

# 선박 충돌 방지를 위한 분산 확률 탐색 알고리즘의 분석

김동균\*\*

\* 고베대학교 해사과학대학원

## Analysis of a Distributed Stochastic Search Algorithm for Ship Collision Avoidance

Kim Donggyun\*\*

\* Graduate School of Maritime Sciences, Kobe University, Japan

**요약** : 다수의 선박이 조우하였을 경우, 충돌 피항을 위해 상대 선박의 의도를 파악하는 것은 매우 중요한 문제이다. 또한 다수의 선박의 의도를 동시에 고려하여 충돌 피항 계획을 세우는 것은 항해사에게 큰 부담이 될 수 있다. 이를 위해 분산 알고리즘이 제안되었다. 분산 알고리즘은 각각의 선박이 다수의 상대 선박과 정보 교환을 통해 안전한 코스를 탐색할 수 있도록 한다. 본 논문에서는 분산 알고리즘의 하나인 분산 확률 탐색 알고리즘을 선박 충돌 피항에 적용하였다. 분산 확률 탐색 알고리즘에서 선박은 비용 감소가 가장 큰 코스와 기존의 코스를 확률과 제한 조건에 따라 선택한다. 분산 확률 탐색 알고리즘은 확률과 제한 조건에 따라 다섯 가지 종류로 나뉜다. 본 논문에서는 다섯 가지 종류의 분산 확률 탐색 알고리즘을 선박 충돌 피항을 위해 적용하였으며 선박 충돌 피항에 미치는 영향을 분석하였다. 또한 어떠한 분산 확률 탐색 알고리즘이 충돌 피항에 적합한지를 실험하였다. 실험 결과 다섯 가지 버전의 분산 확률 탐색 알고리즘에서 A와 B방식이 효과적으로 선박 충돌 피항을 수행하였다. 본 알고리즘은 분산 시스템 환경에서 선박 충돌 방지를 위해 적용 가능할 거라 기대된다.

**핵심어** : 분산 확률 탐색 알고리즘, 충돌 피항, 선박의 의도, 분산 시스템, 다수의 선박이 조우하는 경우

**Abstract** : It is very important to understand the intention of a target ship to prevent collisions in multiple-ship situations. However, considering the intentions of a large number of ships at the same time is a great burden for the officer who must establish a collision avoidance plan. With a distributed algorithm, a ship can exchange information with a large number of target ships and search for a safe course. In this paper, I have applied a Distributed Stochastic Search Algorithm (DSSA), a distributed algorithm, for ship collision avoidance. A ship chooses the course that offers the greatest cost reduction or keeps its current course according to probability and constraints. DSSA is divided into five types according to the probability and constraints mentioned. In this paper, the five types of DSSA are applied for ship collision avoidance, and the effects on ship collision avoidance are analyzed. In addition, I have investigated which DSSA type is most suitable for collision avoidance. The experimental results show that the DSSA-A and B schemes offered effective ship collision avoidance. This algorithm is expected to be applicable for ship collision avoidance in a distributed system.

**Key Words** : Distributed stochastic search algorithm, Collision avoidance, Intention of ship, Distributed system, Multiple-ship situations

### 1. 서론

대부분의 선박 충돌 방지를 위한 연구에서는 1대1 또는 1대 다수 선박이 조우할 경우를 가정하여 피항 방법을 제안하였다(COLREGS, 1972; Fujii and Tanaka, 1971; Goodwin, 1975).

실험에 사용된 타선은 본선의 움직임에 상관없이 초기에 설정된 값, 즉 출발 위치, 목적지, 속력, 코스에 따라 항행하였다. 그러나 실제 항행 시 본선의 움직임에 타선이 반응하며 타선의 결정에 따라 본선도 충돌 피항 계획을 변경한다. 다수의 선박이 조우할 경우, 상대 선박의 피항 의도를 파악하는 것은 충돌 방지를 위해 매우 중요한 일이다.

† kimdg.421@gmail.com

다수의 선박이 조우하는 경우 충돌 회피를 위한 분산 알고리즘이 제안되었다(Kim et al., 2014; 2015; 2017). 분산 알고리즘에서 각 선박은 주위의 선박들과 정보 교환을 통해 안전한 코스를 찾는다. 정보 교환을 반복하면서 점차 충돌 위험도가 가장 낮은 코스를 찾아 다음 위치로 이동한 후 다시 충돌 위험도가 가장 낮은 코스를 찾는 과정을 반복한다. 선박이 목적지에 안전하게 도착하면 프로세스는 종료된다.

본 논문에서는 다섯 가지 버전의 분산 확률 탐색 알고리즘이 선박 충돌 피항에 적합함을 확인하는 실험을 하였다. 또한 각 각의 분산 확률 탐색 알고리즘이 충돌 피항에 어떠한 영향을 미치는 지도 분석하였다. 실험은 5척, 12척의 선박이 항행할 경우를 가정하여 진행하였다. 실험은 MATLAB으로 실시되었다.

## 2. 관련 연구

선박 간 충돌 방지를 위한 탈 집중형 방식과 분산형 방식을 적용한 관련 연구를 설명한다. Hu et al.(2008)은 탈 집중형 방식인 *Collision Avoidance Negotiation Framework(CANFO)*라는 방법을 제안하였다. 두 척의 선박이 조우하였을 때 협상을 통해 충돌 회피를 실시하였다. CANFO는 실험을 통해 효과적인 충돌 회피 방법이라는 것이 증명되었다. 그러나 1대1의 상황에만 국한된다는 한계점을 지녔으며 다수의 선박이 조우할 경우에는 증명되지 않았다.

다른 방법으로는 선박 간 정보 교환의 유무 즉, 협조적인 선박 두 척과 비협조적인 선박 한 척이 조우하였을 경우를 가정하여 최적의 항로를 찾는 방법이 제안되었다(Hornauer, 2013; Hornauer et al., 2015). 협조적인 선박 간에는 정보 교환을 통해 서로의 예상 위치를 확인하며 충돌 피항 계획을 세운다. 비협조적인 선박의 위치를 예상하기 위해 과거의 AIS의 정보를 이용하여 Bayesian 모델을 기반으로 선박의 예상 위치를 계산하였다. 실험 결과 항로의 예측이 가능하다고 기술하였으나 실험에 대한 자세한 내용은 설명되지 않았다.

항해 시 상대 선박의 의도를 명확히 알고 미리 서로의 의도를 결정할 수 있다면 충돌 위험을 줄일 수 있다. 이러한 정보 교환을 통한 충돌 방지를 위해 분산 알고리즘, 즉 분산 지역 탐색 알고리즘(Distributed Local Search Algorithm, *DLSA*) (Kim et al., 2014), 분산 타부 탐색 알고리즘(Distributed Tabu Search Algorithm, *DTSA*)(Kim et al., 2015), 분산 확률 탐색 알고리즘(Distributed Stochastic Search Algorithm, *DSSA*)(Kim et al., 2017)이 제안되었다. *DLSA*에서는 다수의 선박이 조우하였을 경우 일정 거리 이내의 상대 선박들과 정보 교환을 통한 충돌 피항 계획을 세운다. 상대 선박들과 정보 교환을 통한 예상 위치를 교환하고 이를 기반으로 현재 상태에서 가장 크

게 충돌 위험도를 낮출 수 있는 코스를 선택한다. 비용 감소가 가장 큰 선박이 다음 침로를 선택할 권리를 갖는다. 즉, 각 단계에서 가장 크게 충돌 위험도를 낮추는 방법을 선택한다. 이렇게 바뀐 코스를 기반으로 갱신된 예상 위치는 다시 상대 선박과 정보 교환을 통해 새로운 코스를 탐색한다. 이와 같은 과정을 충돌 위험이 없어질 때까지 반복된다. 만약 충돌 위험이 사라지거나 계산 시간이 초과될 경우, 마지막에 선택된 코스를 따라 다음 위치로 이동한다. 목적지에 도착할 때까지 위의 과정을 반복한다.

*DTSA*에서는 *DLSA*에서의 문제점, 즉 *Quasi-local minimum (QLM)*을 해결하기 위해 타부 탐색 알고리즘이 적용되었다(Glover, 1989). 충돌 피항에서 *QLM*이란 선박 간 충돌이 예상되지만 현재 코스의 비용보다 다른 코스의 비용이 높을 경우 선박은 어떠한 선택도 할 수 없는 상황을 의미한다. 이 경우 선박은 계산 시간이 경과하기만을 기다리며 다음 위치로 이동하여 새로운 코스를 탐색해야 한다. 이 때 충돌 위험도는 더욱 높아진다. *QLM*이 발생할 경우, 한 번 탐색된 코스는 타부 리스트에 저장된다. 이 타부 리스트에 저장된 코스는 일정 기간 동안 다시 선택되지 않도록 한다. 즉, 강제로 다른 값을 선택하도록 하여 *QLM*에서 탈출하도록 하였다.

그러나 *DLSA*와 *DTSA*에서는 무한 루프를 방지하기 위해 모든 선박이 동시에 코스를 변경하지 못하게 하였다. 즉, 이웃 관계에 있는 선박(본선을 포함) 중 한 척의 선박만이 코스를 변경할 수 있다. 이와 같은 방법은 점진적으로 모든 선박의 충돌 위험도를 낮춰줄 수 있다.

그러나 이로 인해 선박 간 많은 메시지 교환이 발생할 수 있다. 이는 충돌 위험이 예상되는 긴박한 상황에서 문제를 일으킬 수 있다. 따라서 선박 간 메시지 교환 횟수를 감소시키기 위해 *DSSA*가 제안되었다. *DSSA*는 확률을 이용하여 본선의 다음 코스를 바꾼다. 이는 다수의 선박이 동시에 침로를 선택할 수 있게 하여 선박 간 메시지 교환 횟수를 크게 낮추었으며 알고리즘을 단순화시켰다. *DSSA*는 분산 시스템 환경에서 선박 충돌 방지에 적용 가능성을 보였다. 그러나 *DSSA*를 적용한 실험에서는 다섯 가지 버전 중 한 가지만의 방식(*DSSA-A*)만을 사용하여 실험을 진행하였다.

따라서 본 논문에서는 다섯 가지 버전의 *DSSA*의 성능을 비교하여 선박 충돌 방지에 적용 가능여부를 확인한다.

## 3. 분산 확률 탐색 알고리즘

*DSSA*는 분산 확률 탐색(*DSA*)을 활용하여 선박 충돌 방지에 적용되었다(Zhang et al., 2005). *DSA*는 동기적인 방식으로 일정 거리 이내의 객체와 메시지를 주고받는다. Fig. 1은 각

각의 객체가 해를 찾기 위한 DSA의 작동 방식에 대해 설명한다. 초기에 객체는 해를 무작위로 선택한다. 각각의 객체는 현재 상태의 정보, 즉 자신이 가지고 있는 해가 이전 상태와 다를 경우, 이웃 객체에 전송한다. 이웃 객체로부터 메시지를 수신하였을 경우, 수신된 메시지의 정보와 자신이 가지고 있는 정보를 조합하여 새로운 해를 확률  $p$ 에 따라 선택한다.

```

Randomly select a value
While (no termination condition is met) do
If (new value is assigned) then
Send the value to neighbors
End if
Receive neighbor's values,
Choose and assign the next value as Table 1
End while
    
```

Fig. 1. Procedure of DSA.

DSA에서 가장 중요한 것은 객체가 다음 해를 어떻게 선택하는가이다. 즉 현재 상태를 향상시킬 수 있는 해가 존재할 경우, 일정한 확률에 따라 새로운 해를 선택한다. 그렇지 않을 경우, 이전에 선택된 해를 유지한다. Table 1은 다섯 가지 버전의 DSA를 향상도와 충돌의 유무에 따라 분류한 것이다. 향상도는 비용 감소의 정도를 의미한다. 충돌은 새로운 해를 선택할 경우, 제약 조건이 발생하는 것을 의미한다.  $p$ 는 확률,  $v$ 는 변수를 *Improvement*는 향상도, *Conflict*는 제약 조건 위반을 의미한다. 확률  $p$ 는 각 객체의 활동적인 성향을 조절한다. 즉, 확률  $p$ 가 클수록 새로운 해를 선택하려는 경향이 크고, 확률  $p$ 가 작을수록 과거의 해를 유지하려는 경향이 크다.

Table 1. Five kinds of DSAs

Algorithm	Improvement > 0	Conflict, Improvement > 0	No conflict, Improvement = 0
DSA-A	$v$ with $p$	-	-
DSA-B	$v$ with $p$	$v$ with $p$	-
DSA-C	$v$ with $p$	$v$ with $p$	$v$ with $p$
DSA-D	$v$	$v$ with $p$	-
DSA-E	$v$	$v$ with $p$	$v$ with $p$

향상도와 충돌에 따라 객체가 해를 선택하는 방식에 대해 설명한다.

DSA-A의 경우, 객체는 향상도가 0보다 클 경우에만 해를 선택할 수 있다. 확률  $p$ 로 변수  $v$ 를 선택하거나 확률  $1-p$ 로 기존의 변수를 유지한다.

DSA-B는 DSA-A와 같은 방식으로 작동되나 제약 조건을 만족하지 못한 경우에도 확률  $p$ 로 새로운 해를 선택할 수 있다. 이 방식은 DSA-A의 작동 방식보다 좋은 결과가 기대된다. 비록 현재에는 제약 조건을 만족하지 못하지만, 다음 과정에서 제약 조건을 만족할 수 있기 때문이다. 따라서 전체적으로 객체들은 DSA-A의 방식보다 DSA-B의 방식이 적용될 경우, 더 큰 확률로 각자의 해를 바꿀 가능성이 있다.

DSA-C의 경우, DSA-B보다 더 공격적인 방식이다. 객체는 현재 상태에 만족하고 더 나은 향상도를 가지는 해가 없어도 확률  $p$ 로 새로운 해를 선택한다. 이는 비록 현재 상태에 비해 더 나은 상황을 찾지는 못해도 새로운 상태를 탐색할 수 있도록 한다.

DSA-D와 E는 각각 DSA-B와 C의 확장된 개념으로 더욱 공격적으로 변수를 선택한다. 만약 향상도가 0보다 크다면 결정론적인 방법으로 변수  $v$ 를 선택한다. 이는 탐욕적인 탐색 방법으로 당장 현재 상황을 개선시킬 수 있다면 즉각 해당 해를 선택한다.

## 4. 실험 방법

다섯 가지 종류의 분산 탐색 알고리즘의 성능을 비교, 분석하여 선박 충돌 피항에 적용가능한지를 알기 위해 계산시간과 확률에 따라 비용, 항행거리, 메시지 교환 횟수, 충돌 횟수를 계산하였다.

### 4.1 비용계산

비용은 모든 선박이 출발 위치에서 목적지까지 도착하는 과정에서 코스를 선택할 때 발생한 모든 비용, 즉, 타선에 대한 충돌 위험도와 코스와 목적지의 상대방위를 더한 값이다. 비용 함수는 식(1), (2)와 같이 정의된다(Kim et al., 2017).

$$\begin{aligned}
 & COST_{self}(crs) \\
 & \equiv \alpha \sum_{j \in TargetShips} CR_{self}(crs, j) + \frac{|\theta_{destination} - \theta_{self}(crs)|}{180^\circ} \quad (1)
 \end{aligned}$$

$$CR_{self}(crs, j) \equiv \begin{cases} \frac{Time\ Window}{TCPA_{self}(crs, j)}, & self\ conflict\ with\ j \\ 0, & otherwise \end{cases} \quad (2)$$

$\alpha$ 는 CR의 가중치이다. CR은 상대 선박과의 충돌 위험도이다.  $self$ 와  $j$ 는 각각 본선과 상대 선박을 의미한다.  $crs$ 는 후보 코스,  $\theta_{destination}$ 는 목적지를 향한 방위를 의미한다. Time window는 선박의 예상 위치를 계산하기 위한 시간이다. 비용 함수는 타선박과의 충돌 위험도(CR)와 목적지를 향한 상대 방위의 합으로 계산된다. 충돌 위험도를 계산하기 위해 사용된 TCPA는 DCPA가 본선의 안전영역의 반지름( $r$ ) 보다 작을 경우에만, 즉 충돌이 발생할 경우만 계산된다. 충돌 위험이 발생하지 않을 경우, 0을 반환한다. 목적지를 향한 상대방위는 0과 1사이의 값을 가지며 모든 위치에서 비용이 발생되며 충돌 위험도의 비용에 더해진다.

#### 4.2 항행거리

항행거리는 모든 선박이 출발 위치에서 목적지까지 안전하게 도착하였을 경우, 모든 선박의 항행 거리를 합하여 표시하였다.

$$distance = \sum_{s \in S} \sum_{i \in I} |\vec{v}_{s,i} \cdot t_i| \quad (3)$$

$\forall s \in S = 1, \dots, n$ 이며  $S$ 는 선박의 집합이다.  $\forall i \in I = 1, \dots, m$ 이며  $I$ 는 변침 횟수이다.  $\vec{v}_{s,i}$ 는 선박 $s$ 의  $i$ 번째에 해당되는 속도,  $t_i$ 는 time step간의 시간 간격을 의미한다. time step은  $t$ 시간에서의 선박의 위치이다.

#### 4.3 메시지 교환 횟수

각 선박은 충돌 위험이 발생할 경우, 상대 선박과 메시지를 교환하며 안전한 코스를 탐색한다. 목적지에 도착할 때까지 이와 같은 과정을 반복한다. 이 과정에서 발생된 모든 메시지의 수를 합하여 표시하였다.

#### 4.4 충돌 횟수

선박 간 충돌이 발생하였을 경우, 모든 충돌 횟수를 더하여 표시하였다. 충돌 판단은 식(4)와 같이 정의된다.

$$conflict \equiv \begin{cases} 1, & \text{if } |position_i^s - position_i^t| < sd_r^s \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$s$ 는 본선,  $t$ 는 상대 선박,  $position_i^s$ 는 선박 $s$ 의  $i$ 번째에 해당하는 위치를 의미한다.  $sd_r^s$ 는 선박  $s$ 의 안전영역의 반지름의 길이( $r$ )를 의미하며 고정된 값을 가진다. 상대 선박이 본선의 안전영역 내에 존재할 경우, 충돌로 간주한다.

### 5. 분산 확률 탐색 알고리즘의 실험

본 논문에서는 DSSA를 활용하여 선박 충돌 피항 실험을 실시하였다(Kim et al., 2017). 실험은 각각 5척, 12척의 선박이 항행하는 경우를 가정하였으며 MATLAB으로 진행되었다. 다섯 가지 버전의 DSSA의 성능을 비교하기 위해 계산 시간과 확률의 변화에 따른 비용, 항행 거리, 메시지 교환 횟수, 충돌 횟수를 조사하였다.

#### 5.1 5척의 선박이 조우할 경우

Fig. 2는 5척의 선박이 조우하였을 경우이다. 네 척의 선박은 모서리에서 출발하여 반대편 모서리를 목적지로 항행한다. 왼편 중간에 위치한 선박은 오른쪽으로 항행하면서 중앙을 통과하도록 설정하여 모든 선박이 중앙에서 조우하도록 설정하였다. Table 2는 각 선박의 초기 설정 값, 즉 출발 위치, 목적지, 침로, 속력을 나타낸다.

Table 2. Values for parameters of five ships

Parameter	Value
Origin	0, 0/0, 10/10, 0/10, 10
Destination	10, 10/10, 0/0, 10/0, 0
Course	090, 090, 270, 270, 180
Speed	12 knots (constant)

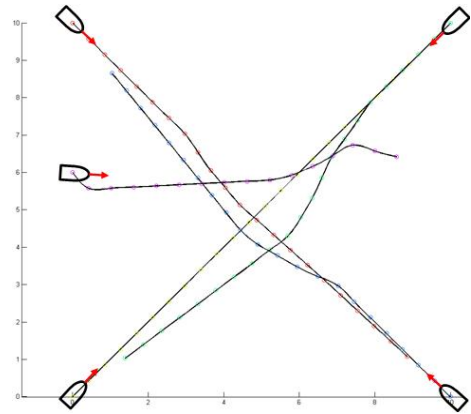


Fig. 2. When 5 ships are encountered.

#### 5.1.1 계산 시간에 따른 분석

##### (1) 항행 거리

Fig. 3은 계산 시간에 따른 선박의 항행 거리와 메시지 교환 횟수를 나타낸다. 계산 시간은 선박이 다음 위치를 결정하기 위한 주어진 시간을 의미한다. 이 시간을 초과할 경우

선박은 마지막에 선택된 코스를 선택한 후 다음 위치로 이동한다. 막대그래프는 항행 거리를 의미하며, 라인은 선박 간 메시지 교환 횟수를 의미한다.

DSSA-A, B, D의 경우(막대그래프), 모든 계산 시간 영역에서 가장 낮은 항행 거리를 기록하였다. 시간의 변화에 따른 거리의 변화가 크지 않았다. DSSA-C의 경우, 모든 계산 시간 영역에서 가장 긴 항행 거리를 기록하였다. DSSA-E는 C에 이어서 두 번째로 긴 항행 거리를 기록하였다. DSSA-C와 E의 경우, 가장 긴 항행 거리를 기록한 이유는 침로를 탐색할 때, 충돌 위험이 없더라도, 즉 현재 상태가 안전함에도 일정한 확률로 다른 코스를 선택하기 때문인 것으로 보인다.

(2) 메시지 교환 횟수

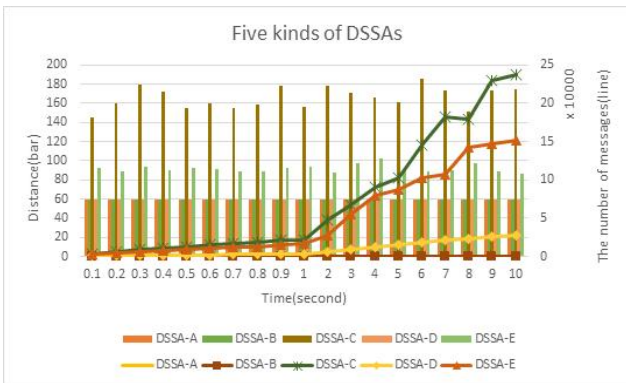


Fig. 3. Distance and the number of messages for five kinds of DSSAs.

Fig. 3의 메시지 교환 횟수(라인)의 경우, DSSA-A, B는 DSSA-C, D, E보다 메시지 교환 횟수가 매우 낮게 기록되었다. DSSA-C, D, E의 경우, 메시지 교환 횟수가 선형적으로 증가하는 경향을 보였다. DSSA-C의 경우, 모든 계산 시간에서 가장 큰 메시지 교환 횟수를 기록하였다. DSSA-D와 E의 경우, 충돌 위험이 없는 경우에도 일정한 확률로 다른 코스를 선택하기 때문에 항행 거리가 길어지며 이에 따라 메시지 교환 횟수도 늘어나는 것으로 보인다.

(3) 비용 변화

Fig. 4는 계산 시간에 따른 비용의 변화 정도와 충돌 횟수를 의미한다. 막대그래프는 비용을 의미하며 라인은 충돌 횟수를 의미한다. DSSA-A와 B의 경우(막대그래프), DSSA-C, D, E에 비해 모든 영역에서 가장 낮은 비용을 기록하였다. DSSA-C의 경우, 모든 계산 시간 영역에서 가장 큰 비용을 기록하였다. DSSA-D의 경우, 세 번째로 높은 비용을 기록하였으며 0.6초에서 가장 낮은 값을 기록하였으며 0.7초에서

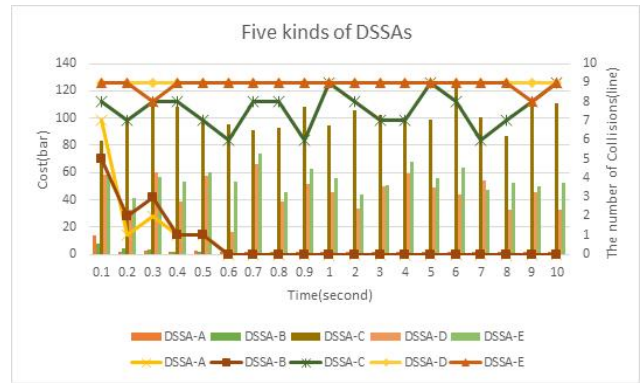


Fig. 4. Cost and the number of collisions for five kinds of DSSAs.

가장 큰 값을 기록하였다. DSSA-E의 경우, 전체적으로 두 번째로 높은 값을 기록하였다.

(4) 충돌 횟수

Fig. 4의 충돌 횟수(라인)의 경우, DSSA-A와 B는 0.5초까지 충돌 횟수가 낮아지는 경향을 보이다가 0.6초 이후부터는 충돌이 발생하지 않았다. DSSA-C의 경우, 모든 영역에서 충돌이 발생하였으며 시간의 경과에 따라 충돌 횟수가 변동하는 경향을 보였다. DSSA-D와 E의 경우, 거의 모든 계산 시간 영역에서 가장 큰 충돌 횟수를 기록하였다. DSSA-C, D, E에서 충돌이 발생한 원인으로 DSA의 해를 선택하는 방법에 있는 것으로 보인다. DSA-C, D, E의 경우, 각각의 객체가 해를 선택하려고 할 때 다른 객체와 충돌이 있거나 현재보다 더 나은 해가 없는 경우라도 일정한 확률로 다른 해를 선택하는 경우가 있다. 이는 현재 상태에서는 문제가 발생하더라도 다음 상태에서는 더 나은 해를 찾을 가능성을 염두에 두기 때문이다.

5.1.2 확률에 따른 분석

Fig. 5은 확률의 변화에 따른 항행 거리와 메시지 교환 횟수를 의미한다. 막대그래프는 항행 거리를 의미하며, 라인은 메시지 교환 횟수를 의미한다. Fig. 6은 확률의 변화에 따른 비용과 충돌 횟수의 변화를 의미한다. 막대그래프는 비용을 의미하며, 라인은 선박 간 충돌 횟수를 의미한다.

(1) 항행 거리

Fig. 5에서 DSSA-A, B, D의 경우(막대그래프), 거의 모든 확률분포영역에서 일정한 항행 거리를 기록하였다. DSSA-C의 경우, 모든 확률분포영역에서 가장 긴 항행 거리를 기록하였다. DSSA-E의 경우, 확률이 커질수록 항행 거리가 증가하는 경향을 보였다. DSSA-C와 E의 경우, 새로운 해를 탐색

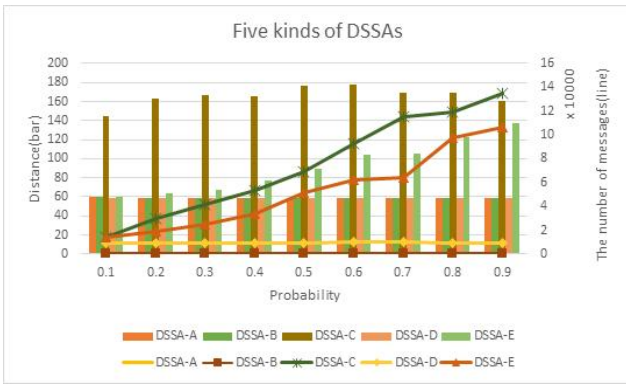


Fig. 5. Distance and the number of messages for five kinds of DSSAs.

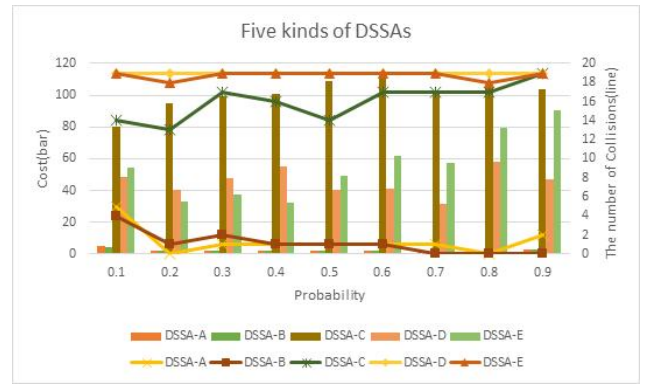


Fig. 6. Cost and the number of messages for five kinds of DSSAs.

하기 위해 현재 상태에 만족하거나 또는 더 나은 해가 없어도 확률 $p$ 만큼 새로운 해를 탐색하는 것이 가능하다. 이러한 이유로 DSSA-A, B, D에 비해 더 긴 항행거리를 기록한 것으로 보인다.

**(2) 메시지 교환 횟수**

Fig. 5에서 DSSA-A, B, D의 경우(라인), 모든 확률분포영역에서 일정한 값을 보였다. DSSA-A와 B는 가장 낮은 메시지 교환 횟수를 기록하였다. DSSA-C와 E는 확률이 증가함에 따라 선형적으로 증가하는 경향을 보였으며 DSSA-C는 모든 확률분포영역에서 가장 많은 메시지 교환 횟수를 기록하였다.

**(3) 비용 변화**

Fig. 6은 확률의 변화에 따른 비용의 변화 정도와 충돌 횟수를 의미한다. 비용의 변화에 대해 DSSA-A와 B의 경우(막대그래프), 모든 확률분포영역에서 가장 낮은 비용을 기록하였다. DSSA-C는 모든 확률분포영역에서 가장 높은 비용을 기록하였다. DSSA-E의 경우 0.4에서 가장 낮은 값을 기록하였으며 그 이후 확률이 커질수록 선형적으로 증가하는 경향을 보였다.

**(4) 충돌 횟수**

Fig. 6에서 DSSA-A의 경우(라인), 0.2와 0.8의 확률에서 충돌이 발생하지 않았다. DSSA-B의 경우, 0.7과 0.8의 확률에서 충돌이 발생하지 않았다. DSSA-C, D, E의 경우, 모든 확률분포에서 충돌이 발생하였으며 그 중 DSSA-D는 모든 영역에서 가장 많은 충돌 횟수를 기록하였다.

DSSA-A의 경우, 0.2 또는 0.8의 확률을 가지고 0.6초 이상의 계산 시간을 가장 좋은 결과를 기대할 수 있다. DSSA-B의 경우, 0.7 또는 0.8의 확률을 가지고 0.6초 이상의 계산 시간

을 설정할 경우, 가장 좋은 결과를 기대할 수 있다. DSSA-C, D, E의 경우, 선박 충돌을 위해서 적용할 만한 알고리즘 방식이 아니라는 결론을 내렸다.

**5.2 12척의 선박이 조우할 경우**

두 번째 실험에서는 Fig. 7과 같이 12척의 선박으로 진행하였다. 3척의 선박이 그룹을 이루어 각각 동서남북에서 출발하여 모든 선박이 중앙에서 교차하도록 설정하였다. 첫 번째 실험의 결과를 참고하여 DSSA-C, D, E는 실험에서 배제하였다. 모든 계산 시간 및 확률분포영역에서 충돌이 발생하여 충돌 회피 알고리즘에 적용하기에는 의미가 없는 것으로 판단하였기 때문이다. Table 3은 각 선박의 초기 설정값이다. 모든 선박은 12 knots의 일정한 속력으로 출발 위치에서 목적지까지 항행하도록 설정하였다.

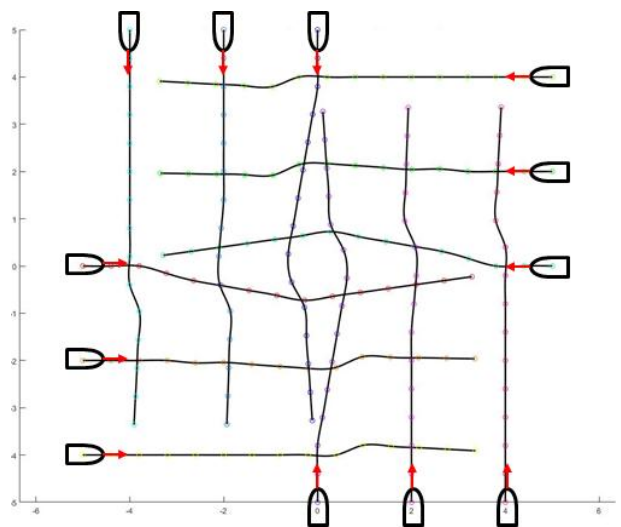


Fig. 7. When 12 ships are encountered.

Table 3. Values for parameters of twelve ships

Parameter	Value
Origin	-5,0/-5,-2/-5,-4/5,4/5,2/5,0/-4,5/-2,5/0,5/0,-5/2,-5/4,-5
Destination	5,0/5,-2/5,-4/-5,4/-5,2/-5,0//4,-5/-2,-5/0,-5/0,5/2,5/4,5
Course	090,090,090,270,270,270,180,180,180,000,000,000
Speed	12 knots (constant)

5.2.1 계산 시간에 따른 분석

Fig. 8은 DSSA-A와 B에 대한 계산 시간의 변화에 따른 항행 거리와 메시지 교환 횟수를 의미한다. 막대그래프는 항행 거리를 의미하며, 라인은 메시지 교환 횟수를 의미한다. Fig. 10은 DSSA-A와 B에 대한 확률의 변화에 따른 비용과 충돌 횟수의 변화를 의미한다. 막대그래프는 비용을 의미하며, 라인은 선박 간 충돌 횟수를 의미한다.

(1) 항행 거리

Fig. 8은 계산 시간의 변화에 따른 항행 거리와 메시지 교환 횟수를 의미한다. 항행 거리에 대해 DSSA-A의 경우(막대 그래프), 계산 시간이 짧을수록 항행 거리가 길어지는 경향을 보였다. 계산 시간이 0.9초 이후로는 일정한 값을 기록하였다. DSSA-B의 경우, 0.1초에서 가장 긴 항행 거리를 기록하였다. 이후 계산 시간이 길어질수록 항행 거리는 줄어드는 경향을 보였다. 2초 이후부터는 일정한 항행 거리를 기록하였다.

(2) 메시지 교환 횟수

Fig. 8의 DSSA-A의 경우(라인), 모든 계산 시간에서 비슷한 메시지 교환 횟수를 기록하였다. DSSA-B의 경우 0.1초에 가장 큰 값을 기록하였다. 그 이후 0.4초까지 줄어들다가 그 이후부터는 일정한 값을 기록하였다.

(3) 비용 변화

Fig. 9는 계산 시간의 변화에 따른 비용의 변화와 선박 간 충돌 횟수를 의미한다. DSSA-A의 경우(막대그래프), 0.2초의 계산 시간에서 가장 큰 비용을 기록하였다. 2초까지는 줄어드는 경향을 보였으며 그 이후에는 일정하게 작은 값을 기록하였다. DSSA-B의 경우, 0.1초에 가장 큰 비용을 기록하였으며 2초까지는 줄어드는 경향을 보였으며 그 이후에는 DSSA-A와 같이 일정하게 작은 값을 기록하였다.

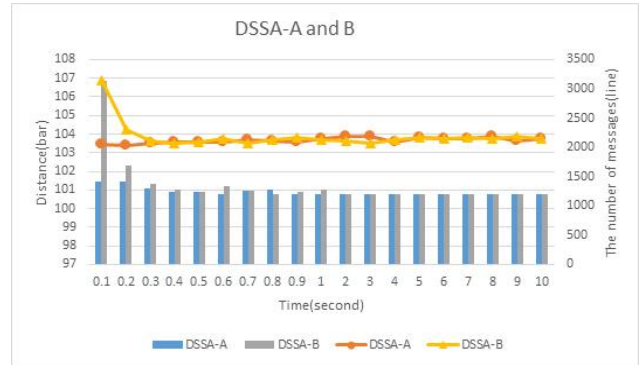


Fig. 8. Distance and the number of messages for DSSA-A and B.

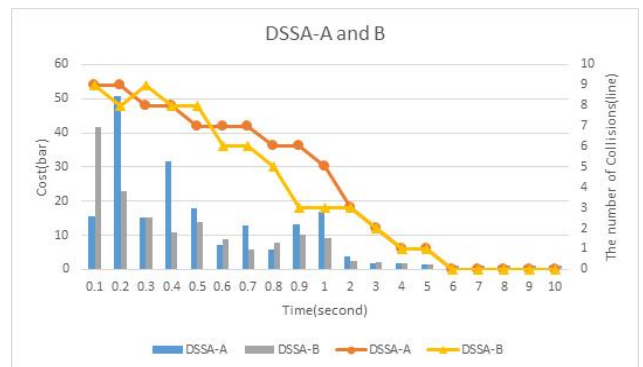


Fig. 9. Cost and the number of collisions for DSSA-A and B.

(4) 충돌 횟수

Fig. 9의 충돌 횟수(라인)에 대해 DSSA-A와 B 모두 5초까지는 충돌 횟수가 감소하는 경향을 보여줬다. 6초 이후에는 모든 확률 분포에서 충돌이 발생하지 않았다. 이전의 실험과 비교하여 선박의 수가 늘어날 경우 충돌 회피를 위한 계산 시간이 증가함을 확인할 수 있다.

5.2.2 확률에 따른 분석

Fig. 10은 DSSA-A와 B에 대한 확률의 변화에 따른 항행 거리와 메시지 교환 횟수를 의미한다. 막대그래프는 항행 거리를 의미하며, 라인은 메시지 교환 횟수를 의미한다. Fig. 12는 DSSA-A와 B에 대한 확률의 변화에 따른 비용과 충돌 횟수의 변화를 의미한다. 막대그래프는 비용을 의미하며, 라인은 선박 간 충돌 횟수를 의미한다.

(1) 항행 거리

Fig. 10은 확률에 따른 항행 거리와 메시지 교환 횟수를 의미한다. DSSA-A의 경우(막대그래프) 가장 긴 항행 거리를 기록하였다. 0.2 이후부터는 일정한 값을 기록하였다. DSSA-B의 경우, 0.1의 확률에서 가장 긴 항행 거리를 기록하였다. 0.3까지 줄어들다가 이후부터는 일정한 값을 기록하였다.

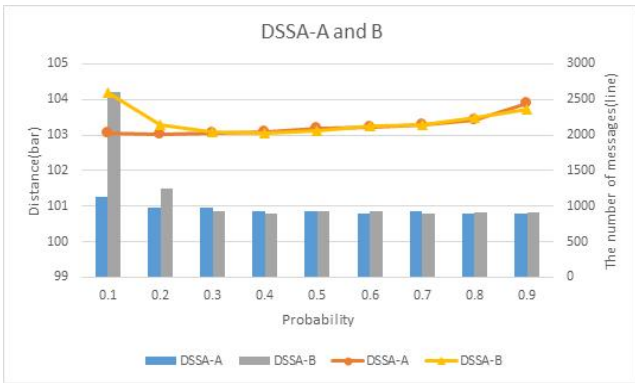


Fig. 10. Distance and the number of messages for DSSA-A and B.

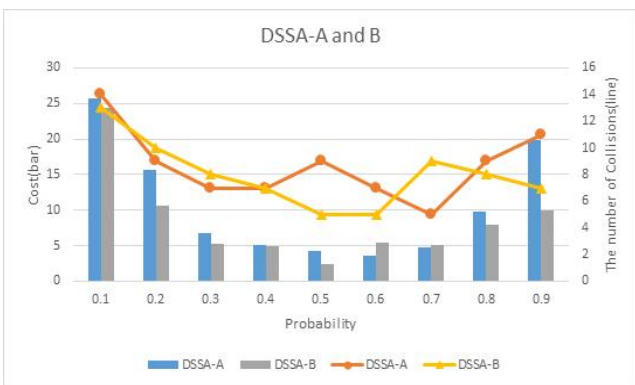


Fig. 11. Cost and the number of collisions for DSSA-A and B.

**(2) 메시지 교환 횟수**

Fig. 10의 메시지 교환 횟수(라인)에 대해 DSSA-A의 경우, 0.1의 확률에서 가장 낮은 메시지 교환 횟수를 기록하였다. 이후 확률이 커질수록 메시지 교환 횟수는 점점 증가하는 경향을 보였다. DSSA-B의 경우, 0.1에서 0.3의 확률에서는 메시지 교환 횟수가 감소하는 경향을 보이다가 0.4부터 다시 증가하는 경향을 보였다.

**(3) 비용변화**

Fig. 11은 확률의 변화에 따른 비용의 변화와 충돌 횟수를 의미한다. DSSA-A의 경우(막대그래프), 0.1의 확률에서 가장 높은 비용을 기록하였으며 그 이후 0.6까지는 감소하는 보였다. 그 이후 다시 증가하는 경향을 보였다. DSSA-B의 경우, DSSA-A의 경우와 유사한 경향을 보였다. 그러나 전체적인 비용은 낮은 값을 기록하였다.

**(4) 충돌 횟수**

Fig. 11의 충돌 횟수에 대해 DSSA-A의 경우(라인), 모든 확률영역에서 충돌이 발생하였다. 0.1에서 가장 높은 충돌 횟

수를 기록하였으며 0.7에서 가장 낮은 충돌 횟수를 기록하였다. DSSA-B의 경우, 모든 영역에서 충돌이 발생하였다. 0.1에서 가장 높은 충돌 횟수를 기록하였다. 0.5와 0.6에서 가장 낮은 충돌 횟수를 기록하였다.

12척의 선박이 Fig. 8과 같이 항행할 때, DSSA-A의 경우, 0.1초에서 0.3초를 제외하고 계산 시간에 따른 항행 거리와 메시지 교환 횟수의 차이는 없었다. 계산 시간이 6초 이상일 경우 충돌은 발생하지 않았다. 0.1의 확률을 제외하고는 모든 확률에서 일정한 값을 기록하였다. 메시지 교환 횟수는 확률이 커질수록 증가하는 경향을 보였다. DSSA-A의 경우, 0.7의 확률을 가지고 6초의 계산 시간을 설정할 경우, 가장 안전하고 최적의 값을 기록하였다.

DSSA-B의 경우, 0.5 또는 0.6의 확률을 가지고 6초의 계산 시간을 설정할 경우, 가장 안전하고 최적의 값을 기록하였다. 메시지 횟수에 따라 분석하면, 확률 0.5의 경우 B의 메시지 교환 횟수가 적었으며, 0.6, 0.7, 0.8의 경우 A의 경우 더 적은 메시지 수를 기록하였다.

**6. 결론**

본 논문에서는 다섯 가지 버전의 DSSA를 선박 충돌 피항에 적용하였다. DSSA가 선박 간 충돌 방지에 적용 가능한지 알아보기 위해 두 가지의 실험을 실시하였다. 실험 결과 다섯 가지 버전의 DSSA중 DSSA-A와 B방식이 모든 실험에서 효과적으로 선박 충돌 피항을 수행하였다.

먼저 Fig. 2와 같이 5척의 선박이 항행할 경우, DSSA-A와 B는 선박 충돌에 적용 가능성을 증명하였다. 그러나 DSSA-C, D, E의 경우, 해를 탐색하는 방법의 특성상 항행 거리가 길어지며 메시지 교환 횟수가 증가하였다. 또한 거의 대부분의 계산 시간 및 확률분포영역에서 선박 간 충돌이 발생하였다. 이는 선박 간 충돌 방지에 적합하지 않은 알고리즘으로 판단된다.

DSSA-A 방식의 경우, 0.2 또는 0.8의 확률을 가지고 0.6초 이상의 계산 시간을 가장 좋은 결과를 기대할 수 있으며 DSSA-B 방식의 경우, 0.7 또는 0.8의 확률을 가지고 0.6초 이상의 계산 시간을 설정할 경우, 가장 좋은 결과를 기대할 수 있다.

Fig. 7과 같이 12척의 선박이 항행할 경우의 실험에서 DSSA-A와 B방식은 선박 충돌 방지에 효용성이 있음을 증명하였다. 5척의 경우와 비교하였을 때 선박이 증가함에 따라 계산 시간도 증가함을 보였다. 이 실험에서 DSSA-A와 B는 계산 시간이 6초 이상, 확률이 0.5와 0.8 사이일 경우 충돌 피항에 응용 가능성을 보였다.



본 논문에서는 모든 선박이 협조적일 경우를 가정하여 실험을 진행 하였다. 추후 연구 과제로 일부의 선박 또는 모든 선박들이 비협조적인 상황도 발생할 가능성이 있다. 이에 대한 대비책이 필요할 것으로 보인다.

---

Received : 2019. 03. 11.

Revised : 2019. 04. 15.

Accepted : 2019. 04. 26.

## References

- [1] COLREGS(1972), Convention on the International Regulations for Preventing Collisions at Sea. International Maritime Organization, London.
- [2] Fujii, Y. and K. Tanaka(1971), Traffic Capacity, The Journal of Navigation, Vol. 24, No. 4, pp. 543-552.
- [3] Glover, F.(1989), Tabu Search-Part I. ORSA, Journal on Computing, Vol. 1, No. 3, pp. 190-206.
- [4] Goodwin, E. M.(1975), A Statistical Study of Ship Domains, The Journal of Navigation, Vol. 28, No. 3, pp. 328-344.
- [5] Hornauer, S.(2013), Decentralised Collision Avoidance in a Semi-Collaborative Multi-Agent System, Multiagent System Technologies: 11th German Conference, MATES 2013, pp. 412-415.
- [6] Hornauer, S., A. Hahn, M. Blaich and J. Reuter(2015), Trajectory Planning with Negotiation for Maritime Collision Avoidance, International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 9, No. 3, pp. 335-341.
- [7] Hu, Q., C. Yang, H. Chen and B. Xiao(2008), Planned Route Based Negotiation for Collision Avoidance between Vessels, The International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 2, No. 4, pp. 363-368.
- [8] Kim, D., K. Hirayama and G. Park(2014), Collision Avoidance in Multiple-Ship Situations by Distributed Local Search, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 18, No. 5, pp. 839-848.
- [9] Kim, D., K. Hirayama and T. Okimoto(2015), Ship Collision Avoidance by Distributed Tabu Search, The International Journal on Marine Navigation and Safety of Sea Transportation, Vol. 9, No. 1, pp. 23-29.
- [10] Kim, D., K. Hirayama and T. Okimoto(2017), Distributed Stochastic Search Algorithm for Multi-ship Encounter Situations, The Journal of Navigation, Vol. 70, pp. 699-718.
- [11] Zhang, W., G. Wang, Z. Xing and L. Wittenburg(2005), Distributed stochastic search and distributed breakout: Properties, comparison and applications to constraint optimization problems in sensor networks, Artificial Intelligence. Vol. 161, No. 1-2, pp. 55-87.