

무기-표적 할당 문제에 대한 메타휴리스틱의 성능 비교

최용호^{*1)} · 이영훈¹⁾ · 김지은²⁾

¹⁾ 연세대학교 산업공학과

²⁾ 국방과학연구소 제1기술연구본부

Comparative Study on Performance of Metaheuristics for Weapon-Target Assignment Problem

Yong Ho Choi^{*1)} · Young Hoon Lee¹⁾ · Ji Eun Kim²⁾

¹⁾ Department of Industrial Engineering, Yonsei University, Korea

²⁾ The 1st Research and Development Institute, Agency for Defense Development, Korea

(Received 11 November 2016 / Revised 20 February 2017 / Accepted 28 April 2017)

ABSTRACT

In this paper, a new type of weapon-target assignment(WTA) problem has been suggested that reflects realistic constraints for sharing target with other weapons and shooting double rapid fire. To utilize in rapidly changing actual battle field, the computation time is of great importance. Several metaheuristic methods such as Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony Optimization, and Particle Swarm Optimization have been applied to the real-time WTA in order to find a near optimal solution. A case study with a large number of targets in consideration of the practical cases has been analyzed by the objective value of each algorithm.

Key Words : Optimization(최적화), Weapon-Target Assignment Problem(무기-표적 할당 문제), Metaheuristic Algorithms (메타휴리스틱 알고리즘)

1. 서론

공중 위협이란 적이 공중 공간을 이용하는 능력과 침투기도 및 기타 요소 등에 의해 아군이 받는 심리적 긴장상태로 항공기(유인 및 무인항공기)와 미사일, 로켓, 포탄 등의 기타 위협으로 구분된다. 특히, 전술

탄도 미사일은 주요한 공중 위협으로 이에 대응하는 무기 체계는 공중 우세와 제공권 장악을 위한 공군의 방공 작전의 핵심적인 요소이다. 우리나라는 중심이 짧은 지형 여건으로 인하여 조기경보시간의 제약이 있어 전쟁 초기에 공중 위협으로부터 성공적으로 방어하지 못하면 전쟁의 승리를 보장할 수 없다.

무기-표적 할당(Weapon-Target Assignment, WTA) 문제란 다수의 표적에 가용 무기 자원을 적절히 할당함으로써 표적의 생존 가치를 낮추는 문제이다. 실시간

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

으로 접근하는 공중 위협에 대하여 아군의 방공 무기를 어떻게 할당할 것인지를 결정하는 문제 또한 대표적인 무기-표적 할당 문제 중 하나이다.

조합최적화 문제로써 무기-표적 할당의 최적해를 찾는 문제는 1986년 Lloyd & Hans^[1]에 의해 NP-Complete 임이 증명되었다. 이러한 NP-Complete 문제들은 계산 시간을 줄이기 위하여 주로 휴리스틱 알고리즘을 이용하여 근사해를 찾는다. Xin et al.(2010)^[2]은 타부 서치(Tabu Search, TS)를 이용하여 해를 찾는 알고리즘을 제시하였으며, 현재 해에서 이웃 해를 생성한 후 해당 이웃해가 예전에 생성한 해와 동일하거나 일부분이 동일할 경우 해당 해로 이동하지 못하고 다른 이웃해로 이동하도록 유도하여 효율성을 높였다. Lee et al.(2002a)^[3]는 개미 군집 최적화(Ant Colony Optimization, ACO)를 발전시켜 표적 할당 문제에 해답을 제공하는 모형을 제시하였다. 이동언(2011)^[4]은 무기 할당 문제에서 다양화 전략과 강화 전략을 혼합한 개선된 ACO 알고리즘을 제안하였다. 홍성삼(2012)^[5]은 공중에서 위협하는 표적에 대해 아군 무기를 적당히 할당하는 문제를 전형적인 NP-Complete 문제로 보고 이를 해결하기 위해 ACO, 개체 군집 최적화(Particle Swarm Optimization, PSO), 유전자 알고리즘(Genetic Algorithm, GA)의 3 가지 휴리스틱 알고리즘을 비교하였으며 시나리오 전반에서 ACO의 성능이 좋은 것을 확인하였다. Lee & Lee(2003)^[6] 및 Lee & Lee(2005)^[7]는 GA와 ACO를 혼합 적용한 메타휴리스틱 알고리즘을 제시하였다. Lee et al.(2002b)^[8]은 단일 표적 할당 문제에 도메인 지식을 활용한 GA 알고리즘을 제시하였다. Lee et al.(2003)^[9]는 무기 할당 문제를 풀기 위해 탐욕적 우생학(Greedy Eugenics)을 결합한 새로운 GA를 제안하였다. 그는 최적화된 자손(Offspring)으로 개선시키는 계획을 도입함으로써 GA의 성능을 향상시켰다. Julstrom(2009)^[10]은 무기할당 문제에서 각각의 적 즉, 목표물과 파괴 효과, 그리고 파괴 효과의 가치를 설정하여 이를 수식으로 표현하고 기대되는 목표물의 가치를 최소화하고자 하였다. Zeng et al.(2006)^[11]은 PSO의 원리를 또 다른 방식으로 발전시켜, 이산형 개체 군집 최적화(Discrete PSO)를 제시하였다. 연속형 변수를 다루는 개체 군집 최적화 알고리즘은 개체의 새로운 위치를 벡터의 합을 통해 나타내는 반면, 이산형 개체 군집 최적화는 개체의 새로운 위치를 탐색할 때, 유전자 알고리즘에서 사용하는 교배의 원리를 이용하여 개체의 일부분을 개체 자신(혹은 집단 전체)의

우수해의 성분으로 치환한다. 이를 바탕으로 다양한 자원 할당 문제에 적용이 가능하도록 하였다.

이와 같이 무기-표적 할당 문제를 효과적으로 풀기 위하여 다양한 휴리스틱 알고리즘을 적용하려는 노력은 상당히 오랫동안 있어왔다. 하지만 대부분의 연구는 충분한 계산 시간을 허용하여 알고리즘의 성능을 비교하였으며, 이는 앞서 언급한 실시간 무기-표적 할당을 요구하는 전쟁초기의 공중위협 대응에 적절하지 않다. 따라서 본 연구는 전쟁 초기의 다수의 공중위협에 대하여 다수의 방공무기를 사용하여 방어하는 상황에서 무기-표적 할당에 적합한 휴리스틱 알고리즘을 찾기 위하여, 기존 연구의 주류를 이루는 다섯 가지의 메타휴리스틱 알고리즘을 구현하고 제한된 짧은 시간 이내에 각 알고리즘의 성능을 비교하였다.

2. 문제 정의

무기-표적 할당의 초창기 모형은 모든 무기가 모든 표적을 공격 가능한 것으로 간주하는데, 실제 전장에서는 무기의 사거리 제약으로 인해 공격하지 못하는 표적이 존재할 수 있다. Hosein Patrick A.(1990)^[12]는 정적 무기-표적 할당(Static Weapon-Target Assignment, SWTA)의 특수한 경우로써 “사거리가 존재하는 무기(Weapons with Limited Target Coverage) 상황”을 제시하였다. 만약 이러한 상황 하에서 무기 i 가 표적 j 를 공격할 때의 파괴 확률 p_{ij} 가 오직 표적의 종류에만 영향을 받는다면 이 문제는 선형 최소 비용 네트워크 흐름 문제로 변환할 수 있게 된다. 즉, 무기의 종류가 하나이면 다항 시간(Polynomial Time) 이내에 풀 수 있는 알고리즘이 존재하지만, 무기의 종류가 여러 가지 라면 사거리가 존재하는 무기 상황이라도 문제의 난이도가 NP-Complete이다. 그런데, 실제 전장 환경에서는 단일 무기로 동종의 표적을 공격하더라도 표적의 궤적, 진행 속도, 무기와의 각도에 따라 파괴 확률이 달라지게 된다. 본 연구에서는 표적의 실시간 정보를 반영하여 매 초 표적 할당을 수행하는 상황을 상정하였으므로 표적(적 공중 위협)이 한 종류이더라도 각 표적마다 파괴 확률이 달라 수리 모형은 NP-Complete의 난이도를 갖는다. 사거리가 존재하는 여러 종류의 무기를 표적에 할당하는 일반적인 수리 모형은 다음과 같다.

- 인덱스

$i \in I$: 방공 무기(포대)

$j \in J$: 표적

- 입력 데이터

v_j : 표적 j 의 위협도

p_{ij} : 무기 i 가 표적 j 를 파괴할 확률

w_i : 무기 i 의 용량

- 결정 변수

x_{ij} : 무기 i 가 표적 j 를 사격하는 횟수

- 목적함수

$$\min f(x) = \sum_j v_j \prod_i (1 - p_{ij})^{x_{ij}} \quad (1)$$

- 제약식

$$\sum_j x_{ij} \leq w_i, \forall i \quad (2)$$

$$x_{ij} \text{ is integer}, \forall i, j \quad (3)$$

목적함수 (1)은 모든 표적의 생존 가치의 합을 최소화한다. 제약 (2)는 각 무기는 용량 이하의 횟수만큼 표적을 사격할 수 있음을 나타내며, 제약 (3)은 결정 변수 x_{ij} 의 정수 조건이다.

무기 i 가 표적 j 에게 한 발을 추가로 사격할 때, 즉 x_{ij} 를 1 만큼 증가시킬 때 목적함수가 감소하는 양을 “한계 파괴 가치(Marginal Destructive Value)”라 하자. 한계 파괴 가치 $m(x_{pq})$ 에 대한 식을 정리하면 다음과 같은 식을 얻을 수 있다.

$$m(x_{pq}) = v_q \left[\prod_{i=1}^I (1 - p_{iq})^{x_{iq}} \right] \times p_{pq} \quad (4)$$

식 (4)에서 $v_q \left[\prod_{i=1}^I (1 - p_{iq})^{x_{iq}} \right]$ 는 표적 q 에 이미 할당된 무기에 따른 표적 q 의 “이전 생존 가치”로 생각할 수 있다. 따라서 x_{ij} 의 한계 파괴 가치는 표적 j 의 이전 생존 가치에 파괴 확률 p_{ij} 를 곱한 값과 같다.

$sur(j) = v_j \left[\prod_{i=1}^I (1 - p_{ij})^{x_{ij}} \right]$ 라 하면 잔여 무기 용량을

할당할 때 $sur(j) \times p_{ij}$ 값이 높은 표적에 할당하는 것이 목적함수를 더 많이 낮출 수 있다. 무기의 종류와 주변 상황에 따라 p_{ij} 값이 달라질 수 있지만, 현대전에서는 대공 포대가 유효 사거리 내의 표적을 타격할 경우 파괴 확률의 편차가 크지 않다. 하지만 $sur(j)$ 값은 v_j 와 $(1 - p_{ij})^{x_{ij}}$ 의 곱의 형태로 이루어지므로 편차가 매우 크게 나타난다. 본 연구에서는 v_j 의 최대값이 최소값의 3 배가 되도록 하고, p_{ij} 는 0.8~0.95 사이의 적절한 값을 갖도록 설정하였다. (데이터에 대한 자세한 설명은 4.실험 설계 파트에서 소개한다.) 이에 따라 표적이 1 회 공격받을 시 생존 가치는 처음보다 5~20 배 감소하고, 2회 공격받을 시는 처음보다 25~400 배 감소하게 된다. 즉, $sur(j)$ 값은 해당 표적이 공격받은 횟수($\sum_i x_{ij}$)에 의해 크게 좌우된다.

결론적으로, 한계 파괴 가치를 높이기 위해서는 공격받은 횟수가 작은 표적, 즉 $\sum_i x_{ij}$ 값이 작은 표적에 무기를 할당해야 하며, 이것은 한정된 무기 용량을 특정 표적에 집중하여 할당하는 것보다 다수의 표적에 고르게 할당하는 것이 전체 목적함수를 낮출 수 있음을 뜻한다. 본 연구에서는 이를 반영하여 하나의 표적에 너무 많은 공격이 집중되는 것을 방지하고자 표적의 위협도에 따라 최대 사격 횟수를 1 발과 2 발로 구분하였다.

전쟁 초기 공중 위협은 크게 탄도 미사일과 항공기로 나눌 수 있다. 탄도 미사일은 아군 방어자산까지의 도달 시간이 비교적 짧고 속도가 빠르지만 궤도가 정해져 있어 예측 사격이 용이하다. 반대로 항공기는 탄도 미사일에 비해 속도는 느리나 경로 예측이 쉽지 않다. 시간을 고려하지 않는 정적 무기-표적 할당 문제에서는 두 종류의 표적을 동일하게 취급하지만, 실제 상황에서는 탄도 미사일은 피격 시 아군 피해가 크므로 예측 사격을 통해 이중, 삼중으로 요격을 시도하고, 항공기는 비교적 속도가 느리고 회피 가능성이 크므로 슛룩슛(Shoot-Look-Shoot) 대응이나 여러 포대의 협공을 시도한다. 따라서 본 연구에서는 표적의 사격 요구 정보를 단발 사격(1회 사격), 연발 사격(동일 포대에서 2회 사격), 그리고 공유 사격(다른 포대에서 각 1회씩 사격)으로 구분하였고, 이는 아래 Fig. 1과 같다.

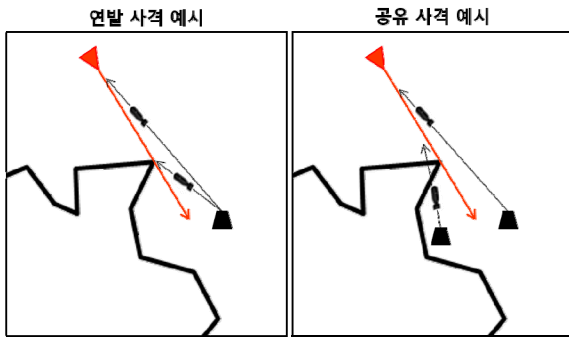


Fig. 1. Example of shoot requirement

연발과 공유 사격을 수리 모형으로 표현하기 위하여 식 (5)와 (6)을 추가하였다.

$$\sum_i y_{ij} \leq s_j, \forall j \quad (5)$$

$$x_{ij} \leq r_j y_{ij}, \forall i, j \quad (6)$$

여기서 y_{ij} 는 무기 i 가 표적 j 를 사격하는 지 여부를 나타내는 이진 결정변수이다. s_j 는 표적 j 를 동시에 공격할 수 있는 무기의 수를 나타낸다. r_j 는 표적 j 의 연발 사격 요구 횟수를 나타낸다.

두 제약을 포함한 수리 모형도 파괴 확률 p_{ij} 가 여전히 무기 i 와 표적 j 에 의존하므로 이 문제는 여전히 NP-Complete이고, 대규모 무기-표적 할당(표적 100 개 이상 수준)에서 Exact Algorithm의 풀이 시간이 매우 많이 걸리므로 메타휴리스틱 알고리즘을 통해 이 문제를 해결하고자 한다.

3. 메타휴리스틱 알고리즘

휴리스틱 알고리즘은 일반적으로 “얼마나 빨리” “최적해에 근접”하는가를 성능 지표로 삼는다. 따라서, 한 사이클의 계산 과정이 단순할수록, 사이클 당 목적함수 개선이 클수록, 최종적으로 수렴한 해의 목적함수가 최적해에 가까울수록 좋은 휴리스틱 알고리즘이라고 할 수 있다.

본 논문에서는 무기-표적 할당 문제를 효율적으로 풀기 위하여 시뮬레이티드 어닐링(Simulated Annealing, SA), TS, PSO, ACO, GA의 5 가지 메타휴리스틱 알고

리즘을 구현하였다. 일반적으로 SA, TS와 같은 단일해 기반 메타휴리스틱은 지역탐색 능력이 강하고, ACO, GA, PSO와 같은 집단해 기반 메타휴리스틱은 전역탐색 능력이 강하다. SA는 구현 논리가 간단하여 시간 대비 탐색 횟수에서 다른 알고리즘에 비해 강점이 있다. TS는 동일한 해의 재탐색을 방지함으로써 다른 단일해 기반 메타휴리스틱보다 지역 최적해를 잘 빠져나올 수 있다. ACO는 각 개체가 그리디 알고리즘에 따라 움직이므로 다른 집단해 기반 메타휴리스틱에 비해 각 개체의 지역탐색 능력이 강하다. GA는 교배 시행을 통하여 세대가 바뀔 때 따라 매우 다양한 해를 생성·탐색할 수 있다. Zeng et al.(2006)^[11]이 제시한 DPSO는 해의 다양성을 통제할 수 있어 GA보다 정교한 지역탐색이 가능하다. 본 연구에서 제시한 새로운 무기-표적 할당 문제의 효율적인 해 탐색을 위하여 모든 메타휴리스틱의 해 표현을 아래 Fig. 2와 같이 설정하였다.



Fig. 2. Example of common format of solutions

연발(혹은 공유) 사격을 요구하는 표적 (c), (d)와 단발 사격을 요구하는 표적 (a), (b)가 존재한다면, 총 사격 횟수는 6 회가 되고 해 표현으로 6 개의 셀이 생성된다. 칠해진 6 번째 셀은 6 번 사격인 동시에 표적 (d)의 두 번째 사격을 의미하며, 각 셀은 위치에 따라 고유의 사격 번호가 정해진다. 셀 안에는 해당 번호의 표적을 공격하는 무기의 번호가 할당되며, 만약 무기를 할당하지 않으면 0을 입력한다.

m 개의 무기와 n 개의 표적에 대해 수리 모형에서는 $m \times n$ 개의 $[0, m]$ 범위의 정수 변수와 이진 변수를 결정해야 하지만, 위의 해 표현을 이용하면 $2n$ 개 이하의 $[0, m]$ 범위의 정수 변수만을 결정하

면 된다.

계산 시간이 길지 않으면서 효과적인 초기 해를 찾기 위해 그리디 탐색(Greedy Search) 알고리즘을 사용하였다.

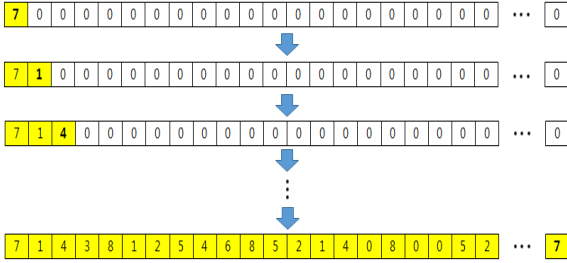


Fig. 3. The process of greedy search

Fig. 3의 순서와 같이, 모든 성분이 0인 초기 해를 생성하고 좌측 셀부터 차례대로 사거리 제약을 만족하는 무기 중 목적함수 값을 가장 크게 낮출 수 있는 무기 번호를 할당한다. 해의 우측 마지막 셀까지 무기 번호를 할당하면 탐욕 탐색 알고리즘을 종료한다.

3.1 시뮬레이티드 어닐링

SA는 현재 해보다 더 높은 비용 함수 값을 갖는 열등한 해도 적당한 확률로 받아들일 가능성을 열어줌으로써 지역 최적에서 빠져나올 수 있게 하여 전체 최적을 찾아나갈 수 있게 한 알고리즘이다.

본 연구의 SA 알고리즘의 한 번의 탐색 시행은 두 단계로 이루어져 있다. 첫 번째 단계는 임의의 한 셀을 선정하여 할당된 무기를 제거하는 것이다. 그리디 탐색이 무기 용량을 최대한 사용하는 방향으로 해를 탐색하기 때문에 초기 해는 각 무기가 가진 용량을 모두 소진하였을 가능성이 크다. 따라서 특정 셀에 할당된 무기를 목적함수 값을 더 낮출 수 있는 다른 무기로 바꾸는 것이 쉽지 않다. 이를 보완하기 위하여 첫 번째 단계에서 하나의 셀의 성분을 0으로 바꿔줌으로써 무기 용량을 확보한다.

두 번째 단계는 임의의 다른 한 셀을 선정하여 목적함수 값을 낮출 수 있는 무기 번호로 바꾸는 것이다. 첫 번째 단계를 통하여 용량이 남는 무기가 최소 하나 이상 확보되므로 첫 번째 단계와 달리 다수의 선택지 중 더 좋은 것을 선택할 수 있다.

열등한 해를 수용할 확률은 아래 식 (7)와 같다.

$$e^{-\frac{\delta_E}{T}} ; (\delta_E = f_{new} - f_{old}) \quad (7)$$

여기서 T 는 현재 온도, f_{new} 는 이웃 해의 목적함수 값, f_{old} 는 기존 해의 목적함수 값, e 는 자연상수이다. 본 연구에서는 식 (8)과 같이 현재 온도를 알고리즘의 시행 횟수에 따라 일정 비율로 감소하도록 설계함으로써 수렴하는 정도를 조절할 수 있게 하였다.

$$T_{n+1} = R \times T_n \quad (8)$$

여기서 T_n 은 n 번째 시행의 온도, R 은 냉각 비율이며, 초기 온도 T_0 와 R 이 SA의 파라미터가 된다.

3.2 타부 서치

TS는 인간의 기억 능력에 착안하여 만들어진 알고리즘이다. 이미 탐색한 공간의 재탐색을 방지함으로써 효율적으로 미지의 해 공간을 탐색한다.

TS는 타부 리스트를 통해 탐색 영역을 제한함으로써 효율적인 탐색을 도모한다. 본 연구에서는 타부 리스트에 k 개의 타부 정보를 저장하며, 하나의 타부 정보는 셀 번호와 무기 번호의 짝으로 구성된다. 만약 타부 리스트에 (6,2)라는 타부 정보가 있다면, Fig. 4와 같이 여섯 번째 셀에 무기 2를 할당하는 모든 해를 탐색하지 않는다.

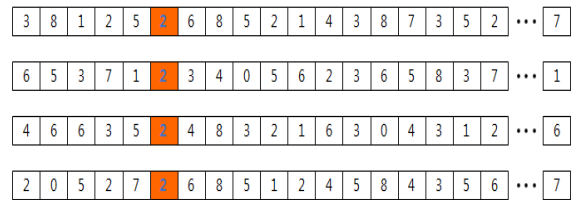


Fig. 4. List of restricted solutions by tabu information of (6,2)

한 번의 이웃 해 탐색 시행에서 총 2 개의 셀의 성분이 바뀌므로 매 시행마다 가장 오래된 2 개의 타부 정보가 급변 시행의 정보로 대체된다.

무기가 8 개, 표적이 요구하는 사격 횟수의 총 합이 140 발인 문제에서 타부 리스트 크기(k)에 따른 탐색 가능 해 개수와 탐색 가능 영역의 비율은 Table 1과 같다.

Table 1. Searching area by tabu list size

타부 리스트 크기	탐색 가능 해 개수	탐색 가능 영역 비율(%)
0	$8^{140} \approx 10^{42}$	100
1	$7^1 \times 8^{139} \approx 8.75 \times 10^{41}$	87.5
2	$7^2 \times 8^{138} \approx 7.66 \times 10^{41}$	76.6
3	$7^3 \times 8^{137} \approx 6.70 \times 10^{41}$	67.0
10	2.63×10^{41}	26.3
20	6.92×10^{40}	6.9
40	4.79×10^{39}	0.479
100	1.59×10^{36}	0.00015
>980	0	0

3.3 개체 군집 최적화

집단해 기반 메타휴리스틱의 초기 해를 설정하기 위하여 “임의 탐색 알고리즘”을 개발하였다. 집단해 기반 메타휴리스틱 알고리즘의 경우 다수의 해를 놓고 서로 간의 정보를 공유하며 탐색하므로 초기 해를 해 공간에 적절히 퍼뜨리는 것이 중요하다. 따라서 임의 탐색 알고리즘에서는 모든 셀에 할당 가능한 무기 중 임의의 무기를 할당함으로써 해의 다양성을 확보한다.

본 연구의 PSO는 Zeng et al.(2006)^[10]이 제시한 DPSO의 원리를 활용하였다. 이웃 해를 생성할 때, 현재 해와 우수해를 혼합하는데, 우수해는 개체 우수해를 p 의 확률로, 전체 우수해를 $1-p$ 의 확률로 선택한다. 해를 혼합하는 방법은 Fig. 5과 같다.

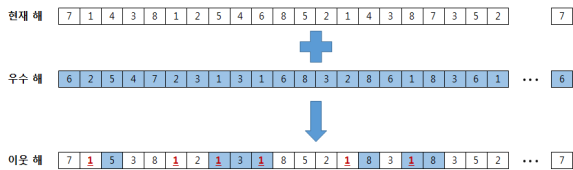


Fig. 5. The process of searching neighbor solution though mixing current and best solutions

현재 해에서 치환 비율 α_{ps0} 만큼 개수의 셀을 우수해의 성분으로 치환한다. 예를 들어, 전체 셀의 길이가 100 개이고 $\alpha_{ps0} = 0.3$ 이라면 30 개 셀의 성분은

우수해의 성분으로 치환하는 것이다. 단, 이러한 시행 후 생성되는 이웃 해는 Fig. 6의 경우처럼 특정 무기(무기 1)가 용량 이상의 표적을 할당받는 등 실행 불가능할 가능성이 크다. 하지만 이러한 시행에 의해 생성된 이웃 해가 비록 실행 불가능할지라도, 섞이기 전의 현재 해와 우수해 각각은 모두 실행 가능한 해이다. 즉, 생성된 이웃 해의 성분을 모두 바꿀 필요 없이 한쪽으로부터 온 성분을 수정하면 된다. 본 연구에서는 우수해의 성분을 보존하면서 현재 해의 성분을 수정하여 해가 실행 가능하도록 만들었다. 그에 따른 수정 절차는 Fig. 6과 같다.

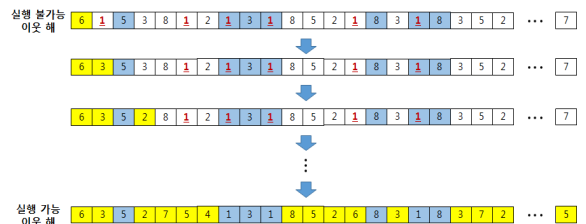


Fig. 6. The correction of infeasible solution created by mixing operation

실행 불가능한 이웃 해의 성분 중에서 현재 해로부터 비롯된 성분을 그리디 탐색의 원리를 적용하여 왼쪽 셀부터 차례대로 바꿔준다. Fig. 6의 예시에서 흰색 셀은 현재 해로부터 비롯된 성분, 파란색 셀은 우수해로부터 비롯된 성분이다. 보정 전의 실행 불가능 이웃 해는 무기 1이 너무 많이 할당되어 용량을 초과하는데, 보정 후엔 2 번째, 6 번째, 14 번째 셀이 1이 아닌 다른 성분으로 바뀐 것을 볼 수 있다. 또한, 이러한 과정을 통해 5 번째, 7 번째, 20 번째 셀처럼 실행 가능성에 영향을 주지 않는 성분도 목적함수 값을 더 낮출 수 있는 성분으로 대체하여 훨씬 우수한 해를 얻을 수 있다.

3.4 개미 군집 최적화

본 연구에서는 수평으로 정렬된 해 표현에 대하여, 각 셀에 할당 가능한 무기 목록을 수직으로 전개한 2차원 행렬을 2차원 평면의 각 지점으로 설정하였다. 목적함수 값을 거리 함수로 지정하면 무기-표적 할당 문제를 좌측 끝에서 우측 끝으로 가는 최단 경로를 찾는 문제로 바꿔 생각할 수 있다. 개미가 2차원 평면에서 움직이는 모습을 Fig. 7과 같이 표현할 수 있다.

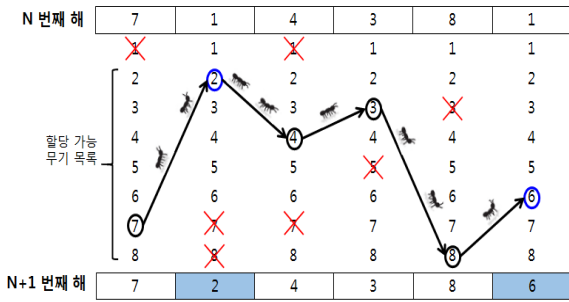


Fig. 7. The movement of ants in 2-dimension matrix

ACO에서 개미가 이동할 때 기준이 되는 값을 세 가지로 설정하였다. 목적함수 값은 특정 무기 번호가 할당되었을 때 그 셀에 해당하는 표적의 생존 가치이다. 페로몬은 개미가 이동하면서 남기는 흔적으로써, 이전 개미의 경로(기존 해) 정보를 담고 있으며, 페로몬이 높을수록 과거에 많은 개미들이 해당 무기 번호를 선택했음을 의미한다. 합산 점수는 목적함수 값의 내림차순 랭크와 페로몬의 오름차순 랭크를 합한 값으로써, 점수가 높을수록 목적함수 값은 낮고 페로몬은 높음을 의미한다.

그런데, 목적함수 값만을 기준으로 개미가 탐색하는 것은 그리디 탐색과 동일하며, 모든 개미가 똑같은 탐색을 하게 된다. 여기에 임의성(Randomness)을 부여하기 위하여 개미가 셀을 이동할 때, 임의 탐색 확률 α_{acc} 로 할당 가능한 무기 중 임의의 무기를 할당하도록 하였다. 이로 인하여 ACO는 페로몬 반영 비율과 임의 탐색 확률이 낮으면 그리디 탐색과 비슷한 특성을 지니게 된다.

ACO에서 페로몬의 증발과 강화 시 페로몬의 증감은 식 (9), (10)과 같다.

$$\text{pheromone evaporation} : \tau_{ik}^{n+1} = (1 - \rho_1) \tau_{ik}^n \quad (9)$$

$$\text{pheromone reinforcement} : \tau_{ik}^{n+1} = (1 + \rho_2) \tau_{ik}^n \quad (10)$$

여기서 τ_{ik}^n 은 n 번째 시행에서 무기 i를 셀 k에 할당하는 해의 페로몬 값, ρ_1 은 페로몬 감소 비율, ρ_2 는 페로몬 증가 비율을 나타낸다.

본 연구에서는 우수해가 갱신되었을 때와 세대의 모든 개미가 해 생성을 완료했을 때 페로몬을 갱신한다. 이러한 방법의 이점은 알고리즘이 한 번의 시행

내에서 각 개미 간의 정보 교류를 최대한 늦춤으로써 각각의 개미가 더 자유롭고 다양한 탐색을 할 수 있다는 것이다. Table 1에서 볼 수 있듯이 본 연구에서 다루는 문제는 탐색해야 할 영역이 매우 넓다. 하지만 탐색 시간이 한정적이므로 초기 해를 다양하게 설정하기 보다는, 어느 정도의 성능을 보장하는 초기 해(그리디 탐색)를 기반으로 반복 시행에서 다양성을 보장하는 방법이 유리하다.

3.5 유전자 알고리즘

본 연구에서는 교배 시행에서 부모를 선택하는 기준을 세 가지로 설정하였다. 첫 번째는 세대 중 가장 우수한 두 해를 부모로 설정하는 것이다. 두 번째는 각 해의 목적함수 값의 순위에 따라 차등적으로 선택 확률을 부여하는 방식이며, 이를 “룰렛 휠 방식 1”로 명명한다. 세 번째는 두 번째 방식에서 확률 값을 보정하여 덜 우수한 유전자가 선택될 확률을 높인 방식으로써, 이를 “룰렛 휠 방식 2”라고 부른다. 세대의 크기가 4인 경우 각 해의 목적함수 값과 내림차순 순위, 그에 따른 각 해의 선택 확률은 아래 Table 2와 같다.

Table 2. An example of choice probability with roulette wheel in case of population size is 4

해	목적함수 값	내림차순 순위	룰렛 휠 방식 1	룰렛 휠 방식 2
S1	15.3	4	4/10	14/50
S2	15.7	3	3/10	13/50
S3	15.8	2	2/10	12/50
S4	16.1	1	1/10	11/50

위 표에서 가장 우수한 해인 S1은 내림차순 순위가 4가 되며, 이에 따른 룰렛 휠 방식 1을 이용한 선택 확률은 (내림차순 순위)/(순위 합) = 4/10이 된다. 룰렛 휠 방식 2는 내림차순 순위 차이의 갭을 줄이기 위하여 {(내림차순 순위) + (순위 합)} / {(순위 합) + (세대 크기) × (순위 합)}으로 계산한다.

본 연구에서는 똑같은 부모를 선택하더라도 교배 시행 후 자식 세대의 해가 서로 다르게 생성되도록 임의의 교배 지점을 지정하여 좌측은 부의 형질을, 우측은 모의 형질을 물려받도록 설정하였다.

GA의 교배 시행 또한 PSO의 이웃 해 생성과 같이 해당 시행으로 생성된 해가 실행 불가능할 수도 있다는 문제를 지니고 있다. 이러한 문제를 해결하기 위하여 GA도 PSO와 같은 방법을 이용하여 실행 불가능한 해를 보정하였다. 다만, GA는 부모 중 어느 쪽이 더 우수하다고 판단하기가 어려우므로 계산 시간을 줄이기 위하여 특별한 연산 과정을 추가하지 않고 좌측부터 차례대로 모든 셀에 대하여 그리디 탐색을 시행하였다.

초기 해가 포함하지 못하는 영역을 탐색하기 위해 돌연변이 시행을 추가한다. 본 연구에서는 교배 시행 직후 확률 α_{ga} 로 돌연변이를 적용하여 해의 다양성을 확보한다.

돌연변이 시행은 PSO의 이웃 해 탐색 과정과 유사하다. 현재 해에서 전체의 10% 만큼 개수의 셀을 실행 가능한 임의의 성분으로 바꾼다. GA의 돌연변이 시행에서는 치환 비율이 너무 커지면 임의 탐색과 유사하게 되어 알고리즘의 성능이 대폭 감소하는 것을 실험을 통해 확인하였다. 따라서 그 값을 10%로 고정하되, 돌연변이를 적용할 확률 α_{ga} 를 조절함으로써 해의 다양성을 통제하였다. 돌연변이 시행으로 생성된 실행 불가능한 해는 교배 시행의 보정과 마찬가지로 좌측부터 차례대로 모든 셀에 대하여 그리디 탐색을 시행함으로써 보정하였다.

4. 실험 설계

국방과학연구소의 도움을 받아 적의 공중 위협체가 아군 방어 자산을 공격하는 가상의 시나리오를 생성하였다. 총 8개의 아군 방공 포대가 100개의 표적을 나누어 담당도록 하였으며, 본 연구에서 제시한 모형은 정적 무기-표적 할당 문제에 적합하도록 설계하였으므로 특정 시점에서의 무기-표적 할당을 시행하였다. 보안상의 이유로 본 연구에 직접적인 관련이 없는 구체적인 수치는 생략하였다.

4.1 적군 데이터

아군 방어 자산을 공격하는 적 공중 위협체의 종류를 전투기와 탄도 미사일로 가정하였다. 전투기는 미리 지정된 시나리오에 따라 비행장으로부터 출발하여 특정 지점을 경유하고 아군 방어 자산을 목표로 일정한

속도로 이동하며, 각 지점 간의 경로는 직선으로 설정하였다. 탄도 미사일은 발사대로부터 출발하여 포물선을 그리며 아군 방어 자산으로 향한다. 탄도 미사일의 구간 별 속도와 궤적은 김홍섭 외(2011)^[12]의 논문에서 연구한 내용을 바탕으로 하였다.

무기 체계 전문가의 의견을 참고하여 평균적으로 탄도 미사일이 전투기보다 두 배 정도 위협하다고 가정하고, 전투기의 위협도를 0.25에서 0.65 사이의 값을 갖는 균등 분포를, 탄도 미사일의 위협도는 0.5에서 0.95 사이의 값을 갖는 균등 분포를 따르도록 설정하였다. 위협도에 따라 전체 100개의 표적 중 60개는 단일 사격, 20개는 연발 사격, 20개는 공유 사격을 요구한다고 가정하였다.

4.2 아군 데이터

모든 무기(포대)는 동일한 종류이며, 무기 용량(보유한 탄약의 양)이 균등하며, 중심각이 90도인 부채꼴 모양의 유효 사정거리를 가지고 있다. 8개의 무기가 표적의 예상 진입 방향을 향하고 있으며, 무기 간 공격 가능 범위가 서로 겹치도록 하여 사거리 내에 진입한 표적에 대하여 공격 무기를 선택할 수 있도록 하였다. 아군 방어 자산은 무기 주변, 혹은 무기 사정거리 이내에 존재한다.

4.3 파괴 확률 산정

본 연구에서는 특정 시점에서 무기와 표적의 거리($\min_{ij}^{distance}$), 무기의 사정거리($range_i$), 무기 방향과 표적 진행 방향의 각도(θ_{ij})를 고려하여 p_{ij} 값을 산정하였다. Fig. 8은 위 세 가지 요소에 대한 예시이다.

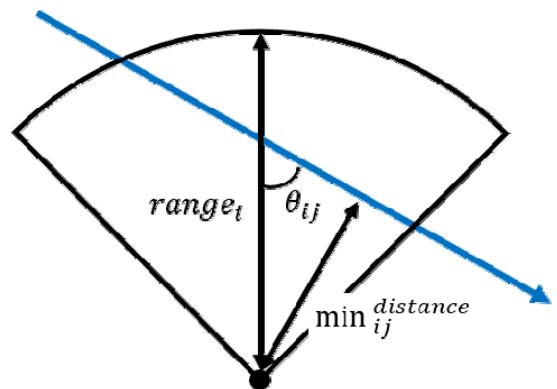


Fig. 8. An example of distance, range, and angle

파괴 확률 p_{ij} 값은 전문가의 조언에 따라 실제 운용되고 있는 특정 무기 체계의 파괴 확률 산정 식을 이용하여 시뮬레이션의 특정 시점의 전장 상황에 대해 추출하였다. 표적이 유효 사정거리 내에 존재할 때 $p_{ij} \in [0.8, 0.95]$ 이며, 유효 사정거리를 벗어난다면 $p_{ij} = 0$ 이 된다. 구체적인 파괴 확률 산정 식은 보안상의 이유로 생략한다.

4.4 실험 시나리오

방공 포대의 표적 할당 문제에서는 격추시킬 표적의 양과 무기 용량의 총 합을의 비율로 유효성을 판단할 수 있다. 무기 용량이 격추시킬 표적의 양을 충분히 커버하지 못한다면 아군이 불리한 상황이라고 할 수 있고, 이 때 의사결정은 “한정된 무기 용량을 어떤 표적에 할당할 것인가”가 된다. 반대의 상황이라면 무기 용량이 풍족하므로 “표적을 어떤 무기에 할당할 것인가”를 잘 선택하는 것이 알고리즘의 성능을 좌우한다. 전자는 무기 용량이 상대적으로 부족하여 대부분의 무기가 모든 용량을 할당할 가능성이 크고, 이에 따라 메타휴리스틱 알고리즘에서 이웃해를 찾기 위해 각 셀에 할당할 수 있는 무기 번호를 탐색할 때 용량이 남아있는 무기가 적어 셀의 성분을 바꾸기 어려운 경우가 많을 것이다. 후자의 경우는 반대로 무기 용량이 풍족하므로 대안으로 선택할 수 있는 무기 번호가 많을 것이고, 한 번의 시행에서 단 2 개의 셀의 성분을 수정하는 SA와 TS가 전자보다 후자에서 더 잘 작동할 것을 기대할 수 있다.

표적 100 개를 모두 공격하기 위해서는 단일 사격 60 회, 연발과 공유 사격 40 회를 해야 하므로 총 140 발의 무기 용량이 필요하다. 따라서 무기 용량의 총 합을 80 발부터 192 발까지 바뀌가며 아군이 불리한 상황에서부터 유리한 상황까지 실험하였다.

4.5 메타휴리스틱 알고리즘 파라미터 설정

메타휴리스틱은 파라미터 설정에 따라 해의 수렴 속도와 수렴 후 목적함수 값이 달라진다. 각 알고리즘의 최적의 파라미터 조합을 찾기 위해 Table 3의 실험을 시행하였다. 모든 실험의 알고리즘 종료 시간은 1초로 설정하였다.

Population 크기는 집단해 기반 메타휴리스틱에서 한 세대에 속하는 개체의 수를 의미한다. PSO 알고리즘에서 현재해와 우수해를 섞을 때, 우수해는 개체(Particle) 우수해와 전체(Global) 우수해 중 하나를 택

하는데, P-G 비율은 개체 우수해와 전체 우수해를 택하는 비율을 나타낸다. 총 실험 횟수는 각 파라미터의 실험 수준과 무기 용량 수준(8 가지), 반복 10 회를 곱하여 나온 수치이다. 실험 결과 파트에서는 각 알고리즘 별 실험에서 종료 시각의 목적함수(식 (1)) 값이 가장 낮은 파라미터 조합을 최적 파라미터 조합으로 보고 결과 비교에 사용하였다. 본 연구의 목적이 메타휴리스틱 간의 비교이므로 각 알고리즘의 파라미터 설정에 따른 성능 변화 실험의 구체적인 결과는 생략한다.

Table 3. Experiment design to find optimal parameter set of metaheuristics

알고리즘	파라미터	실험 수준	총 실험 횟수 (회)
SA	최대 온도 냉각 비율	8	8*4*8*10 = 2,560
		4	
TS	타부 리스트 크기 이웃 탐색 범위	6	6*6*8*10 = 2,880
		6	
PSO	Population 크기 치환 비율 P-G 비율	5	5*5*5*8*10 = 10,000
		5	
		5	
ACO	Population 크기 페로몬 반영 비율 페로몬 증가 비율 페로몬 감소 비율 임의 탐색 비율	5	5*3*3*3*8*10 = 32,400
		3	
		3	
		3	
GA	Population 크기 돌연변이 확률 부모해 선택 방식	5	5*4*3*8*10 = 4,800
		4	
		3	

5. 실험 결과

알고리즘 별 성능 비교는 크게 최종 해의 성능과 탐색 속도 측면에서 이루어진다. 최종 해의 성능은 알고리즘 실행 후 1초가 경과한 시점까지 찾은 해 중 가장 좋은 해의 목적함수 값이다. 탐색 속도는 최종

해 대비 목적함수 값의 차이가 5 % 미만인 해를 찾은 시간을 기준으로 비교한다. 이를 “근사 시간”이라 한다. 근사 시간이 작다는 것은 탐색 초기에 최종 해와 비슷한 목적함수를 갖는 해를 찾았다는 것이고, 이는 알고리즘이 비교적 빨리 수렴한다는 것을 의미한다.

해의 성능 비교 결과는 아래 Table 4과 Fig. 9와 같다.

Table 4. Objective value of each metaheuristic algorithms

무기 용량 (발)	무기 표적 비율 (%)	알고리즘 별 목적함수 값				
		SA	TS	PSO	GA	ACO
80	57	13.15	11.48	14.12	18.96	21.86
96	68	6.75	6.01	8.71	13.37	16.06
112	80	3.47	2.63	5.13	8.71	10.01
128	91	2.77	1.87	3.13	4.11	5.87
144	102	2.49	1.72	1.65	1.86	3.35
160	114	2.03	1.59	1.4	1.62	2.74
176	125	1.89	1.4	1.26	1.49	2.07
192	137	1.75	1.32	1.16	1.32	1.86

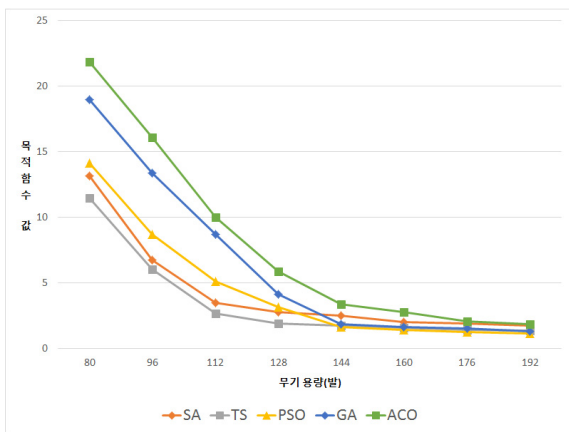


Fig. 9. Tendency of objective value of each metaheuristic algorithm

Table 4에서 무기 용량은 8 개 무기의 용량의 총 합을 나타낸다. 무기 표적 비율은 (무기 용량 총 합) / (표적의 사격 요구량 총 합)을 백분율로 나타낸 값으로써, 각 시나리오의 무기와 표적의 유틸리티를 나타낸다. 무기 표적 비율이 100 %이하이면 무기 용량이 표적을 모두 격추시키기에 충분하지 않으므로 무기가 불리한 상황이라고 할 수 있다.

무기 표적 비율이 100 % 이상일 때, 대부분의 표적에 무기를 할당해 줄 수 있으므로 최적해는 0인 성분을 거의 갖고 있지 않을 것을 쉽게 예측해볼 수 있다. 하지만 단일해 기반 메타휴리스틱(SA, TS)의 경우, 알고리즘 시행 과정에서 하나의 셀을 0으로 할당하는 행위를 수반하게 된다. 이로 인하여 최적해에 근접하기가 쉽지 않고, 무기 표적 비율이 100 % 미만일 경우 두 알고리즘이 항상 목적함수 값이 우수한 것에 비해 상대적으로 낮은 성능을 보이게 된다.

단일해 기반 메타휴리스틱 내부에서 비교를 하자면, TS가 SA보다 항상 우수한 성능을 보인다. SA는 일정 시간 이후에는 지역 최적에 반드시 수렴하도록 설계되어 있지만, TS는 타부 리스트의 크기만 충분하다면 시점에 상관없이 지역 최적에서 빠져나올 수 있다. 모든 해 공간을 탐색할 만큼 시간이 충분하지 않다면 새로운 지역 최적을 계속 찾아 나가는 것이 유리하지만, SA의 경우 온도를 너무 높이면 좋지 않은 해를 많이 수용하므로 그 만큼 탐색 속도가 더더지는 단점이 있다.

집단해 기반 메타휴리스틱끼리의 비교에서는 PSO가 가장 우수한 성능을 보였다. PSO가 GA의 핵심 논리인 교배 시행의 원리를 차용하면서 우수해라는 개념을 도입하였기에 기존해끼리의 교배를 사용하는 GA보다 우수한 성능을 보였다. 또한, GA는 부모 중 더 우수한 형질을 판단하지 않고 해를 보정하지만, PSO는 상대적으로 덜 우수하다고 할 수 있는 기존해의 성분을 선택적으로 보정하므로 GA보다 발전된 알고리즘이라 할 수 있다. 이러한 성질은 PSO가 GA보다 더 강한 지역 탐색 능력을 가지게 해준다. ACO는 모든 경우에서 가장 나쁜 성능을 보였는데, 이는 ACO가 초기 해를 개선시키는 속도가 다른 알고리즘보다 느려 극히 제한된 시간 하에서 효과적인 탐색이 힘들기 때문일 것으로 예상된다.

알고리즘의 탐색 속도 비교 결과는 Table 5와 Fig. 10과 같다.

Table 5. Approach time of each metaheuristic algorithms

무기 용량 (발)	무기 표적 비율 (%)	알고리즘 별 근사 시간(초)				
		SA	TS	PSO	GA	ACO
80	57	0.57	0.67	0.67	0.56	0.31
96	68	0.63	0.67	0.72	0.59	0.35
112	80	0.88	0.83	0.54	0.43	0.74
128	91	0.78	0.81	0.55	0.55	0.78
144	102	0.77	0.73	0.62	0.57	0.62
160	114	0.89	0.7	0.6	0.83	0.73
176	125	0.85	0.78	0.57	0.71	0.2
192	137	0.84	0.86	0.54	0.62	0.74

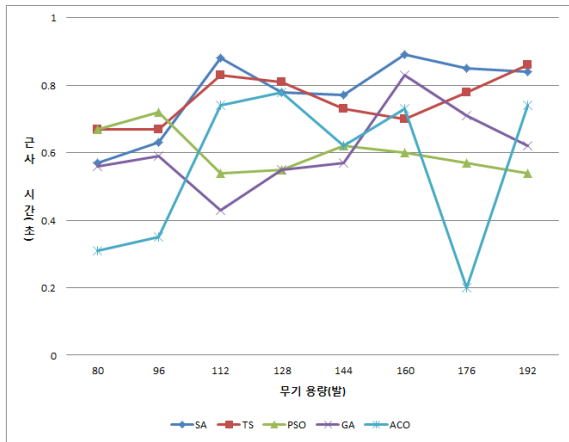


Fig. 10. Tendency of approach time of each metaheuristic algorithm

모든 알고리즘이 무기 용량이 커질수록 근사 시간이 늘어나는 경향을 보인다. 4.4에서 언급하였듯이 무기 용량이 클수록 이웃해 탐색 시 우수해를 발견할 확률이 높으므로 근사 시간도 길게 나타난다.

단일해와 집단해 기반 메타휴리스틱의 비교에서는 집단해 기반 메타휴리스틱의 근사 시간이 대체로 낮게 나타난다. 이에 대한 이유를 크게 2 가지로 예측해 볼 수 있다. 먼저, 단일해 기반 메타휴리스틱은 지역

탐색 능력이 강해 지역 최적을 빠르게 탐색할 수 있지만, 집단해 기반 메타휴리스틱은 다양한 개체가 동시에 탐색을 진행하므로 주어진 시간(1초) 내에 지역 최적에 도달하지 못했을 가능성이 있다. 만약 이러한 경우라면 더 많은 시간이 주어졌을 때 집단해 기반 메타휴리스틱은 특정 값에 수렴하지 않고 우수한 해들을 계속 찾아낼 수 있을 것이다.

또 다른 가능성은 앞의 경우와는 정반대로 집단해 기반 메타휴리스틱이 실제로 빨리 수렴했을 것을 추측해볼 수 있다. 짧은 시간 내의 탐색이므로 최적 파라미터 도출 과정에서 느리게 수렴하지만 1초 이후에 더 좋은 목적함수 값을 갖는 해를 찾는 파라미터가 탈락하고 조기에 수렴하는 파라미터 조합이 최적 조합으로 선정되었을 수 있다.

목적함수 값과 근사 시간을 종합적으로 판단하면, 본 연구에서 다루는 탐색할 공간이 넓으면서도 시간적 제약이 걸린 상황에서는 지역탐색 능력이 우수한 알고리즘이 좋은 성능을 나타내는 것을 관찰하였다. 무기 표적 비율이 100 % 이하인 구간에서는 TS가 가장 우수한 목적함수 값을 가지며 근사 시간이 길게 나타나므로 해 공간을 효과적으로 탐색한다고 판단할 수 있다. 이것은 지역 최적에 쉽게 빠지지 않으면서도 지역탐색 능력이 강한 성질에 기인한다. 무기 표적 비율이 100 % 이상인 구간에서는 알고리즘의 특성 상 단일해 기반 메타휴리스틱의 목적함수 개선 능력이 떨어져 집단해 기반 메타휴리스틱 중 지역탐색 능력이 가장 강한 PSO가 우수한 성능을 나타내었다.

6. 결론 및 토의

본 연구에서는 지대공 전투에 대하여 현실적 제약인 연발 사격과 공유 사격 제약을 포함한 새로운 무기-표적 할당 문제를 제시하였다.

짧은 시간 내에 다수의 무기와 다수의 표적을 할당하기 위하여 다섯 가지 메타휴리스틱을 구현하여 비교 분석하였다. 그 결과, 모든 표적을 격추시키기에 무기 용량이 부족한 경우는 TS가, 무기 용량이 풍족한 경우는 PSO가 성능이 우수함 확인할 수 있었다. 또한, 제한된 짧은 시간 내의 탐색의 경우, 전체 해 공간 중 탐색하지 못하는 영역이 훨씬 넓으므로, 지역탐색 능력이 강하면서도 지역 최적에 쉽게 빠지지 않는 TS가 대체적으로 우수한 성능을 나타내었다.

본 연구의 해 표현과 알고리즘을 바탕으로 3 연발 및 n 연발과 공유사격을 허용하는 일반적인 무기-표적 할당 문제의 해도 효과적으로 탐색할 수 있을 것으로 판단한다.

TS가 무기 표적 비율 100 % 이상인 상황에서 최적 해에 근접하기 힘들다는 약점이 존재하지만 대체적으로 성능이 우수하므로, 추후 연구에서는 약점을 보완할 수 있는 새로운 이웃 해 탐색 방식을 적용하면 더욱 좋은 결과를 얻을 것으로 예상된다.

후 기

본 연구는 국방과학연구소(UD140067CD)의 지원을 받아 작성되었음.

References

- [1] Lloyd, Stuart P., and Hans S. Witsenhausen, "Weapons Allocation is NP-Complete," 1986 Summer Computer Simulation Conference, 1986.
- [2] Xin, B, Chen, J, Zhang, J, Dou, L, and Peng, Z, "Efficient Decision Makings for Dynamic Weapon-Target Assignment by Virtual Permutation and Tabu Search Heuristics," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 40, No. 6, pp. 649-662, 2010.
- [3] Lee Z. J., Lee C. Y., and Su S. F., "An Immunity-Based Ant Colony Optimization Algorithm for Solving Weapon-Target Assignment Problem," Applied Soft Computing, Vol. 2, No. 1, pp. 39-47, 2002.
- [4] Lee Dong Eon, Oh Haeng Rok, Kauh Jang Hyuk, Park Young Woo, and Lee Yee Jin, "Improved Ant Colony Optimization Algorithm Design for Weapon-Target Assignment," Proceedings of Symposium of the Korean Institute of Communications and Information Sciences, pp. 410-411, 2011.
- [5] Sung-Sam Hong, Jeon-Hee Yoo, Seung-Jin Park, Chang-Min Mun, and Myung-Mook Han, "A Suite of Heuristic Algorithms for Weapon-Target Allocation in Air Defense," Journal of Korean Society for Internet Information, Vol. 13, No. 1, pp. 195-196, 2012.
- [6] Lee Z. J., and Lee W. L., "A Hybrid Search Algorithm of Ant Colony Optimization and Genetic Algorithm Applied to Weapon-Target Assignment Problems," International Conference on Intelligent Data Engineering and Automated Learning, pp. 278-285, 2003.
- [7] Lee Z. J., and Lee C. Y., "A Hybrid Search Algorithm with Heuristics for Resource Allocation Problem," Information Sciences, 173(1), pp. 155-167, 2005.
- [8] Lee Z. J., Su S. F., & Lee C. Y., "A Genetic Algorithm with Domain Knowledge for Weapon Target Assignment Problems," Journal of the Chinese Institute of Engineers, 25(3), 287-295, 2002.
- [9] Lee Z. J., Su S. F., & Lee C. Y., "Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms with Greedy Eugenics," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 33(1), 113-121, 2003.
- [10] Julstrom. B. A, "String-And Permutation-Coded Genetic Algorithms for the Static Weapon-Target Assignment Problem," In Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers ACM, 2553-2558, 2009
- [11] Zeng. X, Zhu. Y, Nan. L, Hu. K, Niu. B, & He. X, "Solving Weapon-Target Assignment Problem using Discrete Particle Swarm Optimization," Intelligent Control and Automation, 2006, WCICA 2006, The Sixth World Congress on IEEE, Vol. 1, pp. 3562-3565, 2006
- [12] Hosein, Patrick A., and Michael Athans, "Some Analytical Results for the Dynamic Weapon-Target Allocation Problem," No. LIDS-P-1944. Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1990.
- [13] Kim Heung Seob, Kim Ki Tae, and Jeon Geon Wook, "A Requirement Assessment Algorithm for Anti-Ballistic Missile Considering Ballistic Missile's Flight Characteristics," Journal of the Korea Institute of Military Science and Technology, 14.6: 1009-1017, 2011.

Appendix

1. SA의 Pseudo-code

Begin

find a initial solution through Greedy Search Algorithm

While termination criteria not met **Do**

find a neighbor solution

determine to adopt it or not

If current solution's objective < best solution's **Then**

best solution = current solution

End If

End While

Return best solution

End

2. TS의 Pseudo-code

Begin

find a initial solution through Greedy Search Algorithm

While termination criteria not met **Do**

For i=1 **To** neighbor size

find a neighbor solution

End For

choose the best neighbor do not restricted by tabu list

update tabu list

If current solution's objective < best solution's **Then**

best solution = current solution

End If

End While

Return best solution

End

3. PSO의 Pseudo-code

Begin

find a initial solution through Greedy Search Algorithm

For i=1 **To** population size-1

find initial solution through Random Search Algorithm

End For

While termination criteria not met **Do**

find new solutions

update particle and global best solution

End While

Return global best solution

End

4. ACO의 Pseudo-code

Begin

For i=1 **To** population size

set different weapons to the first cell by each solution

End For

While termination criteria not met **Do**

find new solutions

For i=1 **To** population size

If current solution's objective < best solution's **Then**

best solution = current solution

update pheromone with best solution

End If

End For

update pheromone with current solution

End While

Return best solution

End

5. GA의 Pseudo-code

Begin

find one initial solution though Greedy Search Algorithm

For i=1 **To** population size-1

find initial solution though Random Search Algorithm

End For

While termination criteria not met **Do**

sorting the population by ascending order of objective

For i=2 **To** population size

find new solutions through crossover operation

mutate solution with probability of α_{ga}

If current solution's objective < best solution's **Then**

best solution = current solution

End If

End For

End While

Return best solution

End