있는 구역에 포함하여 제거하는 단계이다. Fig. 3은 미할당구역이 발생하는 한 예로, 미할당구역과 인접한 구역의 모양은 다양하게 생성될 수 있으나, 기본 발생 원리는 동일하므로, Fig. 3의 예를 이용하여, 탐색구역 조정 방법을 설명한다. 각 구역의 좌·우 범위의 값을 R_{q1}, R_{q2} ($q \in Q$)으로 정의하고 하·상 범위의 값을 각각 R_{q3}, R_{q4} 각로 각각 정의하자. 또한, ③구역의 좌측 좌표와 하측 좌표값을 ($R_{31} = R_{33} = 0$)으로 정의하자. ①구역의 예를 보면, 좌·우 범위의 값 R_{11}, R_{12} 는 전체 탐색구역의 가로 범위($R_{11} < R_{12}$)에 대응하며, 하·상 범위의 값은 전체 탐색구역의 세로 범위($R_{13} < R_{14}$)에 대응한다. 각 구역의 좌, 우, 하, 상 좌표를 $R_q = [R_{q1}, R_{q2}, R_{q3}, R_{q4}]$ 으로 정의하고, 유사하게 미할당구역(⑧구역) 범위의 좌표를 $X_8 = [X_{81}, X_{82}, X_{83}, X_{84}]$ 으로 정의하자. 미할당구역을 포함하여 확장하는 구역은 미할당구역에 인접한 4개의 구역(①,⑤,⑥,⑦)중 임의로 하나의 구역이 선택되며, 본 예에서는 ①구역이 확장구역으로 선택되었다고 가정하자. ①구역이 ⑧구역을 포함하기 위해서

는 ①구역의 하측 범위의 좌표 R_{13} 이 ⑧구역의 하측 범위의 좌표 X_{83} 까지 확장되어야 한다. 또한 ①구역과 ⑧구역의 우측 범위의 좌표가 동일하고 ($R_{12} = X_{82}$), ①구역의 좌측 범위의 좌표가 ⑧구역의 좌측 범위의 좌표보다 작기 때문에 ($R_{11} < X_{81}$), ①구역을 확장할 때 ⑧구역의 좌측으로 인접한 ⑤구역 또한 축소된다. 여기서 중요한 것은 ①구역이 확장될 때, ⑤구역이 직사각형 모양을 유지하며 축소되기 위해서는 ①구역과 ⑤구역의 좌측 범위의 좌표가 동일해야 ($R_{11} = R_{51}$) 한다는 것이다.

그러나 Fig. 3에서 보듯이 ①구역의 좌측 범위의 좌표가 ⑤구역의 좌측 범위의 좌표보다 크기 때문에 ($R_{51} < R_{11}$), 확장구역은 ①구역에 좌측으로 인접한 ②구역을 포함하여 재설정되며, 축소구역의 좌측 범위의 좌표와 다시 비교하게 된다. 재설정된 확장구역(①구역+

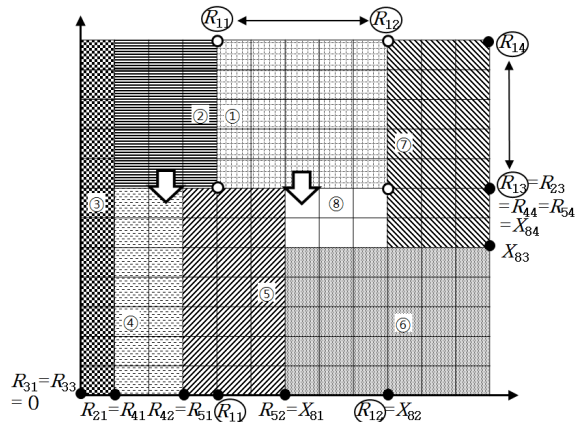


Fig. 3. Algorithm to remove unassigned cells

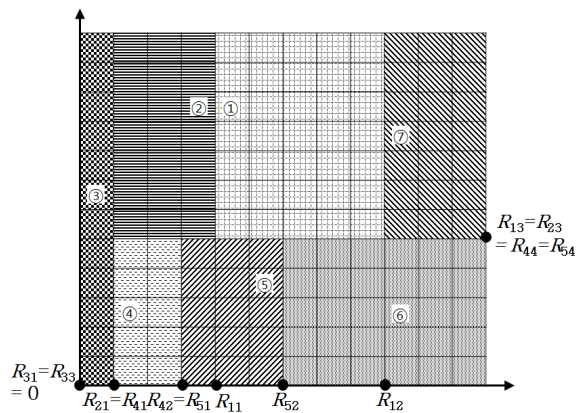


Fig. 4. Partitioned rectangular regions

Table 3. Algorithm of generating partitioned rectangular regions

```

function AdjDetArea(R)
X = [X(1),...,X(v),...,X(V)] % Set of unassigned regions
% Setup of ranges of unassigned regions
X(v) = [X(v,1), X(v,2), X(v,3), X(v,4)] (Left, Right, Down, Up)
while X ≠ empty
v = rand[|X|] %Randomly select an unassigned region
for h = 1:|S|
%Condition capable of being expanding regions
if R(h,3)=X(v,4) and R(h,1)≤X(v,1)...
and X(v,2)≤R(h,2) %Condition for ①
EQ = [EQ;R] %Save ① as an expanding region
ED = down %Direction of removing the unassigned region
%Expansion direction of expanding/shrinking regions
RD = left
elseif ...
end if
end for
%Randomly select an expanding region(j=①)
j = randint[|EQ|]
TEQ = EQ(j)
if RD == Left
for h = 1:|S|
if R(h,2) = X(v,1)
if R(h,3)≤X(v,3) and X(v,4)≤R(h,4)
RQ = R(h,:) %Set ⑤ as an shrinking region
end if
end if
end for
elseif RD==Right or RD==Down or RD==Up...
end if
TRQ = RQ %Save the shrinking region(⑤)
if ED==Down and RD==Left
while TEQ(1,1)≠TRQ(1,1)
if TRQ(1,1)<TEQ(1,1)
%Add ④ to the shrinking region
TEQ(1,1)←TEQ(1,1) %Update ranges
elseif TEQ(1,1)<TRQ(1,1)
%Add ② to the expanding region
TRQ(1,1)←TRQ(1,1)%Update ranges
end if
if TEQ(1,1) == TRQ(1,1)
%Update ranges of regions belonging to TEQ/TRQ
X(v) = [] %Remove X(v) from X
end if
end while
elseif (ED==Up, and RD==Left) or ...
end if
end while
return R
    
```

②구역)의 좌측 범위의 좌표는 ⑤구역의 좌측 범위의 좌표보다 작기 때문에 ($R_{21} < R_{51}$), 축소구역은 ⑤구역에 좌측으로 인접한 ④구역을 포함하여 재설정되며 확장구역의 좌측 범위의 좌표와 다시 비교하게 된다. 재설정된 축소구역(⑤구역+④구역)과 확장구역(①구역+②구역)의 좌측 범위의 좌표가 동일하므로 ($R_{21} = R_{41}$), 확장구역에 속한 ①구역과 ②구역의 하측 범위의 좌표를 각각 $R_{13} \rightarrow X_{83}, R_{23} \rightarrow X_{83}$ 으로 조정하고, 또한 축소구역에 속한 ⑤구역과 ④구역의 상측 범위의 좌표를 $R_{54} \rightarrow X_{83}, R_{44} \rightarrow X_{83}$ 으로 조정하면, Fig. 4와 같이 미할당구역이 제거된 탐색구역을 얻을 수 있다. 이런 방법은, 미할당구역이 존재하지 않을 때까지 수행되며, 세부 절차는 Table 3과 같다. 다음 절에서는 탐색구역 조정단계에서 생성된 구역을 각각의 플랫폼에 할당하는 방법에 대해 설명한다.

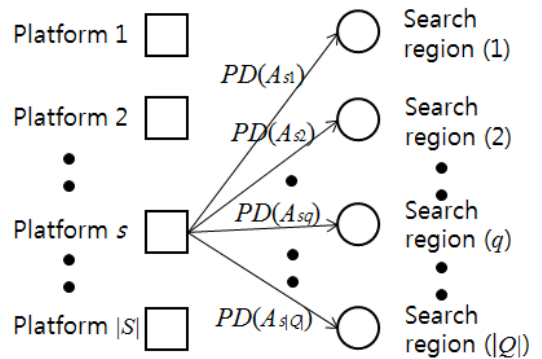


Fig. 5. One to one assignment problem for platform-search region pairs

3.2.3 탐색구역 배치

탐색구역 배치단계는 탐색구역 생성 및 조정단계에서 식 (2)의 제약조건을 만족하도록 임의로 생성된 분할구역에 각 플랫폼을 할당하는 단계이다. 분할구역 ($A_q, q \in Q$)이 주어진 경우 식 (2)는 각 플랫폼을 분할구역에 할당하는 일대일 배치(one-to-one assignment) 문제로 변환된다. Fig. 5는 플랫폼과 분할구역의 일대일 배치 문제의 개념을 나타낸다. 플랫폼과 분할구역을 연결하는 연결선(arc)은 각 플랫폼 s 가 분할구역($A_q, q \in Q$)에 배치되는 경우에 대한 탐지확률($PD(A_{sq}), \forall q \in Q, \forall s \in S$)을 나타내며, 식 (1)을 이용하여 계산할 수 있다. Fig. 5의 일대일 배치(one-to-one assignment) 개념의 수학적 모델링은 식 (3)과 같다.

$$\begin{aligned} \max \sum_{s \in S} \sum_{q \in Q} PD(A_{sq}) f_{sq} & \quad (3) \\ \text{s.t. } \sum_{s \in S} f_{sq} = 1 & \quad (a) \\ \sum_{q \in Q} f_{sq} = 1 & \quad (b) \\ f_{sq} \geq 0, \forall s, q, |S| = |Q| & \quad (c) \end{aligned}$$

식 (3)에서 f_{sq} 는 플랫폼 s 가 분할구역 A_q 에 할당되면 $f_{sq} = 1$, 그렇지 않다면 $f_{sq} = 0$ 의 값을 가지는 변수이며, 옥션(Auction), 헝가리안(Hungarian) 알고리즘 등의 방법을 적용하여 식 (3)의 목적함수를 최대로 하는 f_{sq}^* 구할 수 있다. 3단계 절차로 부터 산출된 해는, 식 (2)의 최적의 해는 아니지만, 반복 수행을 통해 매번 산출되는 탐지확률을 이전에 산출된 탐지확률과 비교하여, 점차 향상된 분할구역을 얻을 수 있다. 다음 장에서는, 식 (2)의 제약식을 완화한 후, Lagrange 방법을 이용하여 해석적 해를 손쉽게 얻을 수 있는 특수한 경우를 소개한다.

3.3 탐색구역 할당문제의 해석적 해(analytic solution)가 산출 가능한 경우

식 (2)에서 전체 탐색구역을 구성하는 모든 셀에 탐지표적 존재 확률값이 $PP(g) = \frac{1}{MN} (\forall g \in G)$ 으로 균일하게 분포되어 있고, 모든 플랫폼의 탐지성능(탐색폭(w_s), 탐색속력(v_s), 탐색시간(τ_s))이 동일한 경우를 가정하자. 또한 식 (2)에서 각 플랫폼의 탐색 구역은 분리된 구역이어야 한다는 제약식 2(a)와, 직사각형 모양이 되어야 한다는 제약식 2(b), 그리고 탐색구역에 속한 셀들의 수가 1보다 큰 정수가 되어야 한다는 제약식 2(d)를 완화한다면, 식 (2)는 식 (4)가 된다.

$$\begin{aligned} \max \sum_{s \in S} \frac{|A_s|}{MN} \left(1 - \exp\left(-\frac{C}{|A_s|}\right) \right) & \quad (4) \\ \text{s.t. } \sum_{s \in S} |A_s| = MN & \end{aligned}$$

식 (4)의 C 는 식 (2)에서 플랫폼의 탐색폭(w_s), 탐색속력(v_s), 탐색시간(τ_s), 단일 셀의 넓이 b 와 관련된 상수이다. 이러한 문제를 풀기 위해 식 (4)를 Lagrangian 형태로 변형하면, 식 (5)와 같이 나타낼 수 있다.

$$\begin{aligned} L(|A_1|, \dots, |A_{|S|}|, \lambda) & \quad (5) \\ = \sum_{s \in S} \frac{|A_s|}{MN} \left(1 - \exp\left(-\frac{C}{|A_s|}\right) \right) + \lambda \left(MN - \sum_{s \in S} |A_s| \right) \end{aligned}$$

식 (4)의 목적함수를 분할 탐색구역을 구성하는 셀들의 수 $|A_s|$ 의 함수 $f(|A_s|)$ 로 정의할 때, 식 (5)를 $|A_s|$ 에 관하여 미분한 결과는 식 (6)과 같고 λ 에 관하여 미분한 결과는 식 (7)과 같다.

$$\begin{aligned} \frac{\partial L}{\partial |A_s|} &= \frac{\partial f(|A_s|)}{\partial |A_s|} - \lambda & (6) \\ &= \frac{1}{MN} \left(1 - \exp\left(-\frac{C}{|A_s|}\right) \right) \\ &\quad - \frac{1}{MN} \left(\exp\left(-\frac{C}{|A_s|}\right) \right) \times \left(-\frac{C}{|A_s|^2} \right) - \lambda \\ &= 0 \end{aligned}$$

$$\frac{\partial L}{\partial \lambda} = MN - \sum_{s \in S} |A_s| = 0 \quad (7)$$

식 (6)을 만족하는 해석적 해를 구하는 것은 불가능하나 $\frac{\partial f(|A_s|)}{\partial |A_s|}$ 은 $|A_s| > 0$ 에 대해 단조 감소하는 함수이므로 식 (6)을 만족하는 해가 존재한다면 단일해를 가진다. 또한 식 (6)은 $|A_s|$ 이외의 모든 구역에 대해 디커플링(Decoupling)되어 있으므로 분할된 구역은 모두 동일한 해를 가져야 하며, 식 (7)에 대입하면 $|A_s|^* = \frac{MN}{|S|} (s \in S)$, 값을 가진다. 또한, $f(|A_s|^*)$ 은 concave 함수이므로, Lagrange의 이론에 따라 $|A_s|^*$ 은 최적해가 된다.

식 (4)의 목적함수를 식 (2)에 적용하여 얻을 수 있는 최적해를 $|A_s|^\star$ 로 정의하자. 식 (4)의 $|A_s|^\star$ 는 식 (2)의 제약식 (a), (b), (d)를 완화하여 얻은 최적해이므로, 식 (4)의 탐지확률은 식 (2)의 탐지확률보다 항상 크거나 같으며($f(|A_s|^\star) \leq f(|A_s|)$), 따라서 식 (2)의 탐지확률에 대한 upper-bound가 된다. 만약 $|A_s|^\star$ 가 정수이고, $|A_s|^\star$ 만큼의 셀 수를 포함한 직사각형 형태로, 전체 탐색구역을 분할할 수 있다면, 식 (4)의 최적해는 식 (2)의 최적해와 동일한 값을 가지며 ($|A_s|^\star = |A_s|$) 목적함수 또한 동일한 탐지확률을 ($f(|A_s|^\star) = f(|A_s|)$) 가진다. 이와 같이 식 (2)의 제약식을 완화한 후, Lagrange 방법을 이용하여 유도된 해석적 해는 탐색구역할당 문제의 upper-bound를 제공한다. 다음 절에서 시뮬레이션을 통하여 제안한 알고리즘의 모의 결과와 upper-bound와의 비교를 통해 알고리즘의 성능을 검토한다.

3.4 모의 결과

단일 셀의 넓이가 $b = 1$ 이고, 탐지표적 존재 확률이 균일한 40×40 탐색구역에서, 동일한 탐지 성능($w_s = 5, v_s = 20, \tau_s = 5$)을 가진 2개 플랫폼이 배치되는 경우를 가정하자. 3.3절에서 설명한 것처럼, 식 (2)의 제약식 (a), (b), (d)를 완화하여 식 (4)를 통해 얻을 수 있는 해석적 해는 $|A_s|^* = \frac{MN}{|S|} (\forall s \in S)$ 으로 각각의 플랫폼을 800개의 셀을 포함한 탐색구역에 할당하는 것이며, 또한 그 해를 적용시 산출되는 탐지확률은 식 (2)의 upper-bound를 제공한다. 800개의 셀은 전체 탐색 구역을 가로 또는 세로로 이등분 한 직사각형 구역으로 분할 할 수 있으므로, 식 (2)의 최대 탐지확률과 식 (4)의 최대 탐지확률은 동일한 값을 가져야 한다. Fig. 6는 본 논문에서 제안한 알고리즘의 탐지확률과 식 (4)의 해석적 해를 적용시 산출되는 탐지확률을 비교한 것으로 탐색구역 생성, 조정, 배치 절차 단계를 25번 반복 수행한 후에 upper-bound에 도달한다.

Fig. 7은 동일한 탐지 성능($w_s = 5, v_s = 20, \tau_s = 5$)을 가진 3개의 플랫폼을 탐색구역에 할당하는 경우로, 본 논문에서 제안한 알고리즘의 탐지확률과 식 (4)의 해석적 해를 적용시 산출되는 탐지확률을 비교한 값을 보인다. 식 (4)의 해석적 해는 각각의 플랫폼을 $1600/3$ 개의 셀을 포함한 탐색구역에 할당하는 것이나, 그 값은 정수가 아니므로, 식 (2)의 제약조건을 만족하지 않으며, 따라서 Fig. 7과 같이 식 (4)의 해석적

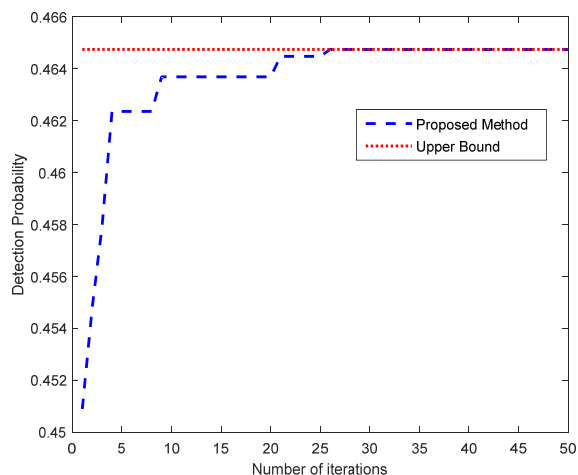


Fig. 6. Upper-bound detection probability and detection probability obtained by the proposed algorithm ($|S| = 2, C = 500, 40 \times 40$ cells)

해보다 작은 값으로 수렴하는 것을 알 수 있다.

Fig. 8은 동일한 탐지 성능($w_s = 5, v_s = 20, \tau_s = 5$)을 가진 각각 4, 6, 8, 10개의 플랫폼을 4, 6, 8, 10개의 분할 탐색구역에 할당하는 경우 탐지확률의 변화를 나타내며, Fig. 9는 10개의 플랫폼에 대해 탐색구역 생성, 조정, 배치 방법의 100번 반복수행 후 얻어지는 분할 탐색구역을 나타낸다.

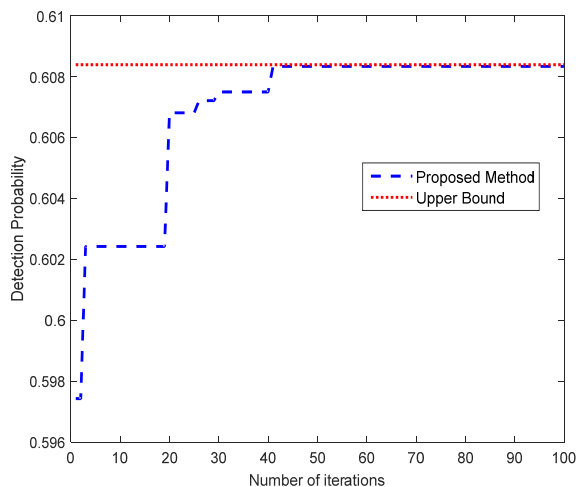


Fig. 7. Upper-bound detection probability and detection probability obtained by the proposed algorithm ($|S| = 3, C = 500, 40 \times 40$ cells)

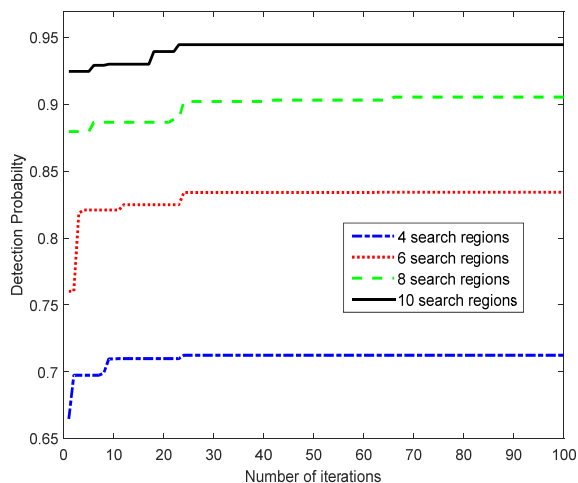


Fig. 8. Detection probability obtained by the proposed algorithm ($|S| = 4, 6, 8, 10, C = 500, 40 \times 40$ cells)

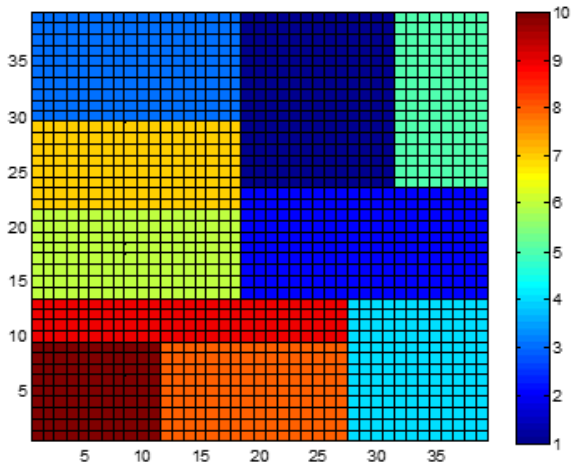


Fig. 9. Partitioned search area($|S| = 10, C = 500, 40 \times 40$ cells)

Table 4. Performance of the proposed algorithm

구 분		2구역	4구역	6구역	8구역	10구역
탐 지 율	Upper Bound	0.465	0.713	0.846	0.917	0.956
	평균	0.465	0.711	0.840	0.907	0.943
	표준 편차	0	0.001	0.002	0.003	0.003
	평균/ Upper Bound	1	0.997	0.992	0.989	0.986
연산 시간 (sec)	평균	1.535	1.813	2.049	2.257	2.465
	표준 편차	0.039	0.035	0.032	0.055	0.031

Table 4는 분할구역 수의 변화에 따른 알고리즘의 성능평가 지표로서, 탐색구역 생성, 조정, 배치 단계를 100번 반복수행할 때 탐지율과 연산시간의 평균과 표준편차를 나타낸다. Table 4에서 보듯이, 분할구역의 수가 증가함에 따라 평균 탐지율과 식 (4)의 해석적 해의 탐지확률(upper-bound)의 비는 1에서 0.986으로 감소하고, 연산시간은 1.535초에서 2.465초로 증가하는 것을 알 수 있다. 주의할 점은 2, 4, 8, 10개의 탐색구역은 정수개의 셀을 포함한 직사각형 모양으로 분할이 가능하므로 upper-bound는 식 (2)의 최적 탐지확률

과 동일한 값을 가지나, 6개의 탐색구역에 대한 식 (4)의 해석적 해는 각각의 플랫폼에 1600/6개의 셀을 포함한 탐색구역을 할당하는 것으로, 그 값은 정수가 아니기 때문에, 식 (2)의 최적 탐지확률보다 큰 값을 가진다는 것이다.

4. 결 론

본 논문에서 탐색구역을 직사각형 모양으로 제한하고, 각각의 구역이 중첩되지 않도록 전체 탐색구역을 분할한 후, 각각의 플랫폼을 분할된 구역에 할당하는 문제를 논의하였으며, 이러한 NP-hard 문제를 해결하기 위해, 탐색구역 생성, 조정, 배치와 같은 3단계의 절차를 갖는 알고리즘을 제안하였다. 또한, 탐색구역 할당 문제의 제약식을 완화하여 해석적 해를 유도하였으며, 그 해와 비교를 통해 본 논문에서 제안한 구역분할 알고리즘의 성능을 분석하였다. 향후 본 논문에서 제안한 구역분할 알고리즘의 성능 향상을 위해 임의적인 방법이 아닌, 이전에 생성된 분할 탐색구역 또는 탐지표적이 존재할 확률 정보를 활용하여 효율적으로 구역을 생성하고, 조정하는 방법을 연구할 계획이다. 또한 탐색임무에 대한 교대가 필요한 경우로 문제를 확장하여 플랫폼의 배치, 운용, 복귀와 관련된 해법을 연구할 계획이다.

References

- [1] J. Eagle, "The Optimal Search for a Moving Target When the Search Path is Constrained," Operations Research, Vol. 32, pp. 1107-1115, 1984.
- [2] G. H. Martins, "A New Branch-and-Bound Procedure for Computing Optimal Search Paths", Master's Thesis, Naval Postgraduate School, 1993.
- [3] H. Lau, "Optimal Search in Structured Environments," PhD Thesis, University of Technology, Sydney, 2007.
- [4] A. G. Santos, "Using Multiple Searchers to Locate a Randomly Moving Target," Master's Thesis, Naval Postgraduate School, 1993.
- [5] W. An, M. Mishra, C. Park, and K. R. Pattipati, "An Integrated Asset Allocation and Path Planning

- Method to Search for Targets in a Dynamic Environment,” 15th International Command and Control Research and Technology Symposium, Santa Monica, CA, June 2010.
- [6] W. An, D. F. Martinez Ayala, D. Sidoti, M. Mishra, X. Han, K. R. Pattipati, E. D. Regnier, D. L. Kleinman and J. A. Hansen, “Dynamic Asset Allocation Approaches for Counter-Piracy Operations,” 15th International Conference on Information Fusion, Singapore, July 2012.
- [7] S. Muthukrishnan, V. Poosala, and T. Suel, “On Rectangular Partitionings in Two Dimensions: Algorithms, Complexity and Applications,” International Conference on Database Theory, Springer Berlin Heidelberg, 1999.
- [8] B. O. Koopman, “Search and Screening: General Principles with Historical Applications,” Pergamon Press, New York, 1980.
- [9] W. L. Winston, “Operations Research,” Fourth Edition, Brooks/Cole.