

## 論文

J. of The Korean Society for Aeronautical and Space Sciences 42(2), 127-133(2014)

DOI:<http://dx.doi.org/10.5139/JKSAS.2014.42.2.127>

## 대공위협에 대한 무인기 생존성 최대화 경로점 결정기법

박상혁\*, 홍주현\*, 하현종\*, 유창경\*\*, 신원영\*\*\*

Determination of Waypoints to Maximize the Survivability of UAV  
against Anti-air Threats

Sanghyuk Park\*, Ju-Hyeon Hong\*, Hyun-Jong Ha\*, Chang-Kyung Ryoo\*\* and Wonyoung Shin\*\*\*

Department of Aerospace Engineering, Inha University\*,\*\*

Agency for Defense Development\*\*\*

## ABSTRACT

This paper proposes a determination method of waypoints to maximize the survivability of a UAV. Voronoi diagram which is used for the initial selection of waypoint candidates is the most widely used path planning technique to avoid the threat as far as possible when the location and strength of the threat are given. But if threat strength is different each other and flight path is constrained along with straight lines, Voronoi diagram has limitations in real applications. In this study, the initial waypoints obtained from Voronoi diagram are optimized considering the shape of each threat. Here, a waypoint is optimized while adjacent waypoints are fixed. By repeating this localized optimization until whole waypoints are converged, computation time for finding the best waypoints is greatly reduced.

## 초 록

본 논문은 무인기의 생존성을 최대화하기 위한 경로점을 결정하는 기법을 제안한다. 본 논문에서 초기 경로점을 선정하기 위해 사용된 Voronoi diagram은 위협의 위치와 크기가 주어져 있을 때, 위협으로부터 최대한 멀리 피해서 가는 경로를 계획하는데 많이 이용되고 있는 기법이다. 하지만 위협의 크기가 서로 다르고, 비행경로가 직선경로로 계획되어야 한다면 실제 경로 계획 시 Voronoi diagram으로는 한계가 있다. 본 논문에서는 Voronoi diagram을 통해 선정한 초기 경로점을 위협의 형태에 맞춰 최적화를 수행하는 방식을 적용하였다. 각 경로점의 인접한 경로점들을 고정시킨 상태에서 경로점을 하나씩 최적화를 수행하는 방식으로 최적화를 수행하였다. 이 방식은 전체 경로점들이 수렴할 때 까지 국소최적화를 반복함으로써, 최적의 경로점을 찾기 위해 소요되는 연산시간을 크게 감소시켰다.

**Key Words** : Path planning(경로 계획), Voronoi diagram(보로노이 다이어그램), Threat(위협), Optimization(최적화)

† Received: September 25, 2013 Accepted: January 27 2014

\*\* Corresponding author, E-mail : ckryoo@inha.ac.kr

<http://journal.ksas.or.kr/>

pISSN 1225-1348 / eISSN 2287-6871

## I. 서론

순항미사일을 비롯하여 무인정찰기, 무인전투기 등의 무인비행체는 전장에서 멀리 떨어진 안전한 영역에서 출발하여 적의 위험 영역을 통과한 후 목표물을 정찰하거나 타격하는 임무를 갖는다. 따라서 무인비행체의 운용을 위해서는 출발점에서 목표점까지 비행체를 이동시킬 수 있는 경로점의 결정이 이루어져야 한다. 이러한 경로점의 결정은 최종목표 달성을 최우선으로 하는 가장 효율적인 경로를 계획하기 위하여 비행체가 비행을 시작하기 전에 이루어진다.

경로점의 결정기법은 무인비행체뿐만 아니라 로봇분야에서도 많은 연구가 이루어지고 있다. 경로점을 선정하는 방식은 크게 경로점을 직접 선정하는 방식과 경로를 선정하고 선정된 경로위에 경로점을 추가하는 방식으로 나눌 수 있다. 경로점을 직접 선정하는 방식으로는 Voronoi diagram, Visibility graph를 이용하는 방식이 있고, 경로를 선정하고 선정된 경로위에 경로점을 추가하는 방식에는 Potential field, 궤적최적화, Sampling based planning 기법 등을 들 수 있다. 이 중에서도 Voronoi diagram[1,2,3]은 빠른 비행 경로점 산출이 가능하고 공간적 위협요소의 고려가 용이하여, 단순하고 신뢰성 높은 비행경로점 산출 알고리즘 구성에 적합하다. 물론 산출되는 경로가 직선경로로만 이루어져있기에 비행체의 동특성을 반영한 경로계획을 위해서는 경로 수정이 요구된다. 하지만 고속으로 순항하거나 비교적 장시간 비행을 요구하는 순항미사일, 고고도 무인정찰기 등은 대개 그 경로계획이 직선으로 이루어지고 이를 추종하는 방식으로 유도/제어되기에 직선 형태의 경로계획 알고리즘이 더 유용할 수 있다.

본 논문은 적지에 침투하는 무인비행체의 경로계획 상황을 가정하고, 초기 경로점을 선정하기 위하여 Voronoi diagram 기법을 적용하였다. 단, 각 위협이 균일한 크기를 갖는 것이 아니라 서로 다른 크기를 가질 수 있고, 각각의 위협은 지수함수 형태를 갖는다는 전제로 문제를 정의하였다.

위험의 크기가 다른 문제에 대한 경로점 결정 연구도 다수 수행되어 왔다. 위험을 원으로 가정하고 각각의 반경을 달리한 후 위협의 경계들 사이 거리를 등분하는 원집합의 Voronoi diagram[4]이나, 위협 사이의 유클리디안 거리에 가중치가 부여된 임의의 공간 거리를 이용하여 등분하는 Multiplicatively weighted voronoi

diagram[5] 등이 대표적이다. 또한 각 경로점에 가중치를 달리 부여하여 두 경로점 사이의 내분점이 이등분점이 아니도록 하는 Improved Voronoi diagram[6]도 있다. 하지만 이들 방법은 계획된 경로가 곡선 형태로 나타나 직선경로 계획에 적합하지 않거나, 가중치가 거리에 따라 달라지는 문제에 대해서는 적용이 어렵다.

이에 본 논문에서는 Voronoi diagram을 통하여 초기 경로점을 선정하고, 다양한 위협의 형태에 적용하기 위하여 최적화를 적용하였다. 또한 전체 경로점에 대해 최적화를 수행할 시 발생하는 연산시간 문제를 해결하기 위하여 초기 산출된 경로점의 위치를 순차적으로 최적화하는 방법을 제안한다.

## II. 본론

### 2.1 경로계획 문제

본 논문에서는 안전한 지역에서 발사된 무인비행체가 적지에 침투하여 위협을 회피하며 목표점에 도달하기까지의 생존율을 극대화하는 경로를 계획하고자 한다. 이때 각 위협들은 그 위치가 알려져 있고, 각 위협들로 부터의 피격율이 알려져 있는 것으로 가정하였다. 또한 계획되는 경로는 직선으로 이루어져야 한다고 가정하였다.

수식 (1)은 각 직선구간에서 하나의 위협으로부터 받는 피격율( $P_{kill}$ ) 산출식을 나타낸 것이고, Fig. 1은 이를 나타낸 것이다. 위협 하나에 대한 피격율은 각 직선구간에서 가장 높은 값을 사용하였고 가장 가까운 위치에서 피격율이 가장 높은 것으로 나타난다.

$$P_{kill,j} = \exp(-d_{i,j}^2/\sigma_i^2) \quad (1)$$

$d_{i,j}$ 는  $i$ 번째 위협중심으로부터  $j$ 번째 세그먼트에

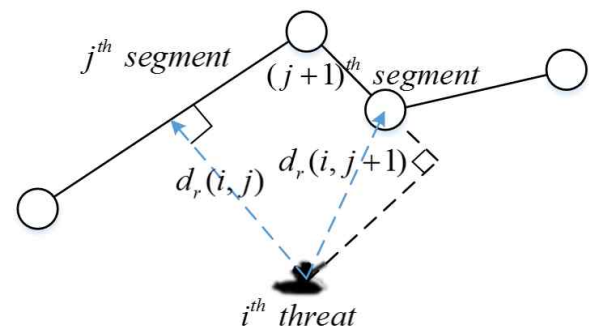


Fig. 1. Kill probability of each segment

수직한 거리이고,  $\sigma_i$ 는  $i$ 번째 위협의 위협 크기를 나타내는 상수로 주어지는 값이다.

Voronoi diagram을 최적화시키기 위해서는 목적함수의 정의가 필요한데 위의 피격율을 이용하여 수식 (2)와 같은 성능평가지수를 산출하고 성능평가지수를 최소화하는 것으로 최적화 문제를 정의하였다.

$$P_{kill} = 1 - \prod_{i=1}^N \prod_{j=1}^M (1 - P_{kill_{i,j}}) \quad (2)$$

$N$ 과  $M$ 은 각각 위협의 개수와 직선구간의 개수를 나타낸다.

## 2.2 최적 경로점 선정기법

### 2.2.1 각 경로점 최적화

Voronoi diagram으로부터 얻어지는 초기 경로점은 각 위협을 동일한 크기의 점으로 보고 얻어진 결과이다. 하지만 본 논문에서 고려하는 위협은 그 크기가 동일하지 않고, 점으로 나타나지 않기 때문에 각 경로점은 최적화를 통해 주변 위협들로부터 피격율을 최소화하는 최적화된 위치로 이동해야 한다.

세 점이 이루는 외접원 내부에는 항상 세 개의 점만 존재하도록 하는 조건을 만족하는 세 점들을 연결한 삼각형을 Delaunay 삼각형이라고 한다. Fig. 2는 단일 Delaunay 삼각형 ( $\Delta W_1 W_2 W_3$ )과 3개의 직선구간 ( $\overline{V_1 V_1}$ ,  $\overline{V_2 V_2}$ ,  $\overline{V_3 V_3}$ ), 4 개의 경로점 ( $V(x, y)$ ,  $V_1, V_2, V_3$ )를 보여준다. 이를 전체 영역으로 확장하면 각 경로점과 직선구간은 각기 다른 Delaunay 삼각형의 경로점과 직선구간이 된다. 우선 하나의 Delaunay 삼각형에 대해서 경로점  $V(x, y)$ 를 최적화 변수로 하여 최적화를 수행한다. 이때 나머지 경로점인  $V_1, V_2, V_3$ 는 그 위치가 고정된다. 세 개의 위치가 고정되기 때문에  $V(x, y)$  최적 위치는

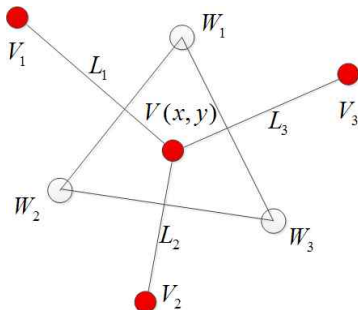


Fig. 2. Single delaunay tri-angle for vertex optimization

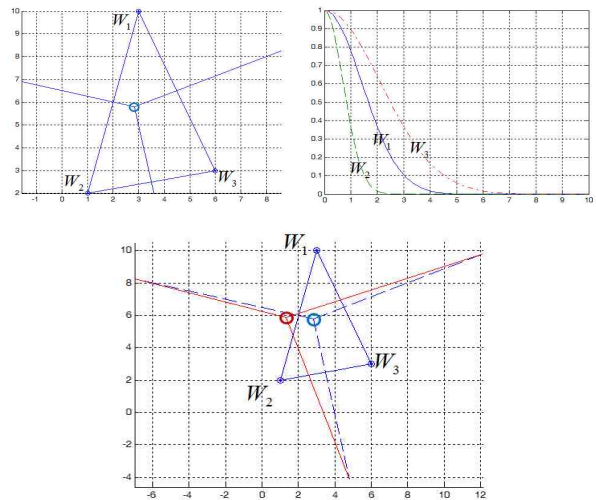


Fig. 3. Result of vertex optimization

비교적 쉽게 찾아진다. Fig. 3은 각 경로점 최적화의 예를 든 그림이다. 최적화전의 경로점인 파랑색점에서 구한 성능평가지수는 0.6056이었는데, 최적화 후에 경로점의 위치가 빨간색점과 같이 이동하면서 성능평가지수가 0.4527로 감소하였다.

### 2.2.2 전체 경로점 순환 최적화

하나의 경로점이 최적화 되면 연속해서 다음 경로점을 최적화하게 된다. Fig. 4는 2개의 Delaunay 삼각형이 그려진 Voronoi diagram에서의 최적화 개념을 나타내고 있다. 먼저 경로점  $V_3$ 에 대한 개별 최적화를 수행하기 위하여 Vertex  $V_3$ 와 연결된 경로점  $V_1, V_2, V_4$ 는 그 위치가 고정된다. 최적화를 통해 앞서 설명한 바와 같이  $V_3$ 가  $V_3^*$ 로 이동하면 다음으로  $V_4$ 가 최적화된다. 마찬가지로  $V_3^*, V_5, V_6$ 는 고정된다. 이때  $V_4$ 가  $V_4^*$ 로 이동하면 직선구간  $L_3$ 의 길이와 기울기가 바뀌기 때문에  $V_3^*$ 가 여전히 최적의 위치라고 판단할 수 없다. 따라서  $V_3^*, V_4^*$ 는 순차적으로 반복하여 최적화가 이루어진다. 이러한 반복은 전체 경로점의 피격율 변화가 일정값 이하로 낮아질 때 까지 계속된다.

본 논문에서는 Voronoi diagram에서 생성되는 모든 경로점에 대하여 이러한 최적화를 반복하는 경우  $V_1, V_2, V_5, V_6$ 와 같이 가장 바깥쪽에 위치하는 경로점은 최적의 직선구간을 따라 그 해가 무수히 많을 수 있으므로 최적화에서 제외하였다. 일반적으로 Voronoi diagram의 외곽 경로점은 중앙에 분포한 위협들로부터 멀리 떨어뜨려 생성시킬 수 있기 때문에 위치 이동을 하더라도 연결된 직선구간의 기울기 변화가 크지 않아 전

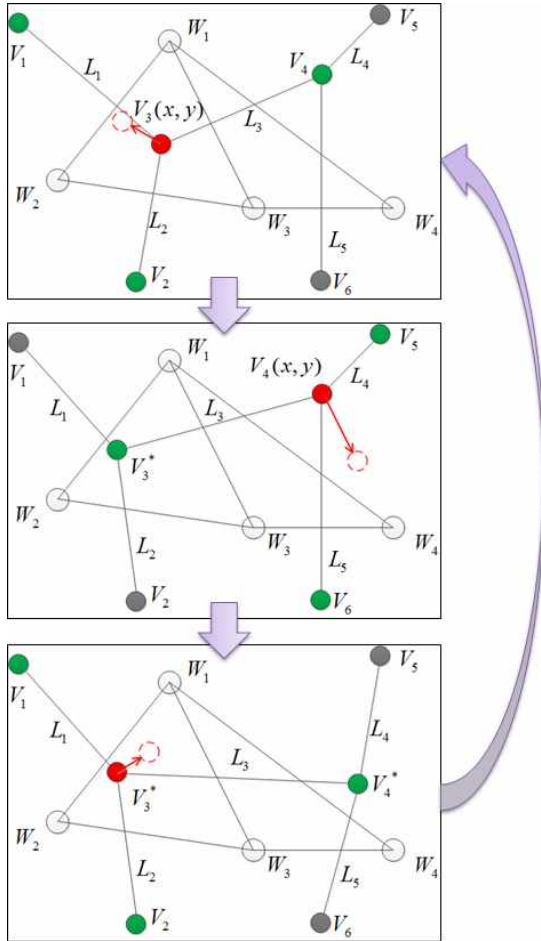


Fig. 4. Concept of Voronoi diagram optimization

체 피격을 변화에 큰 영향을 끼치지 못한다.

2.2.3 경로점 최적화 순서에 따른 결과 검증

경로점의 수가 많을수록 연결은 복잡해지고 최적화를 수행하는 순서에 따라서 수렴결과가 달라질 수 있다. 따라서 다양한 순서에 따른 수렴 결과를 확인하였다. 경로점이 26개 존재하는 경우 26!개의 최적화 순서가 존재한다. 하지만 이 모든 경우에 대해서 검증할 수는 없기 때문에 본 논문에서는 각 경로점을 시작점으로 하여 TSP(Traveling Salesman Problem)[7] 알고리즘을 통해 개별 최적화 순서를 결정하고 26가지 순서에 대해서만 테스트를 수행하였다.

제시한 경로점 순환 최적화는 목적함수( $J_q$ ) Q 개를 각각 최적화하는 문제와 같다고 볼 수 있고, 이는 Multi-object 최적화로 고려할 수 있다.

따라서 전체 경로점의 최적화 목적함수  $J$ 는 각각의  $J_q$ 가 모두 동일한 가중치를 갖는다고 가정할 때, 다음과 같이 정의될 수 있다.

$$J = \sum_{q=1}^Q J_q = \sum_{q=1}^Q \left( 1 - \prod_{i=1}^n \prod_{j=1}^m (1 - P_{kill,q,i,j}) \right) \quad (3)$$

$n$ 과  $m$ 은 각각 하나의 경로점을 최적화 하는데 고려된 위협의 개수와 연결된 직선구간의 개수이다.

최적화 문제 A-I은 순환최적화의 결과와 비교하기 위해 수식 (3)과 같은 목적함수를 갖고 전체 경로점들을 모두 최적화 변수로 설정한 최적화 문제이다. 반면 최적화 문제 A-II는 각각의 경로점을 순차적으로 최적화하는 문제이다. 최적화 문제 A-II에서는 다양한 최적화순서에 따른 결과를 살펴보기 위해 TSP알고리즘을 통해 정해진 26개의 순서에 따라 26세트의 최적화문제를 풀어 내어 연산시간과 성능지수의 평균을 구하여 A-I과 비교하였다.

문제 설정을 위해 먼저 200km x 200km 크기의 영역 안에 위협을 20개 배치하였다. 그리고 각 위협의 크기  $\sigma_i$ 는  $5km \leq \sigma_i \leq 15km$ , ( $i = 1, 2, \dots, 20$ )사이에서 랜덤으로 선택하였다. Fig. 5에서 별 표식은 위협의 위치를 나타내고, 점선은 각 위협을 중심으로 하여 피격율이 40%인 위치를 원으로 나타낸 것이며, 쇠선은 직선구간을 나타낸다. 또한 지도영역의 경계선에 위치한 원은 개별 최적화와 동일한 조건으로 맞추기 위하여 최적화 시 위치가 변경되지 않도록 하였다. 이러한 조건하에서 경로점으로 선택되는 점은 총 26개가 생성되고, 최적화 파라미터의 총 개수는 52개가 된다.

최적화 문제 A-I은 52개의 최적화 파라미터를 수식 (3)을 목적함수로 하여 최적화를 수행하였는데 동시에 26개의 경로점을 흔들기 때문에 순서에 무관한 최적 결과를 얻을 수 있다.

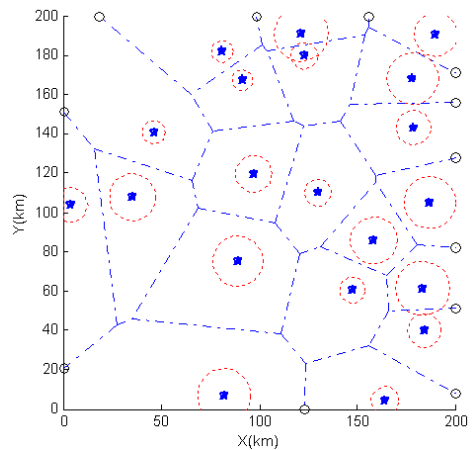


Fig. 5. Threats arrangement and voronoi diagram

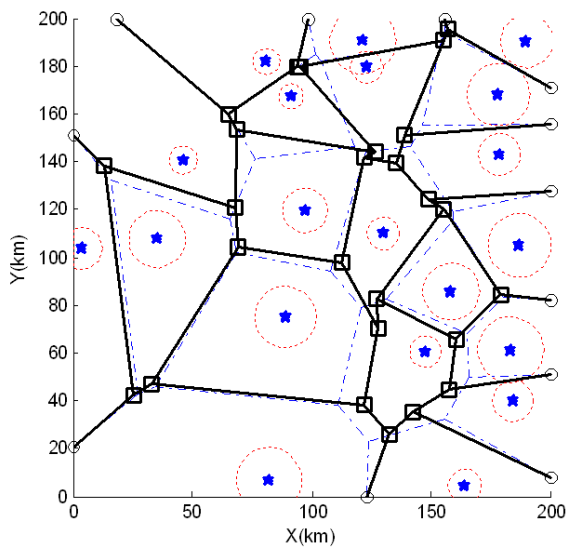


Fig. 6. Optimization result(A-I)

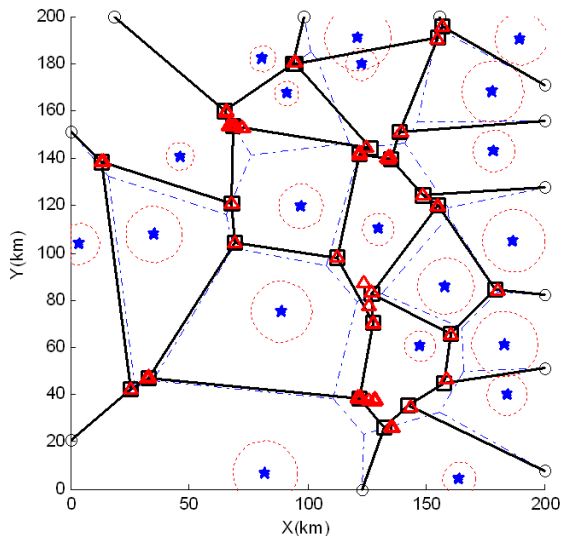


Fig. 7. Optimization result(A-II)

Figure 6은 최적화 문제 A-I의 최적화 결과이다. 이때의 최적 비용은  $J^* = 7.1258$ 이고, 연산시간은 25.8초이다. Fig. 6에서 실선은 최적화된 직선구간을 나타내고, 사각형 표시는 최적화된 경로점을 나타낸다.

Figure 7은 최적화 문제 A-II로 26번의 최적화를 수행한 결과를 하나의 그림에 중첩시킨 것이다. 그림에서 삼각형 표시는 최적화된 경로점의 위치를 나타낸다. 삼각형 표시가 거의 동일한 위치에 표시되는 것을 확인함으로써 A-II 방법으로 순서에 상관없이 거의 동일한 최적화 결과를 얻을 수 있음을 확인하였다. 일부 경로점에서는 위치오차가 크게 나타나는 것으로 보이기도 하지만 확인 결과 피격율이 아주 작은 지역에서는 위

Table 1. Result comparison of two optimization method

	A-I	A-II(average)
cost	7.1258	7.1371
run-time	25.8	4.8

치가 크게 변하더라도 피격율 변화가 작기 때문에 발생하는 문제였고, 이로 인하여 피격율에서 다소 손해를 볼 수는 있지만 무시해도 무방한 수준으로 판단되었다.

다음으로 두 방법으로부터 얻어진 비용과 연산시간을 비교하여 Table 1에 정리하였다. A-II 방법은 26번의 결과를 평균하여 나타낸 것이다. A-II 방법이 비용은 약간 높게 나타났지만, 그에 비하여 연산시간이 1/5수준으로 감소한 것을 확인할 수 있었다.

### 2.2.4 연산시간 비교

앞서 A-II 방법의 연산시간이 A-I에 비하여 적게 소요됨을 앞서 언급하였다. 최적화 문제는 최적화 변수의 개수가 증가함에 따라 그 연산시간이 크게 증가하기 때문에 A-II 방법을 적용하여 연산시간을 감소시킬 수 있는 것은 큰 의미를 갖는다.

두 방법의 연산시간을 비교하기 위하여 다음과 같은 최적화 환경을 구성하였다. 200km x 200km 영역에 위협의 개수는 10개, 15개, 20개, 25개, 30개, 40개 순으로 증가시켰고, 위협 크기의 표준편차를  $5km \leq \sigma_i \leq 15km$ 의 범위를 갖도록 설정하였다. 위협의 수가 증가하면 Voronoi diagram이 복잡해지면서 초기 경로점의 수도 증

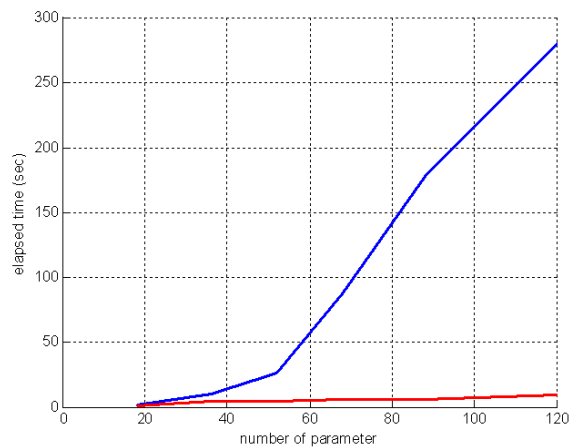


Fig. 8. Run-time comparison between A-I and A-II

가하는데 이는 곧 최적화 변수의 증가를 의미한다. 여기서 최적화 연산시간은 위협의 개수뿐만 아니라 위협 범위의 크기와 같은 문제 설정 등에 의해서도 영향을 받으므로 절대적인 값은 아니다. 단지 두 방법을 비교하여 연산시간의 단축 효과에 대해서 확인하고자 하였다.

Figure 8에서 실선은 A-I 방법을 적용했을 때의 연산시간을 나타내고 점선은 A-II 방법을 적용했을 때의 연산시간을 나타낸다. 이를 통해 A-I 방법은 최적화 변수의 개수가 증가하면 연산시간이 큰 기울기로 증가하지만 A-II 방법은 아주 작은 기울기로 증가하는 것을 확인할 수 있다.

### 2.2.5 경로계획 예

400km x 400km 영역에 100개의 위협을 분포시킨 후, 각 위협 크기의 표준편차를  $5km \leq \sigma_i \leq 25km$ 로 랜덤하게 할당하였다. 시작점은 [0km, 23km]이고, 목표점은 [177km, 364km]로 설정하였다. 또한 생성된 Voronoi diagram에서 시작점과 목표점을 잇는 최단경로를 찾는 알고리즘은 최단경로탐색 문제에서 많이 사용되는 Dijkstra algorithm[8]을 사용하였다.

Figure 9에서 굵은 점선은 최적화하기 이전의 Voronoi diagram으로부터 구한 초기 경로점을 통해 생존성이 최대인 경로를 찾은 결과이고, 굵은 실선은 Voronoi diagram으로부터 구한 초기 경로점을 최적화하여 나온 경로점을 이용하여 구한 비행경로이다. 최적화 이후의 전체 성능평가지수는 0.9005로 최적화 전의 0.9825보다 낮은 값으로 나타났다.

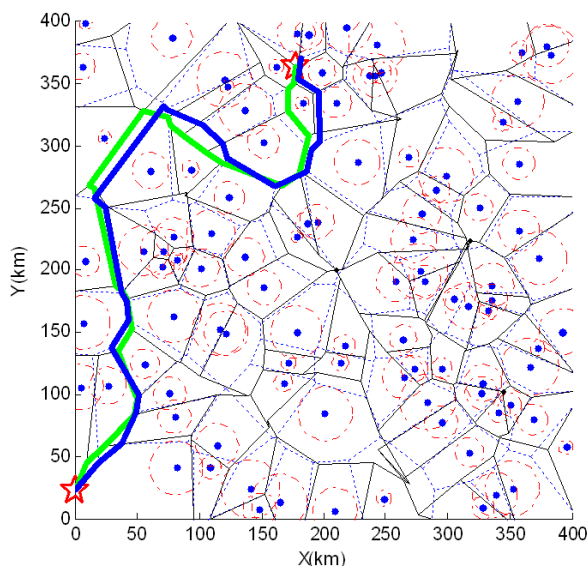


Fig. 9. Path planning example

특히 두 경로를 비교해보면 단순히 위협의 크기가 작은 위협 쪽으로 경로가 평행이동 하는 것뿐만 아니라 아예 다른 길로 돌아가는 구간도 발견되는데, 이는 초기에 위협의 크기가 제대로 반영되지 않아 완전히 잘못된 비행경로를 산출하는 경우이다. 한편, 두 경우에서 모두 성능평가지수가 0.9 이상으로 높게 나타나는 것은 비행경로의 성능평가지수로 수식 (2)를 적용했기 때문이다. 수식 (2)를 적용하는 경우 피격율이 낮은 구간도 그 수가 많아질수록 성능평가지수가 높아진다. 실제로 각 구간별로 피격율을 살펴보면 가장 큰 값이 0.4651로 나타난다.

## III. 결 론

본 논문에서는 무인비행체의 생존성을 최대화하기 위한 경로점 결정기법을 제안하였다. 위협을 회피하는 비행경로를 산출하기 위하여 Voronoi diagram을 사용하였다. 이때 위협 모델을 크기가 각기 다른 지수함수의 형태로 가정하였는데, 일반적인 Voronoi diagram으로 구한 초기 경로점을 사용할 경우 정확한 피격율을 반영한 비행 경로점의 산출이 어렵다. 따라서 Voronoi diagram으로 구한 초기 경로점을 최적화하여 정확한 피격율이 반영된 비행 경로점을 산출하였다.

또한 일반적인 최적화 알고리즘으로는 비행영역이 넓어지고, 위협의 수가 많아질수록 최적화 변수의 수가 크게 증가하여 연산시간 역시 크게 증가한다. 연산시간의 증가는 최적화 알고리즘을 실시간 경로계획에 사용하기 어렵게 만든다. 본 논문에서는 이러한 문제를 해결하기 위하여 경로점을 개별적으로 최적화하는 방안을 제시하였다. 제안한 방안으로 얻은 최적화 결과를 통해 성능지수에서 다소 손해를 보더라도 연산시간에서 크게 이득을 볼 수 있음을 확인하였다.

## 후 기

본 연구는 국방과학연구소 및 인하대학교의 지원으로 수행되었으며, 이에 대해 깊이 감사드립니다.

## References

- 1) Novy, M. C., Jacques, D. R. and Pachter, M., "Air vehicle optimal trajectories between

two radars," *American Control Conference*, Vol. 1, 2002, pp.785~790.

2) Hammouri, O. M. and Matalgah, M. M., "Voronoi path planning technique for recovering communication in UAVs," *Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications*, 2008, pp.403~406.

3) Judd, K. B. and McLain, T. W., "Spline based path planning for unmanned air vehicles," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Montreal, Canada, 2001.

4) Lim, K., Park, S. and Shin, H., "Dynamic Construction of the Voronoi Diagram for the Circle set," *Proceedings of the Society of*

*CAD/CAM Conference*, 2007.

5) Mu, L., "Polygon characterization with the multiplicatively weighted Voronoi diagram," *The Professional Geographer*, Vol. 56, No. 2, 2004, pp.223~239.

6) Xiao, Q., Gao, X., Fu, X. and Wang, H., "New local path replanning algorithm for unmanned combat air vehicle," *Proceedings of the 6th World Congress on Intelligent Control and Automation*, 2006, pp.4033~4037.

7) Gutin, G. and Punnen, A. P., *The traveling salesman problem and its variations*, Springer, 2007.

8) Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., *Introduction to Algorithms*, 2nd McGraw-Hill Book, 2001.