

## 격자 지도 기반의 다수 무인 이동체 임무 할당 및 경로 계획

정병민<sup>1</sup>, 장대성<sup>2</sup>, 황남웅<sup>3</sup>, 김준원<sup>3</sup>, 최한림<sup>1,†</sup><sup>1</sup>한국과학기술원,  
<sup>2</sup>한국항공대학교,  
<sup>3</sup>한화시스템

## Task Allocation and Path Planning for Multiple Unmanned Vehicles on Grid Maps

Byeong-Min Jeong<sup>1</sup>, Dae-Sung Jang<sup>2</sup>, Nam-Eung Hwang<sup>3</sup>, Joon-Won Kim<sup>3</sup> and Han-Lim Choi<sup>1,†</sup><sup>1</sup>Korea Advanced Institute of Science and Technology (KAIST),<sup>2</sup>Korea Aerospace University (KAU),<sup>3</sup>Hanwha Systems Co.

## Abstract

As the safety of unmanned vehicles continues to improve, their usage in urban environments, which are full of obstacles such as buildings, is expected to increase. When numerous unmanned vehicles are operated in such environments, an algorithm that takes into account mutual collision avoidance, as well as static and dynamic obstacle avoidance, is necessary. In this paper, we propose an algorithm that handles task assignment and path planning. To efficiently plan paths, we construct a grid-based map and derive the paths from it. To enable quick re-planning in dynamic environments, we focus on reducing computational time. Through simulation, we explain obstacle avoidance and mutual collision avoidance in small-scale problems and confirm their performance by observing the entire mission completion time (Makespan) in large-scale problems.

## 초 록

무인 이동체의 안전성이 점차 증대되면서 빌딩과 같은 장애물이 많은 도심환경에서의 무인 이동체의 활용이 증가할 것으로 예상된다. 도심 환경에서 다수의 무인 이동체가 운용될 경우, 임무 할당 뿐만 아니라 정적 및 동적 장애물 회피와 더불어 상호 충돌 회피가 가능한 경로를 생성하는 알고리즘이 필요하다. 본 논문에서는 임무 할당 및 경로 계획을 수행하는 알고리즘을 제안한다. 장애물 및 경로 계획을 효율적으로 수행하기 위해 맵을 격자 기반으로 구성한 다음 경로를 도출하였다. 동적인 환경에서 빠르게 재계획하기 위해 계산 시간 단축에 집중하였다. 시뮬레이션을 통해 작은 규모의 문제에서 장애물 회피 및 상호 충돌 회피 방안에 대해 설명하였고, 큰 규모의 문제에서 임무 전체 종료 시간(Makespan)을 관찰하여 성능을 확인하였다.

**Key Words** : Unmanned Vehicle(무인 이동체), Task Allocation(임무 할당), Path Planning(경로 계획)

## 1. 서 론

향후 무인 이동체의 안전성이 점차 증대됨에 따라 도심 환경에서의 이용이 많아질 것이다. 도심 환경에서의

Received: Jan. 18, 2024 Revised: Mar. 11, 2024 Accepted: Mar. 12, 2024

† Corresponding Author

Tel: +82-42-350-3727, E-mail: hanlimc@kaist.ac.kr

© The Society for Aerospace System Engineering

다수 무인 이동체는 다양한 임무에 이용될 수 있다. 예를 들면, 도심 내 감시 정찰[1], 무선 통신중계[2], 지도 작성을 위한 데이터 획득[3] 등이 있다.

도심 환경에서 무인 이동체를 운용하기 위해서는 빌딩 등의 장애물을 고려한 임무 할당 및 경로 계획 기법이 필요하다. 또한 비행 영역이 넓지 않기 때문에 서로 충돌이 발생하지 않도록 경로 계획을 수행해야 한다.

임무 할당 및 충돌 회피 경로 계획에 대한 연구는 다양한 분야에서 수행되었다. 다수의 고정된 무인기를 이용한 임무 수행 시 충돌 회피를 위한 프레임 워크를 제시한 연구[4]도 있고, 공장 환경에서 로봇 간 충돌 방지 및 임무할당을 위한 알고리즘에 대한 연구[5]도 있다. 또한, 분산화된 기법을 이용하여 임무를 수행하면서 상호 충돌 및 동적 장애물에 대한 충돌을 회피하는 기법에 대한 연구[6]도 있다.

무인 이동체는 일반적으로 경로를 계획할 때, 출발지와 목적지를 직선으로 잇는 경로를 주로 사용한다. 비행이 가능한 무인 이동체는 장애물에 구애를 받지 않기 때문이다. 그리고 평원에서의 임무를 계획할 때에도 직선으로 잇는 경로만으로 충분하다. 하지만 빌딩이 들어선 도심 환경에서는 적합하지 않은 경로 계획이다. 장애물이 있는 환경에서의 다수의 로봇에 대한 경로계획 문제는 CBS - MAPF[7] 등이 있다.

본 논문에서는 장애물이 있는 환경에서의 다수의 무인 이동체가 다수의 임무를 충돌없이 수행할 수 있는 경로의 생성하는 방안을 연구하였다. 이에 대한 최적해를 구하기 위해서는 상당히 많은 시간이 소요된다. 동적인 환경에서는 장애물이나 상대 위치가 지속적으로 변화하기 때문에 이에 대응하기 위해서는 계산 시간을 단축할 수 있는 방안이 중요하다고 판단하였고, 계산 시간 단축을 위한 알고리즘을 제시하였다. 이 알고리즘은 두 단계로 구분되어 첫 번째 단계는 임무 할당, 두 번째 단계는 충돌 회피 경로 생성을 각각 수행한다. 첫 번째 단계에서 휴리스틱을 기반으로 알고리즘을 작성하여 계산 속도를 향상시켰다.

본 논문에서는 임무 할당 및 경로 계획 문제를 건물과 같은 밀집한 장애물 환경에서 다루었다. 비행체로 3차원 공간에서 작동 가능하지만, 드론의 아래쪽으로 강한 바람이 나가는 특성 때문에 2차원 격자 환경으로 간주하였다. 또한 상호 간의 충돌을 피하기 위한 기동도 고려했다. 2차원 격자 환경에서 해결되었기 때문에 이 방법은 무인 항공기뿐만 아니라 지상 무인 차량에도 적용 가능하다. '임무 할당 및 경로 계획 알고리즘'이라는 긴 이름 대신 GMTPA(Gird-based Multi-agent Task allocation and Path planning Algorithm)라는 영어 약어로 본 알고리즘을 지칭한다.

본 논문은 아래와 같이 구성된다. 2장에서 논문에서 다룰 임무계획 문제에 대해 설명하고 정식화한다. 3장에서는 본 논문에서 제시하는 기법에 대해 설명하고, 4장에서는 시뮬레이션 결과를 통해 기법의 유효성을 확인한다. 그리고 5장에서 논문의 결론을 서술한다.

## 2. 문제 정의 및 정식화

### 2.1 문제 정의

무인 이동체들의 임무 환경은 2차원 격자 공간 위에 표현되며, 장애물이 있는 조건에서 상호 충돌 없이 임무를 수행하는 것이 목적이다. 그리고 모든 무인 이동체들은 임무를 수행하고 나서 출발지로 돌아와야 한다. 동시에 임무 전체 종료 시간을 최소화할 수 있도록 임무를 할당해야 하고, 충돌이 없는 경로를 찾아야 한다.

### 2.2 문제 정식화

다수 무인 이동체 체계를 구성하는 이동체의 총 수는  $N_u$ 이고, 임무의 수는  $N_t$ 로 정의한다. 2차원 격자로 구획된 공간에서 이동체와 임무의 위치는 정수값을 가지는 벡터( $[x, y]^T$ ,  $x, y \in \mathbb{Z}$ )로 표현된다. 이동체가 임무를 수행하는 동안 진행되는 시간  $t$  역시 이산화된다. 이산화된  $s$ 번째 스텝에서의 시간을  $t_s$ 로 표기하며, 이동체들이 임무를 시작할 때  $s = 0$ ,  $t_0 = 0$ 이다. 위의 개요에서 정의한 문제는 아래 식 (1~7)를 통해 정식화할 수 있다.  $T_i \in \mathbf{T}$ 는  $i$ 번째 무인 이동체가 임무를 전체 수행하고 출발지로 돌아오는데 걸리는 시간을 의미한다. 따라서 임무 전체 종료 시간을 최소화하는 문제는 식 (1)과 같이 목적함수  $T_{\max} = \max_{i \in I} T_i$ 를 최소화하는 문제가 된다.

$$\min_{r, P, T} T_{\max} \quad (1)$$

subject to:

$$\sum_{j=1}^{N_t} r_{ij} \leq N_t, \forall i \in I \quad (2)$$

$$\sum_{i=1}^{N_u} r_{ij} = 1, \forall j \in J \quad (3)$$

$$P_{i, t_s} \neq P_{k, t_s}, \forall i, k \in I, \forall t_s \quad (4)$$

$$p_j r_{ij} = P_{i, t_s} r_{ij}, \exists t_s \quad (5)$$

$$\|P_{i, t_s} - P_{i, t_{s+1}}\|_2 \leq 1, \forall i \in I, \forall t_s \quad (6)$$

$$P_{i, t_0} = P_{i, t_s}, \forall i \in I, \forall t_s \geq T_i \quad (7)$$

$r_{ij} \in \mathbf{r}$ 는  $i$ 번째 이동체가  $j$ 번째 임무를 할당 받을 때 1 그렇지 않을 때 0이 되는 이진 변수이다. 이동체  $i$ 가 최대 수 수행할 수 있는 임무의 수는  $N_i$ 와 같다고 설정한다(식 (2)). 그리고 임무  $j$ 는 오직 하나의 이동체에 할당될 수 있다. (식 (3)).  $\mathbf{P}_i \subset \mathbf{P}$ 는  $i$ 번째 이동체가 움직인 경로에 속하는 위치 좌표 벡터의 집합이고,  $\mathbf{P}_{i,t_s} := [x_{i,t_s}, y_{i,t_s}]^T \in \mathbf{P}_i$ 는  $i$ 번째 이동체가 시간  $t_s$ 일 때 격자 상의 위치를 뜻한다. 이동체 간의 충돌은 같은 시간에 두 이동체가 같은 위치에 있을 수 없다는 조건(식 (4))으로 제한된다.  $\mathbf{p}_j := [u_j, v_j]^T$ 는  $j$ 번째 임무의 위치를 뜻한다.  $i$ 번째 이동체가  $j$ 번째 임무를 맡았다면( $r_{ij} = 1$ ),  $i$ 번째 이동체의 경로 상에 임무 위치가 포함되어야 한다. 이는 식 (5)으로 표현된다. 모든 이동체는 시간 스텝이 증가할 때 격자 상에서 전후좌우 중 한 칸 이동하거나 제자리에서 대기할 수 있다(식 (6)). 이동체가 임무를 종료하고 출발지  $\mathbf{P}_{i,t_0}$ 에 돌아온 이후 이동체는 계속 그 자리에 있는 것으로 간주한다. 따라서 이동체  $i$ 가 임무를 종료하고 출발지점으로 돌아온 시간  $T_i$ 는 식 (7)을 만족하면서 식 (1)에 의해 그러한  $T_i$  중 가장 작은 값이 된다. 이 최적화의 결정변수는  $r_{ij}$ 와  $\mathbf{P}_{i,t_s}$  그리고  $T_i$ 이다. 식 (7)은 수식을 적용하는 조건에 결정변수가 포함되므로, 이를 피하고 결정변수에 대한 제한조건을 명시적으로 표현하기 위해서 다음과 같은 식들을 식 (7) 대신 사용할 수 있다.

$$\|\mathbf{P}_{i,t_0} - \mathbf{P}_{i,t_s}\|_2(t_s - T_i) \leq 0, \quad \forall i \in I, \forall t_s \quad (8)$$

$$\|\mathbf{P}_{i,t_0} - \mathbf{P}_{i,t_s}\|_2 \leq M|t_s - T_i|, \quad \forall i \in I, \forall t_s \quad (9)$$

여기서  $M$ 은  $t_s \neq T_i$ 일 때, 항상 식 (9)를 만족하게 하는 충분히 큰 양의 상수이다.

### 2.3 문제 특성 및 풀이 기법

2.1절에서 제시한 문제의 경우, 임무와 무인 이동체의 개수가 증가할수록 최적해를 구하기가 어렵다. 식 (1~7)에서 정식화한 문제는 비선형 정수계획이므로 일반적인 정수계획법 솔버를 이용하거나, 마르코프 결정과정으로 모델링하여 해를 구할 수 있다. 그러나 이러한 일반적인 접근법으로 최적해를 구하기 위해서는 많은 계산 시간이 소요된다. 기본적으로 이 논문에서 다루고 있는 문제는 임무의 수행 순서를 정하는 할당 문제이므로 공간과 시간이 이산화된 TSP(Traveling Salesman Problem) 또는 VRP(Vehicle Routing Problem)의 변형이다. 따라서 NP-hard이며 추가적으로 경로계획까지 하기 위해서는 매우 많은 계산량이 필요하다.

다수의 무인 이동체를 운용하는 상황에서는 장애물의 형상이나 임무가 변경되는 등 동적으로 환경이 변하는 경우가 많으며, 정밀한 배치와 경로 계획을 위해 세밀한 간격의 격자가 필요할 수도 있다. 따라서 이와 같이 문제 크기 증가에 따라 계산량이 기하급수적으로 증가하는 NP-hard 문제에서 반복적인 풀이가 요구된다면 최적해를 구하는 대신에 효과적인 해를 빠른 시간에 도출할 수 있는 휴리스틱 기반 알고리즘이 필요하다.

## 3. 임무 할당 및 경로 계획 기법

### 3.1 임무 할당 및 경로 계획 기법의 구조

본 논문에서 제안하는 기법은 1) 휴리스틱 기반 순차적 임무 할당과 2) 충돌 회피 경로 계획의 두 단계로 구성된다. 임무 할당을 진행할 때 임무 간 이동시간을 계산하기 위해서  $A^*$  알고리즘[8]을 이용하여 장애물을 고려한 최단거리를 산출하며 이 때 이동체 간의 충돌은 고려하지 않는다. 충돌을 고려하지 않은 상태에서  $T_{\max}$ 를 줄일 수 있는 임무 할당을 결정한 다음, 이 할당을 수행할 때 예상되는 충돌들을 지엽적으로 해결하는 방식을 반복하여 최종적으로 충돌이 일어나지 않는 경로를 결정한다.

### 3.2 임무 할당 기법

임무와 무인 이동체 및 임무와 임무 간 장애물을 회피하는 최단 경로의 거리와 소요시간 정보가 있을 때, 이동체에 임무를 할당하는 방식은 다음과 같다. 임무 개수만큼 루프를 반복하면서 그리디한 규칙에 따라 매 루프마다 1개씩의 임무를 이동체에 할당한다. 따라서 루프가 반복되면서 이동체들의 임무 수행 계획은 점진적으로 증가하며, 이전 루프에서 결정된 기존 계획에 임무를 추가한다. 각 루프에서는 우선 Algorithm 1의 3~9번째 줄과 같이 각 임무  $j$ 를 이동체  $i$ 의  $k$ 번째 수행하는 것으로 계획하였을 때 예상되는 임무 전체 종료 시간  $T_{\max} = \max_{i \in I} T_i$ 의 증가치  $c_{ijk} = T'_{\max} - T_{\max}$ 를 계산한다.  $T_{\max}$ 는 이전 루프까지의 할당에 따른 임무 전체 종료 시간이고,  $T'_{\max}$ 는 추가적으로 임무  $j$ 가 이동체  $i$ 에서  $k$ 번째로 수행될 때의 임무 전체 종료 시간이다. 이 때 만약 이동체  $i$ 가 수행하기로 계획한 임무의 개수가  $n_i$ 라면, 새로운 임무를 추가할 때 그 임무의 수행 순서를 첫번째(1)에서부터 이전 계획 바로 뒤( $n_i + 1$ )까지만 고려한다. 임무가 중간 순서에 삽입되는 경우 기존의 임무들은 삽입된 임무 뒤에 수행된다고 가정하여  $T'_{\max}$ 를 계산한다. 할당되지 않은 임무들의 모든 임무 전체 종료 시간 증가치  $c_{ijk}$ 가 계산되면 그 중 가장 작은 증가치를 가지는 할당  $i^*, j^*, k^*$ 을 선택하고

임무 계획 및  $T_{i^*}$ ,  $T_{\max}$ ,  $n_{i^*}$ 를 갱신한다. 또한 이미 할당된 임무  $j$ 는 고려하지 않기 위해 임무의 집합  $J$ 에서 제거한다.

Algorithm 1을 통해 유한한 시간 안에 임무 할당 및 수행 계획을 완성할 수 있다. 전체 루프는 정확히  $N_t$  번 반복되며, 내부의 반복문(3~9번째 줄)은 최대  $(N_t - n) \max(1, n_i + 1) \leq (N_t - n)(N_u + n)$  번 수행된다. 따라서 알고리즘의 수행시간 점근적 복잡도는  $O(\sum_n^N (N_t - n)(N_u + n)) = O(N_t^2(N_u + N_t))$ 가 된다.

---

**Algorithm 1** Greedy Task Allocation

---

```

1:  $T_i = n_i = 0, \forall i$  and  $T_{\max} = 0$ 
2: for  $n = 1$  to  $N_t$  do
3:   for  $j \in J$  do
4:     for  $i = 1$  to  $N_u$  do
5:       for  $k = 1$  to  $n_i + 1$  do
6:          $c_{ijk} = T'_{\max} - T_{\max}$ 
7:       end for
8:     end for
9:   end for
10:   $i^*, j^*, k^* = \arg \min_{(i,j,k)} c_{ijk}$ 
11:  Assign  $j^*$ , in the  $k^*$ th position in agent  $i^*$ 's
    task sequence & update  $T_{i^*}$  and  $T_{\max}$ 
12:   $J = J \setminus j^*, n_{i^*} = n_{i^*} + 1$ 
13: end for
    
```

---

### 3.3 충돌 회피 경로 계획

임무 할당 단계에서 경로를 계획할 때에 고정 장애물은 고려하지만 무인 이동체 간의 충돌은 고려하지 않는다. 따라서 임무 할당에서 계산한 경로로 무인 이동체들이 진행하면 예상하지 못한 충돌이 발생할 수 있다. 이를 해결하고 충돌이 발생하지 않는 경로를 계획하는 과정은 다음과 같다. 먼저 격자 환경에서  $A^*$  알고리즘을 수행하여 경로를 생성할 때에 각 격자를 방문하는 시간을 기록한다. 모든 임무 할당이 완료되면, 이산화된 시간의 스텝  $s$ 를 증가시키면서 무인 이동체들의 계획된 경로를 따라 이동체들의 위치를 확인한다. 이때 두 대 이상의 이동체가 같은 위치에 있게 되어 충돌이 발생하면 경로를 수정하는 우회(de-tour) 과정을 수행한다. 우회 과정은 충돌에 관계하는 이동체들에 한정하여 다음 임무 위치까지  $A^*$  알고리즘을 다시 적용하는 것이며, 이때 해당 충돌이 일어난 위치와 시간을 장애물로 취급하여 경로를 재생성한다. 만약 반복적으로 경로를 수정하여도 계속 충돌이 발생하는 경우, 충돌하는 이동체 중에 남은 거리가 더 짧아서 양보하더라도 임무 전체 종료 시간 증가에 관여할 가능성이 적은 이동체가 충돌 직전 위치에서 일정 스

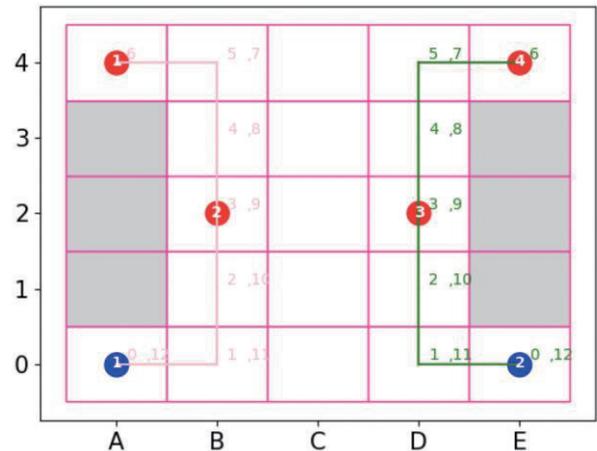
텟 대기한 후 진행하도록 경로를 수정한다.

## 4. 시뮬레이션 및 결과

기법의 특성을 확인하기 위해서 가로 세로 크기가 5인 영역의 문제를 풀었다. 문제를 풀고 도출된 경로를 이용해서 상호 충돌 회피 방안에 대해 설명을 수행하였다. 그리고 가로 세로 크기가 10인 영역의 문제를 풀고 GMPTA의 계산 시간과 임무 전체 종료 시간을 각각 확인하였다. GMPTA의 성능을 확인하기 위해 임무 할당 부분을 참고문헌 [9]를 참조하여 유전 알고리즘으로 대체한 것을 비교군으로 설정하였다. 이 때 사용한 영역은 장애물이 없는 영역, 좁은 통로가 있는 영역, 큰 장애물이 분포한 영역 3가지이다. 그리고 이 중에서 장애물이 없는 영역에 동적 장애물을 배치하여 장애물 회피 방안에 대해 설명하였다.

### 4.1 기법의 특성 확인

Fig.1의 5 × 5 규모의 격자에서는 파란색 점은 무인 이동체, 빨간색 점은 임무의 위치를 나타낸다. 회색 격자 영역은 장애물을 나타낸다. 그리고 모든 무인 이동체는 1초에 한 칸씩 이동하는 것으로 설정하였다. 별도로 최적해를 구하는 알고리즘을 이용하지 않더라도, 1번 무인 이동체가 임무 1, 2를 선택하고, 2번 무인 이동체가 임무 3, 4을 선택하여 수행 후 출발지로 복귀하는 것이 최적해인 것이 자명해 보인다. Fig. 2에서 빨간색 선은 0번 무인 이동체가 움직인 경로이고, 빨간 색 숫자는 해당 격자 영역을 방문한 시간을 기록한 것이다.



**Fig. 2** Graphical Illustration of GMPTA on 5 by 5 grid (Without mutual collision)

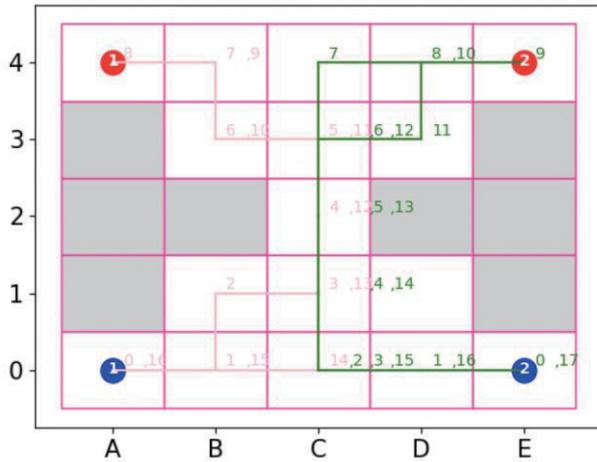


Fig. 3 Graphical Illustration of GMTPA on 5 by 5 grid (With mutual collision)

A\* 알고리즘을 이용하여 경로 길이를 계산하고, 임무 할당 기법을 활용해 임무 할당을 수행한 다음, A\* 알고리즘을 이용해 경로를 계획하였다. 이 경우, 무인 이동체 간 충돌은 발생하지 않기 때문에 기존에 계산한 경로 길이를 계산할 때와 동일하게 경로를 계산한다. 충돌이 발생하는 경우를 Fig. 3과 같이 임의로 조성할 경우, 둘 다 최단 경로를 따를 때 (C,1) 위치에서 충돌이 일어난다. 이 때, 2번 무인 이동체가 (C,0) 위치에서 한 스텝 더 머무르면서 충돌을 회피한다.

4.2 시뮬레이션 주요 변수 설정

시뮬레이션을 수행하기 위해 가로 12, 세로 11 칸의 격자 영역을 정의하고, 임무를 수행할 환경을 3가지 정의했다. 하나는 장애물이 전혀 없는 환경, 하나는 좁은 통로가 있는 환경, 남은 하나는 적절히 장애물이 분포한 환경이다. 이는 Fig. 4 ~ 6에 도시되었다.

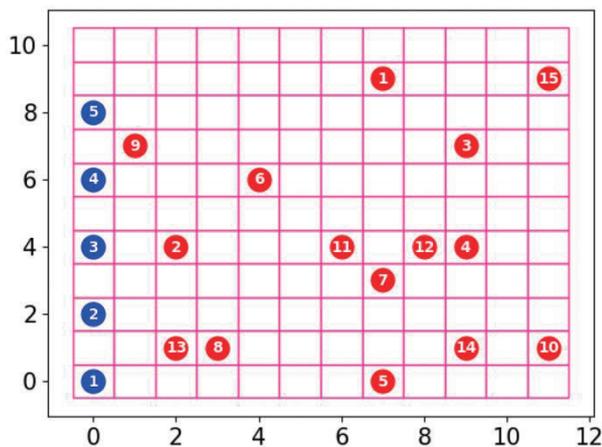


Fig. 4 Empty environment grid map

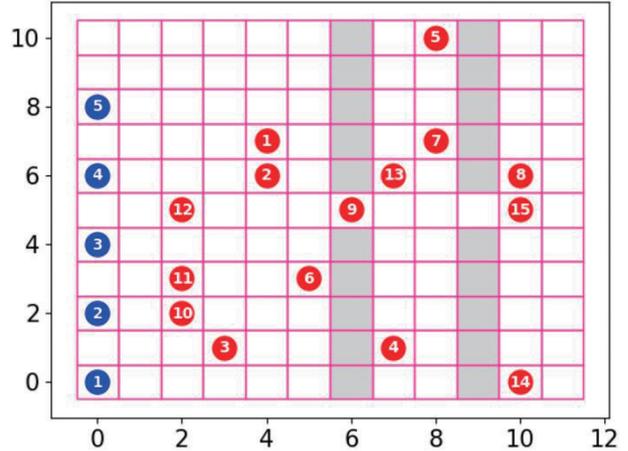


Fig. 5 Narrow gap environment grid map

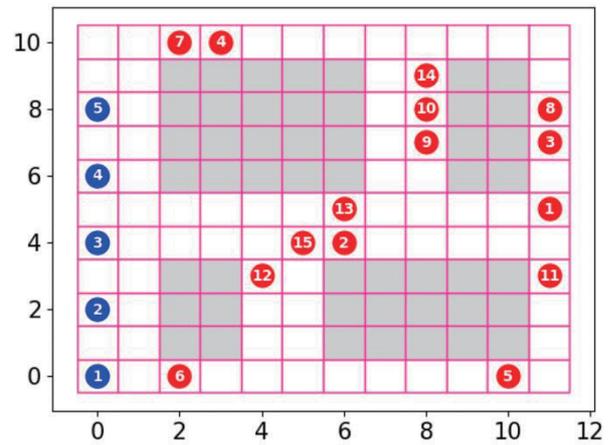


Fig. 6 Obstacle environment grid map

무인 이동체의 위치는 고정되어 있고, 임무 위치는 장애물과 무인 이동체의 위치를 제외한 지역을 무작위로 선정하여 배치될 수 있게 하고 조건 별로 50회 임무 할당 및 경로 계획을 수행하고 그 결과를 비교하였다. 이 때 조건은 무인 이동체의 대수와, 임무의 개수이다. 무인 이동체, 임무 개수 조합은 서로 비율 1:3으로 정하고, 무인 이동체가 2대, 3대, 4대, 5대일 때 시뮬레이션을 수행하였다. 또한 3.4절에서 설명한 유전 알고리즘을 이용한 임무할당[9]에 대한 결과와 비교를 수행하였다. 유전 알고리즘은 임무할당만 수행할 수 있으므로 경로 계획 기법은 제안된 알고리즘(GMTPA) 중 경로계획 부분은 같이 사용하였다.

4.3 시뮬레이션 결과

무인 이동체, 임무 개수 조건 별 임무 전체 종료 시간은 Fig. 7 ~ 9 에 그렸고, 계산 시간은 Fig. 10 ~ 12 에 그렸다. 임무의 개수가 증가할 수록 임무 전체 종료 시간은 감소하고, 계산 시간은 증가한다.

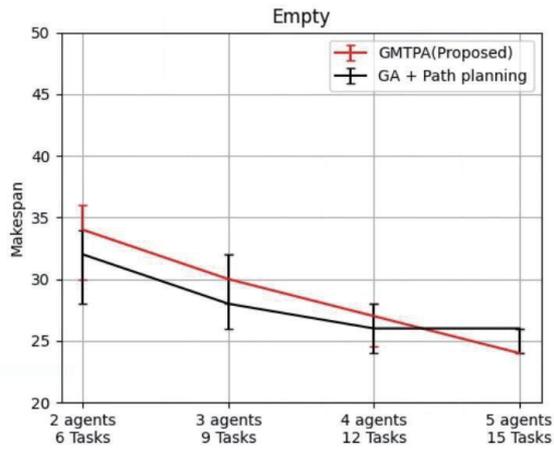


Fig. 7 Makespan at empty environment

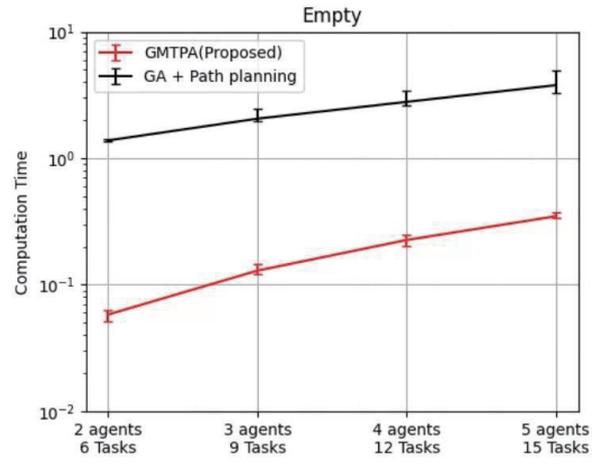


Fig. 10 Computation time at empty environment

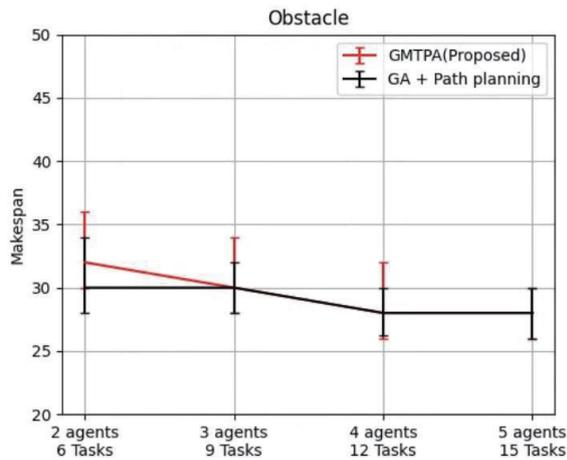


Fig. 8 Makespan at obstacle environment

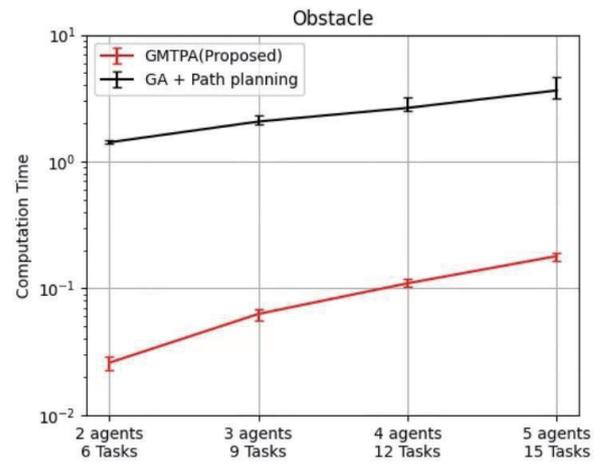


Fig. 11 Computation time at narrow gap environment

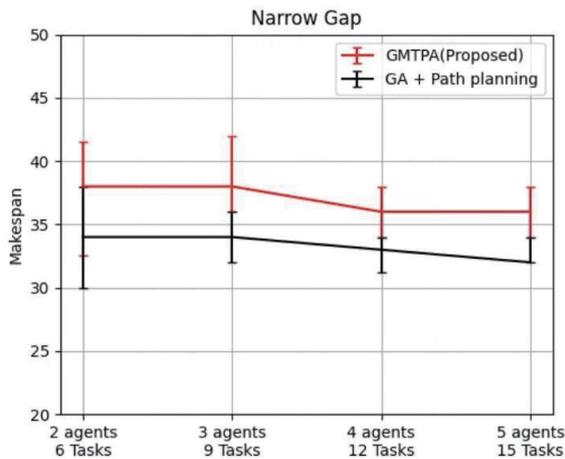


Fig. 9 Makespan at narrow gap environment

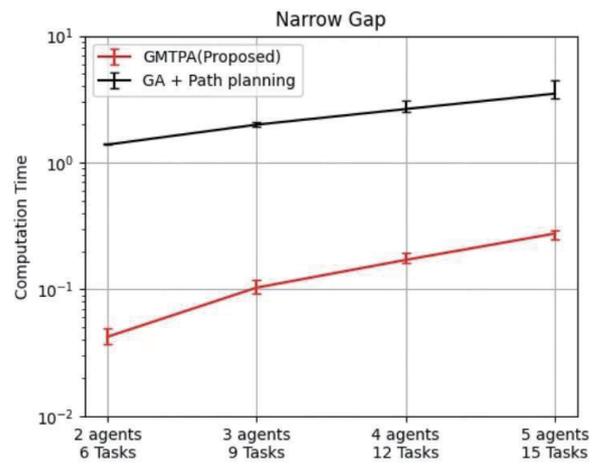


Fig. 12 Computation time at obstacle environment

임무 전체 종료 시간이 감소하는 이유는 무인 이동체의 대수가 많아져서 같은 임무를 맡더라도 효율적인 경로를 선정할 수 있기 때문이다. 계산시간이 증대되는 이유는 고려해야 하는 무인 이동체, 임무의 조합이 증대되기 때문이다. 그리고 Fig. 10 ~ 12 중 제안된 기법의 값을 보면 장애물이 없는 환경(Empty), 좁은 통로가 있는 환경(Narrow gap), 장애물이 분포한 환경(Obstacle)이 순으로 계산시간이 줄어든다. 이는 경로를 계획할 때 장애물이 많으면 탐색해야할 노드의 수가 적어지기 때문이다. 유전 알고리즘의 경우 경로 계획 대비 임무할당에 소모되는 시간이 길어서 계산시간에 영향을 크게 주지 않는다. 무인 이동체가 5대일 때 GMTPA의 평균 계산시간은 0.35 초에서 0.18 초로 거의 절반 정도로 감소하지만, 유전 알고리즘의 경우 평균 계산시간이 3.78초에서 3.64초로 상대적으로 적게 감소한다.

좁은 통로가 있는 환경에서는 탐욕적 방식보다 유전 알고리즘으로 얻은 임무 전체 종료 시간이 4스텝 크다. 장애물이 없는 환경에서는 무인 이동체 수가 적을 때는 제안된 알고리즘의 임무 전체 종료 시간이 2스텝 크지만, 무인 이동체 대수가 5대일 때는 2스텝 작다. 그리고 장애물이 많은 환경에서는 무인 이동체가 2대인 경우를 제외하고는 제안된 알고리즘(GMTPA)의 임무 전체 종료 시간이 2스텝 크지만, 나머지는 다 동일하다. 이처럼 대부분의 환경에서 임무 전체 종료시간은 비슷하게 도출되지만, 계산시간은 10배이상 작다.

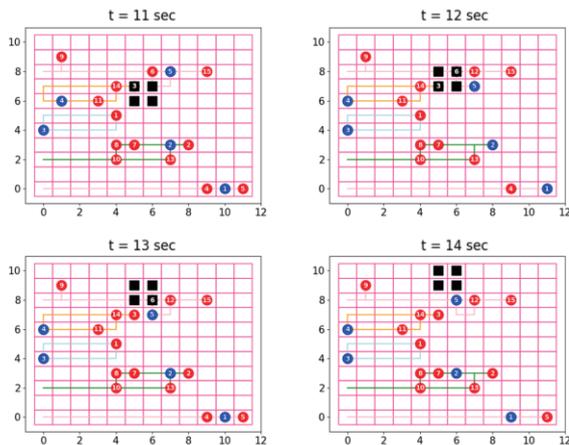


Fig. 13 Example of Dynamic obstacle avoidance

장애물 회피에 대한 방안을 보기 위해 무인 이동체 5대, 임무 위치 15 곳일 때의 상황 중 일부를 Fig. 13과 같이 도시하였다. 1 ~ 4번 무인 이동체는 정해진 대로 임무를 수행하고, 5번 무인 이동체는 1초에 한 칸씩 위로 올라가는 동적 장애물을 피하기 위해 12초일 때 한 칸 아래로 이동하고 13초일 때 한 칸 왼쪽으로

이동한 후 14초일 때 한 칸 위쪽으로 가서 6번 임무를 수행하게 된다. 장애물이 없다면 12초에서 바로 왼쪽으로 이동하면서 6번 임무를 수행할 수 있다. 장애물과의 충돌이 예상되면, 그 부분을 장애물로 설정하여 우회하는 것을 잘 보여준다. 무인 이동체 간의 충돌에도 이와 같이 대응할 수 있다.

## 5. 결 론

본 논문에서는 도심환경에서 다수의 무인 이동체가 주어진 많은 임무를 해결할 수 있는 임무할당 및 경로 계획 기법(GMTPA)을 제안하였다. 이 기법은 빠르게 임무를 할당하고, 충돌 없는 경로를 계산하는 두 단계로 나뉘어서 각 무인 이동체의 경로를 도출해낸다. 그 결과 유전 알고리즘과 비교했을 때에도 계산시간이 적지만 임무 전체 종료 시간이 유사한 경로를 도출할 수 있었다. 작은 규모의 환경에서 본 논문의 기법이 충돌 없는 임무할당 결과를 제시하는 것을 확인하였고, 다양한 환경에서 유전 알고리즘과 성능을 비교했다. 적은 계산시간으로 임무 할당 및 충돌 없는 경로 계획 결과를 도출하는 것을 확인하였다.

## 후 기

이 논문은 2022년도 정부(방위사업청)의 재원으로 국방기술진흥연구소의 지원을 받아 수행된 연구임 (No. KRIT-CT-21-009, 전장정보 기반 실시간 자동임무실행/수정기술 개발)

## References

- [1] Semsch, Eduard, et al. "Autonomous UAV surveillance in complex urban environments." *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. Vol. 2, pp. 82-85, 2009.
- [2] Galkin, Boris, Jacek Kibilda, and Luiz A. DaSilva. "Coverage analysis for low-altitude UAV networks in urban environments." *In GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1- 6, 2017.
- [3] Boonpook, Wuttichai, et al. "UAV-based 3D urban environment monitoring." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. IV-3, pp. 37-43, 2018.
- [4] Lei, Tingjun, et al. "Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots." *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, Issue 6, pp. 5129-5140, 2022.
- [5] Jose, Kelin, and Dilip Kumar Pratihar. "Task allocation and

collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods." *Robotics and Autonomous Systems*, Vol. 80, pp. 34-42, 2016.

- [6] Biswas, Sumana, Sreenatha G. Anavatti, and Matthew A. Garratt. "Nearest neighbour based task allocation with multi-agent path planning in dynamic environments." *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. pp. 181-186, 2017.
- [7] Sharon, Guni, et al. "Conflict-based search for optimal multi-agent pathfinding." *Artificial Intelligence*, Vol. 219, pp. 40-66, 2015.
- [8] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics*, Vol. 4, Issue 2, pp. 100-107, 1968.
- [9] Byeong-Min Jeong, Dae-Sung Jang, Nam-Eung Hwang, Joon-Won Kim, & Han-Lim Choi. "Genetic algorithm based multi-UAV mission planning method considering temporal constraints." *Journal of Aerospace System Engineering*, Vol.17, No.2, pp.78-85, 2023.