經營科學
第34卷 第1號
2017年 3月, pp.47-56
https://doi.org/10.7737/KMSR.2017.34.1.047

비계획 사격상황에서 적 위협 최소화를 위한 실시간 사격순서 결정 연구

차영호[1] · 방준영[2†] · 심상오[3]
[1]육군 분석평가단, [2]성결대학교 산업경영공학부, [3]국립한밭대학교 경영회계학과

Real-time Algorithms to Minimize the Threatening Probability in a Fire Scheduling Problem for Unplanned Artillery Attack Operation

Young-Ho Cha[1] · June-Young Bang[2†] · Sangoh Shim[3]
[1]Center for Army Analysis and Simulation
[2]Department of Industrial and Management Engineering, Sungkyul University
[3]Department of Business Administration, Hanbat National University

Abstract

We focus on the Real time Fire Scheduling Problem (RFSP), the problem of determining the sequence of targets to be fired at, for the objective of minimizing threatening probability to achieve tactical goals. In this paper, we assume that there are m available weapons to fire at n targets ($\rangle m$) and the weapons are already allocated to targets. One weapon or multiple weapons can fire at one target and these fire operations should start simultaneously while the finish time of them may be different. We suggest mathematical modeling for RFSP and several heuristic algorithms. Computational experiments are performed on randomly generated test problems and results show that the suggested algorithms outperform the firing method which is generally adopted in the field artillery.

Keywords : Military, Firing Scheduling, Real Time Algorithm

논문접수일 : 2017년 02월 10일 논문게재확정일 : 2017년 03월 10일
논문수정일 : 2017년 03월 07일
† 교신저자, jybang@sungkyul.ac.kr

1. 서 론

본 연구에서는 다수 포병부대의 작전지역이 할당된 상황에서 작전지역내에서 식별된 이동 및 고정표적에 대한 비계획 사격 문제를 다룬다. 포병부대의 비계획 사격은 표적위치 및 표적성질 등 특성 정보가 식별된 것으로 가정하고, 이 정보를 바탕으로 실시간으로 사격계획을 수립한다. 작전이 개시되면 공대지사격, 지대지사격 및 미사일 공격 등 다양한 아군 공격수단을 활용하여 실시간으로 탐지된 적 표적들을 효과적으로 파괴하여야 한다. 본 연구에서는 여러 공격유형 중 작전지역이 사전에 할당되어 있는 포병부대의 효과적인 비계획 사격 방법론을 제안하였다.

포병부대의 비계획 사격 작전은 두 단계의 의사결정을 거쳐서 수립된다. 우선, 할당 단계에서는 아군 포병부대의 작전지역 내 식별된 표적을 각 포대에 할당한다. 본 연구에서 사용하는 '포병부대' 용어는 사격을 위한 최소단위로써 곡사포 6문으로 구성된 포대를 의미한다. 작전개시 이후, 각 포대는 작전지역에서 출현하는 적 표적을 담당한다. 작전지역에 2개 이상의 적 표적이 탐지된 경우, 모든 표적을 동시에 고려한 할당계획이 수립되어야 한다. 다음 단계인 사격 스케줄링 단계에서는 각각의 포대에 할당된 표적에 대한 사격순서와 사격개시 시점을 결정한다. 작전개시 이후 위치가 식별된 적 표적들을 각각의 작전지역을 담당하는 포대에 할당하고, 해당 표적 규모를 고려하여 대규모 표적일 경우 인근의 작전지역을 담당하는 포대를 추가 할당할 수 있으며, 할당된 모든 포대들은 해당 표적에 동시에 사격을 시작하여야 한다. 본 연구에서는 아군 포병부대와 작전 지역과의 할당관계는 주어진 것으로 가정하고, 각 무기의 사격순서와 시간을 결정하는 실시간 사격순서 결정문제(RFSP; Real time Fire Scheduling Problem)로 한정한다. 계획사격 상황은 할당-스케줄링 단계가 작전개시되기 전 사전에 준비되어야 하는 반면, 본 연구에서는 작전 개시 후 예상되는 비계획 사격 상황을 다루었으므로 실시간에 스케줄링 단계를 위한 신속한 해를 제시하는 데 연구의 목적이 있다.

RFSP는 다중 프로세서 작업순서 결정문제(MTSP; Multiprocessor Task Scheduling Problem)의 일반화된 형태로 볼 수 있다. Hoogeveen et al.[19]은 n개의 작업이 m개의 프로세서에서 수행되는 모델을 제안하였다. 작업 $J_j(j = 1, \cdots, n)$는 프로세서를 p_j 시간 동안 점유하여 각 작업은 미리 정해진 프로세서들을 동시에 사용하여 수행할 수 있고, 작업이 일단 시작되면, 종료 시까지 다른 작업으로 인해 중단되지 않는다. 각 프로세서는 동시에 최대 1개의 작업을 수행할 수 있다. RFSP 문제의 경우, 동일한 사격(Operation)의 경우에도 할당된 포대의 종류에 따라 사격지속시간 (작업시간)이 달라질 수 있으므로 한 작업을 위한 프로세서 별 작업시간이 동일하지 않은 MTSP의 일반적인 문제로 볼 수 있다.

사격순서 결정문제(FSP)는 매우 적은 수의 연구가 진행되어 왔다. Kwon et al.[26]은 표적이 고정되어 움직이지 않는 다는 가정 하에 사격 종료 시간을 최소화하는 문제를 고려하였다. 해당 연구에서는 사격작전을 최대한 빠르게 종료함으로써 기습의 효과를 얻고자 하였다. Kim and Lee[23]은 Kwon et al.[26]과 동일한 FSP를 고려하였고, 휴리스틱 알고리즘을 제안하였다. Kim and Lee[22]은 표적할당 문제와 사격순서 문제를 결합하였고, 휴리스틱 알고리즘을 제시하였다. 김동현과 이영훈[1]은 표적 할당과 사격 순서를 동시에 결정하는 휴리스틱 알고리즘을 개발하였다. 정치영[4]은 격추확률을 최대화하는 미사일 배치문제를 혼합정수계획법으로 모델링하였고, 이대력과 양재환[3]은 요격미사일의 할당 및 교전 일정계획 문제를 혼합정수 계획법으로 모델링하였다. 윤상흠[2]과 차영호[5]는 계획 사격 상황 하에서의 분지한계 알고리즘을 개발하였고, 최용백과 김경섭[6]은 실시간 사격 상황에서 사격시간에 기반한 간단한 실시간 알고리즘을 제시하였다.

FSP를 연구한 문헌이 매우 부족하여, 본 연구에서는 MTSP을 주로 참고하여 연구하였다. MTSP는 설비 일정계획 이론과 상당한 유사성이 있다. Bozoki et al.[10]가 MTSP 문제를 처음으로 다루었다. 그러

나, 그의 연구는 당시에는 크게 주목 받지 못하다가 1990년대 컴퓨터의 계산 능력이 획기적으로 개선되고 나서야 재조명되어 활발하게 연구되기 시작했다. Drozdowski[15]와 Lee et al.[27]이 정리한 조사 논문에 상당량의 MTSP 문제를 다루었다. 이 중에서 Bozoki et al.[10], Bianco et al.[8]와 Krämer[24]는 MTSP를 위한 분지한계법을 제안하였다.

Hall[18]에 의하면 MTSP는 $P_k|fix|C_{max}$로 표현된다. 많은 연구자들이 $P_k|fix|C_{max}$ 문제의 실행가능성(feasibility)과 근사성(approximability)을 연구하였다. $P_2|fix|C_{max}$ 문제는 고전적인 2개 설비의 스케줄링 문제의 일반화된 문제이고[16], 따라서 MTSP 문제는 NP-hard이다. Kubale[25]은 하나의 작업이 최대 2개의 미리 정해진 프로세서를 점유할 수 있는 보다 제한된 문제의 복잡도(complexity)에 대하여 연구하였다. Chen and Lee[13]은 $P_2|fix|C_{max}$의 다항시간 근사 알고리즘을 개발하였다. Hoogeveen et al.[19]은 $P_3|fix|C_{max}$ 문제가 the strong sense로 NP-hard이고, 따라서 P = NP가 아닌 이상, 다항시간 근사 방법 알고리즘이 존재할 수 없음을 보였다. Blazewicz et al.[9]은 $P_3|fix|C_{max}$ 문제에 대해서 4/3비율로 다항시간 근사 방법 알고리즘을 개발하였고, Dell'Olmo et al.[14]가 이를 5/4비율의 다항시간 근사 방법 알고리즘으로 개선하였다. 두 알고리즘은 일반적인 일정계획 알고리즘에 기반한 특정 문제에 기초한 알고리즘이다. Goemans[17]은 $P_3|fix|C_{max}$ 문제의 알고리즘을 최적해의 7/6 수준까지 제시하는 다항시간 근사법 알고리즘으로 개선하였다. Brucker et al.[11]은 보다 일반적인 목적식을 가진 멀티프로세서 작업 문제에 대해서 다항시간 알고리즘을 제안하였다. 최근의 연구결과를 살펴보면, Amoura et al.[7]은 $P_k|fix|C_{max}$ 문제의 모든 정수 k에 대한 다항시간 근사법 알고리즘을 개발하였다. $P_k|fix|C_{max}$ 문제의 보다 일반화된 다항시간 근사법 알고리즘이 최근에 Chen et al.[12]과 Jansen et al.[21]에 의하여 개발되었다. Huang et al.[20]은 $P_4|fix|C_{max}$ 문제를 위한 최적해의 1.5 수준에 해당하는 근사 알고리즘을 개발하였는데 이 연구는 기존의 최적해의 2.0 수준의 알고리즘을 상당부분 향상시켰다는 의의가 있다.

본 논문을 다음과 같이 구성하였다. 제 2장에서 실시간 비계획 사격계획 문제 상황을 상세히 기술하고 수학적 모델을 제시하였다. 제 3장에서 실시간 사격계획을 제시할 수 있는 휴리스틱 알고리즘을 제시하였고, 제 4장에서 실제 비계획 사격 상황에 기반한 문제로 제안한 휴리스틱 알고리즘의 성능을 검증하였다. 마지막으로 제 5장에서 본 연구의 내용을 요약하였고 향후 연구 전망을 기술하였다.

2. 실시간 비계획 사격 문제 정의

본 절에서는 다수의 아군포대에 대한 실시간 비계획 사격 스케줄링 문제를 구체적으로 서술한다. 각 아군포대는 고유의 작전지역이 할당되어 있으며, 할당된 작전지역에 출현한 표적은 해당 아군포대가 전담하여 사격한다. 적의 주타격 방향 등 대규모 표적 식별이 예상되는 지역에는 2개 이상의 아군 포대를 할당하거나 작전지역을 겹치도록 할당하여 2개 이상의 포대에 의한 동시 사격이 가능하도록 운영한다.

적 표적의 파괴 확률은 최초 포병사격 이후 경과시간에 비례하여 감소하는 것으로 가정하였는데, 이것은 아군 포병사격이 개시되면 적은 전투상황을 인지한 후 이동하거나 방호 상태를 추가적으로 갖추게 된다는 가정하에서 성립한다. 본 연구에서는 제외되었으나, 이동한 표적의 위치를 재확인하였을 때, 이 파괴확률을 초기값으로 환원하여, 보다 실제에 가까운 문제로 제시할 수 있다. 문제 상황을 요약하면 다음과 같다.

1. 아군 포대의 작전지역과 포대-표적 할당은 사전에 완료되어 있고, 각 작전지역에서 신규로 식별된 표적은 해당 포대에서 담당한다.
2. 담당 지역의 단일 포대로 격멸하지 못하는 대규모 표적은 인근 작전지역을 담당한 포대와 동시 사격한다.
3. 포대의 사격을 상이한 표적으로 전환하기 위해서는 일정시간의 사격 전환 시간이 소모된다.

4. 적 표적의 파괴 확률은 최초 사격 이후 경과시간에 비례하여 감소한다.
5. 최초 사격이 시작된 이후 적군의 대응 사격으로 인한 아군의 전력 손실은 고려하지 않는다.

본 연구에서 고려하는 실시간 사격순서 결정문제의 목적식은 표적 위협도의 합으로 정의하였으며 이 값을 최소화하는 것이다. 즉, 표적의 파괴확률이 시간에 비례하여 감소하는 가운데, 아군에게 미치는 표적의 위협도를 최소화하는 사격순서를 결정하고자 한다. 다음은 사격 순서 결정 알고리즘과 수학적 모델에서 사용한 표기법이다.

인덱스와 파라메터

i 아군포대의 인덱스(i = 1, …, m)
j 표적의 인덱스(j = 1, …, n)
V_j 표적 j의 초기 위협도
E_j 표적 j에 할당된 아군 포대의 집합
d_{ij} 아군 포대 i가 표적 j에 사격해야 하는 시간
p_{ij} 초기 파괴확률. 즉, 표적 j가 발견되었을 때, 아군 포대 i가 즉시 사격시의 파괴확률.
s_i 아군 포대 i가 표적을 바꾸어 사격하기 위한 준비시간
α_j 표적 j의 파괴확률의 시간 지연에 따른 감소율
$W(j,k)$ 표적 j와 k에 동시 사격이 필요한 아군 포대의 집합
M 매우 큰 수(> 1,000,000)

결정변수

x_{jk} 사격 순서를 결정하는 이진 변수. 표적 j에 대한 사격이 표적 k에 대한 사격보다 앞서 실행되는 경우에 1, 아니면 0.
t_j 표적 j에 대한 사격을 시작하는 시점.

위와 같이 정의 되었을 때, 주어진 표적에 대한 사격 순서 결정 문제의 비선형 정수 계획 모델은 다음과 같다[2, 5].

$$\text{Min} \sum_{j=1} V_j \prod_{i \in E_j} \{1 - \max(p_{ij} - t_j \alpha_j, 0)\}$$

Subject to

$$t_k - t_j + M(1 - x_{jk}) \le \max_{i \in W(j,k)} (d_{ij}) + s_i \quad \forall i \in W(j,k),\ j,\ k \tag{1}$$

$$t_j - t_k + M x_{jk} \le \max_{i \in W(j,k)} (d_{ik}) + s_i \quad \forall i \in W(j,k),\ j,\ k \tag{2}$$

$$t_j \ge 0 \quad \forall j \tag{3}$$

$$x_{jk} \in \{0, 1\} \quad \forall j,\ k (j < k) \tag{4}$$

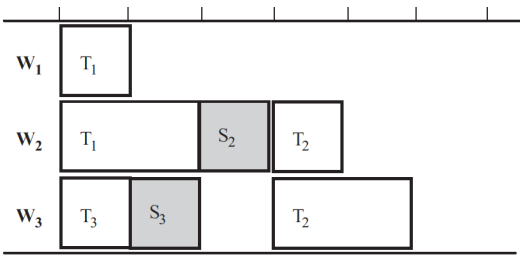
목적식은 모든 사격의 종료시점에 잔류하는 표적의 위협도와 파괴되지 않을 확률의 곱들의 총합이고, 본 연구에서는 이 값을 최소화하는 사격 순서를 결정한다. 각 표적의 파괴확률은 포격전 시작 시점 혹은 표적 탐지시점으로부터 t_j의 비율로 선형적으로 감소한다고 가정하였다. 사격의 첫 시간구간은 0부터 시작하고, k번째 시간 구간은 k-1 시점부터 시작한다. p_{ij}는 0시점에서의 표적의 파괴확률이다. 파괴확률은 시간이 지남에 따라 선형으로 감소하고, t_j시점부터 표적 j에 대한 사격이 시작되므로, 사격 후 표적이 파괴되지 않을 확률은 $\{1 - \max(p_{ij} - t_j \alpha_j, 0)\}$이다. 제약식 (1)과 (2)는 두 개의 표적에 대한 사격 순서의 선후 관계에 대한 분리(Disjunctive) 제약이다. 여기에 쓰인 M은 x_{jk} = 0이나 x_{jk} = 1의 값을 가질 때, 두 식이 모두 만족될 수 있을 정도로 큰 값이다. 즉, $M \ge \max_{i,j}(d_{ij}) + \max_i s_i + \sum_{j=1}^{n} \max_i (d_{ij})$을 만족하는 수로 정한다.

사격순서의 표현 방법

본 연구에서 사격의 부분 스케쥴은 사격작업의 순서로 표현한다. 각각의 포대에 대해서 먼저 사

격하게 되는 표적은 표현법의 앞쪽에 위치하고 나중에 사격하게 되는 표적은 뒤쪽에 위치한다. 표적과 아군 포대와의 할당은 이미 주어진 것으로 가정했으므로, 표적의 사격 순서만으로 포대가 사격할 사격의 스케쥴이 결정된다. 단, 두 개 이상의 아군 포대가 동시에 사격해야 하는 표적에 대해서는 동시에 사격을 시작해야 한다.

<그림 1>에 간단한 예를 제시하였다. W_i는 아군포대, T_j는 표적을 의미한다. 할당관계와 사격시간은 표 1에 표시하였다. W_1과 W_2가 T_1을 동시 사격해야 하고, W_2와 W_3이 T_2를 동시 사격해야 한다. 그림1과 같은 사격 스케쥴은 (1-3-2) 또는 (3-1-2) 두 가지로 표현할 수 있다. 이와 같이 동일한 사격 스케쥴이 여러 순서열로 나타날 수 있는 문제를 방지하기 위하여, 동일 시점에 사격을 시작하는 표적 중 인덱스의 숫자가 작은 것부터 앞에 놓음으로써 1개의 사격 순서는 1개의 표현법으로 나타낼 수 있다.



〈Figure 1〉 A Simple Example for Solution Representation

〈Table 1〉 Data for the Example

Target	Weapon[a]	Duration
1	1(2)	2 periods
	3(1)	1
2	2(1)	1
3	3(1)	2
4	1(2)	1
	2(1)	1

[a]Setup times(in periods) for the weapons are given in parentheses.

3. 휴리스틱 알고리즘

본 절에서는 비계획 실시간 사격 순서 결정문제 (RFSP)의 사격 순서를 구하기 위한 휴리스틱 알고리즘을 제시한다. 제 2장에서 제시한 수학적 모델은 비선형 정수계획법으로 작은 크기의 문제에서도 상당한 시간을 소모하여 해를 도출하게 된다. 실시간으로 사격 대응을 해야 하는 실제 문제의 특성 상, 신속하게 사격 순서를 도출해야 한다. 따라서, 간단하고도 빠르게 해를 도출하여 각 포대에 사격 순서 지시를 내리고, 사격 중 새로운 표적이 출현하였을 경우, 기존 사격 순서를 재계산하여 수정 사격 지시를 내릴 수 있는 신속한 휴리스틱 방법론이 필요하다. 새로운 표적이 발생하거나, 기존 표적이 이동하여 탐지되지 않다가, 사격 중 새로운 표적이 탐지 되면, 파괴확률을 초기값 p_{ij}로 재설정된다. 본 연구에서는 Dispatching rule에 기반하여 사격 순서를 도출하는 방식과 주어진 해를 개선할 수 있는 방법을 제시하였다.

본 연구에서 제시한 방법론은 새로운 적 표적이 출현하지 않는 경우에는 계획사격 문제에서 상한 값(Upper Bound)으로도 활용할 수 있다.

3.1 Dispatching Rule 기반의 구성 알고리즘 (Constructive Algorithm)

Dispatching rule 기반의 순서 결정 알고리즘은 간단한 규칙으로 빠르게 해를 도출할 수 있는 장점이 있으나, 표적의 순서를 결정하는 우선순위 규칙에 따라 결과가 큰 차이를 보인다. 본 연구에서는 다음의 7가지 우선순위 규칙을 제안하였다. 단, 2개 이상의 포대가 동시 사격해야 하는 표적의 경우에는 우선순위에 의해 순서가 확정되면 해당하는 포대가 동시에 사격에 참여하는 것으로 한다.

1) 최소사격시간 표적 우선 규칙(SPT : Shortest Processing Time)

표적을 파괴하기 위하여 필요한 시간이 짧은 표

적부터 사격하는 규칙이다. 1개 이상의 포대가 사격해야 하는 표적은 각 포대에서 사격하는 시간의 합으로 정한다. SPT는 표적의 수를 빠르게 줄이고자 하는 규칙이다.

2) 위협도가 가장 높은 표적 우선 규칙(LTR : Largest Threatening Rate first)
위협도 V_j가 가장 높은 표적을 우선으로 사격하는 방법으로써 현재 비계획 사격 교전 시에 일반적으로 적용하는 규칙이다.

3) 긴급도($V_j \cdot \alpha_j$)가 가장 높은 표적 우선(LER : Largest Emergency Rate first)
위협도 V_j와 함께 표적 파괴 확률을 동시에 고려하여, 위협도가 높고 이동성을 지닌 표적에 우선적으로 사격을 가하는 규칙이다.

4) 위협도/사격 시간 우선 규칙(LTR/D : Largest Threatening Rate over Duration first)
위협도가 높으면서 사격시간이 짧은 표적에 우선순위를 두는 규칙이다. 긴급도와 SPT 규칙을 결합한 규칙이다.

5) 목적식 $V_j\prod_{i\in E_j}\{1-\max(p_{ij}-t_j\alpha_j,0)\}$ 우선 규칙(MinC)
위협도와 시간에 따른 파괴확률의 감소를 고려하는 규칙으로써, 사격 시점 t_j는 이미 사격 순서를 결정한 표적들의 부분 사격 순서(partial sequence)를 고려한 가장 빠르게 사격할 수 있는 시점으로 정한다.

6) $V_j\prod_{i\in E_j}\{1-\max(p_{ij}-t_j\alpha_j,0)\}/d_j$ 우선 규칙(MinC/D)
목적식에 기여하는 값을 사격 시간으로 나눈 값이 높은 표적을 우선 사격하는 규칙이다. 사격 전 초반에 효과적으로 목적값을 높이는 의도로 제안한 우선순위 규칙이다.

3.2 NEH 기반 알고리즘(MNEH)

NEH(Nawaz Enscore Ham) 알고리즘은 m-machine flowshop의 일정계획을 수립을 위해 Nawaz et al. [28]가 제안하였으며, 공저자 3명의 이름 머릿글자를 이용하여 NEH 알고리즘이라고 명명하였다. NEH 알고리즘에서는 작업시간의 합의 내림차순으로 작업들을 정렬하고, 기존에 구성된 부분 사격 순서의 사이마다 삽입하여 가장 좋은 부분 사격 순서의 목적식 값을 갖는 위치에 삽입하는 지역 탐색 알고리즘이다. 본 연구의 문제에 적용하기 위하여, 초기 작업들을 정렬할 때 작업시간 대신 사격의 긴급도를 적용하여 긴급도의 내림차순으로 작업들을 정렬한다. 즉, 초기 위협도(V_j)와 표적 j의 파괴확률의 시간 지연에 따른 감소율 α_j이 긴급도의 중요 요소이므로 $V_j \cdot \alpha_j$의 내림차순으로 표적들을 정렬한다. 이후는 NEH의 방법을 그대로 따르되, 부분 사격 순서를 다음의 값이 최대화가 되는 위치에 삽입한다.

$$\sum_{j=1} V_j\prod_{i\in E_j}\{1-\max(p_{ij}-t_j\alpha_j,0)\}$$

3.3 개선 알고리즘(improvement Algorithm)

제 3.1절과 제 3.2절의 알고리즘으로 구한 초기해를 기준으로 지역탐색 방법을 이용하여 이웃해를 구성하고 이 이웃해 중에서 가장 좋은 목적값을 제시하는 해를 새로운 현재해(current solution)로 선택하고 계속해서 지역탐색을 진행한다. 이웃해를 만드는 방법은 삽입(Insertion; IS), 교환(Interchange; IC), 삽입-교환(ISIC), 교환-삽입(ICIS) 등의 네 가지 방법론을 개발하여 적용하였다.

IS 알고리즘에서는 주어진 초기해에 대해서 1번째 표적부터 선택하여 이후의 사격 순서 사이마다 삽입하여 목적식 값이 개선되는 경우 위치를 변경한다. 이 과정을 반복하여 마지막 표적 순서까지 진행하면 종료한다. IC알고리즘에서는 첫 순서 표적과 이후 순서의 표적의 사격 순서를 1:1 교환하여 목적식 값이 개선되는 경우 위치를 서로 교환한다. 이 과정을 반복하여 마지막 표적 순서까지 진행하면 종료한다.

IS와 IC 알고리즘의 세부 과정은 다음과 같다.

IS 알고리즘

Step 1. 주어진 초기해 Λ_0를 Λ로 설정하고, 인덱스 p, q는 각각 $p \leftarrow 1$, $q \leftarrow p+1$로 초기화 한다.

Step 2. 현재해 Λ에서 p번째 표적을 q번째와 $q+1$번째 표적 사이로 위치를 변경하여 새로운 해 Λ'을 생성한다.

Step 3. Λ'의 목적식 $V_j\prod_{i\in E_j}\{1-\max(p_{ij}-t_j\alpha_j, 0)\}$을 계산하여 Λ의 목적식 값보다 개선된 경우, $\Lambda \leftarrow \Lambda'$로 변경하고 Step 2로 이동한다. 개선되지 않은 경우 step 4로 이동한다.

Step 4. q번째 위치의 표적이 현재해 Λ의 마지막 위치이면 step 5로 이동한다. 아닌 경우에는 $q \leftarrow q+1$로 설정한 후 step 2로 이동한다.

Step 5. $p+1$번째 위치의 표적이 현재해 ㉮의 마지막 위치이면 알고리즘을 종료한다. 아닌 경우에는 $p \leftarrow p+1$, $q \leftarrow p+1$로 설정한 후 step 2로 이동한다.

IC 알고리즘

Step 1. 주어진 초기해 Λ_0를 Λ로 설정하고, 인덱스 p, q는 각각 $p \leftarrow 1$, $q \leftarrow p+1$로 초기화 한다.

Step 2. 현재해 Λ에서 p번째 표적을 q번째 표적과 교환하여 새로운 해 Λ'을 생성한다.

Step 3. Λ'의 목적식 $V_j\prod_{i\in E_j}\{1-\max(p_{ij}-t_j\alpha_j, 0)\}$을 계산하여 Λ의 목적식 값보다 개선된 경우, $\Lambda \leftarrow \Lambda'$로 변경하고 Step 5로 이동한다. 개선되지 않은 경우 step 4로 이동한다.

Step 4. q번째 위치의 표적이 현재해 Λ의 마지막 위치이면 step 5로 이동한다. 아닌 경우에는 $q \leftarrow q+1$로 설정한 후 step 2로 이동한다.

Step 5. $p+1$번째 위치의 표적이 현재해 Λ의 마지막 위치이면 알고리즘을 종료한다. 아닌 경우에는 $p \leftarrow p+1$, $q \leftarrow p+1$로 설정한 후 step 2로 이동한다.

ISIC와 ICIS는 IS와 IC를 복합적으로 적용하는 개선 알고리즘이다. ISIC는 IS를 먼저 시행하여 나온 1차 개선해에 대해서 IC를 재차 시행하여 최종 개선해를 얻고, ICIS 알고리즘은 IC를 먼저 시행하고 IS를 시행하여 최종 개선해를 얻는 방법론이다.

4. 실험 및 결과

먼저 작은 크기의 문제에 대해서 제 2장에서 제시한 사격 순서 결정을 위한 수학적 모델에 대하여 ILOG CPLEX 11.0을 이용하여 최적해를 도출하였다. 2절에서 제시한 사격 순서 결정 모델은 포대의 수가 목적함수의 지수값에 해당하여 비선형 모델이 된다. CPLEX는 제약식과 목적식이 1차식인 Linear Programming과 제약식이 1차식이고 목적식의 차수가 2차인 Quadratic Programming에 대한 해를 제공한다. 따라서, 포대의 수를 2로 제한하여 최적해를 계산하였다. 이 테스트를 위하여 33개의 문제를 생성하였다. 즉, 아군 포대의 수 1단계(2개 포대), 표적의 숫자에 따라 11단계(10~20개), 단독 포대-표적 케이스의 비율을 세 단계(20%, 50%, 80%)로 구분하였다. 최적해의 결과를 <표 2>에 표기하였다. 이 표에는 평균 CPU 소요시간, 제한 시간인 3,600초 안에 최적해를 구하지 못한 문제의 수를 명시하였다. <표 2>에서와 같이 포대의 수가 2개인 작은 크기의 문제에 대해서도 최적해를 도출하는데 1시간 이상의 시간이 소요됨을 알 수 있다.

휴리스틱 알고리즘을 비교하기 위하여, 총 135개 문제를 실험하였다. 포대 수를 3단계(3, 6, 9개), 표적의 수를 3단계(22, 24, 26 개 표적), 단독 포대-표적 케이스의 비율을 세 단계(20%, 50%, 80%)로 구분하였고, 다중 포대-표적에 대해서는 파괴 필요 포대 수를 2와 3사이에 같은 확률로 발생시켰다. 즉 다중 포대-표적이 전체의 80%인 경우 40%는 필요 포대가 2개, 나머지 40%는 필요 포대가 3개인 것으로 정하였다.

〈Table 2〉 Results of the CPLEX

		CPLEX	
W[a]	T[b]	CPUT[c]	NPNS[d]
2	10	0.958	0
	11	2.068	0
	12	90.323	0
	13	225.078	0
	14	140.672	0
	15	650.036	0
	16	1305.934	1
	17	2492.024	2
	18	2545.585	2
	19	3600.033	3
	20	3600.417	3

[a] number of weapons
[b] number of targets
[c] average CPU time(in seconds) required to solve a problem
[d] number of problems(among 15 problems) that were not solved to the optimality in 3,600 seconds

본 연구에서 제안한 Dispatching rule과 MNEH 알고리즘의 테스트 결과를 <표 3>에 정리하였다. 구성 알고리즘 중에서는 MinC/D와 MNEH가 좋은 결과를 보여주었다. MinC/D의 경우는 사격 시간이 적게 걸리면서 위협도 함수값이 가장 큰 표적을 선택하는 방식으로 사격 순서를 작성함으로써, 이웃 탐색 방식의 MNEH 방식과 거의 유사한 수준의 결과를 보여 주었다. 미세한 차이로 MNEH가 MinC/D보다 좋은 결과를 보여주었다.

네 가지 개선 알고리즘을 평가하기 위하여, 초기해는 MNEH로 고정하여 실험한 결과를 <표 4>에 정리하였다. 전체적으로 ICIS가 좋은 결과를 보여주었다. <표 5>에서는 모든 실험 문제에 대하여 제안된 순서결정 알고리즘의 RDI(Relative Deviation Index) 값과 최적 순서를 도출한 개수(NBS)를 정리하였다. 본 연구에서 제안한 알고리즘 중 ICIS가 RDI 및 NBS 지표에 대해서 전반적으로 좋은 결과를 나타내었다.

〈Table 3〉 Performance of the Heuristic Algorithms

W[a]	T[b]	SPT	LTR	LER	LTR/D	MinC	MinC/D	MNEH
3	22	0.783[c](0.222)[d]	0.876(0.137)	0.699(0.177)	0.823(0.156)	0.556(0.249)	0.596(0.230)	0.450(0.144)
	23	0.821 (0.183)	0.775(0.143)	0.834(0.193)	0.772(0.146)	0.514(0.246)	0.649(0.221)	0.425(0.169)
	24	0.819 (0.228)	0.834(0.136)	0.743(0.144)	0.891(0.109)	0.543(0.249)	0.647(0.223)	0.537(0.215)
6	22	0.811 (0.235)	0.806(0.177)	0.712(0.197)	0.658(0.150)	0.520(0.279)	0.300(0.122)	0.298(0.201)
	23	0.933 (0.106)	0.778(0.196)	0.759(0.174)	0.619(0.154)	0.452(0.194)	0.298(0.133)	0.222(0.126)
	24	0.849 (0.179)	0.857(0.181)	0.786(0.167)	0.711(0.145)	0.458(0.269)	0.345(0.166)	0.364(0.267)
9	22	0.815 (0.160)	0.658(0.268)	0.720(0.203)	0.725(0.258)	0.687(0.234)	0.435(0.239)	0.474(0.239)
	23	0.765 (0.248)	0.642(0.275)	0.684(0.275)	0.733(0.208)	0.673(0.265)	0.503(0.174)	0.436(0.260)
	24	0.832 (0.198)	0.596(0.264)	0.760(0.215)	0.655(0.283)	0.587(0.240)	0.459(0.212)	0.464(0.310)

[a] number of weapons
[b] number of targets
[c] average RDI
[d] standard deviation of RDI

〈Table 4〉 Performance of the Improvement Algorithms

W[a]	T[b]	IS	IC	ISIC	ICIS
3	22	0.093(0.065)	0.103(0.057)	0.075(0.063)	0.020(0.026)
	23	0.069(0.045)	0.077(0.053)	0.080(0.060)	0.021(0.032)
	24	0.104(0.051)	0.090(0.065)	0.061(0.052)	0.031(0.042)
6	22	0.098(0.048)	0.057(0.046)	0.061(0.049)	0.017(0.035)
	23	0.106(0.038)	0.064(0.043)	0.051(0.055)	0.016(0.023)
	24	0.098(0.081)	0.081(0.060)	0.076(0.052)	0.031(0.038)
9	22	0.096(0.088)	0.071(0.066)	0.078(0.074)	0.022(0.037)
	23	0.119(0.105)	0.093(0.061)	0.091(0.101)	0.051(0.061)
	24	0.123(0.114)	0.128(0.121)	0.088(0.069)	0.037(0.051)

[a,b,c,d] see footnote of <Table 3>.

〈Table 5〉 Performance of Algorithms for Overall Cases

Methods	Relative Deviation Index		NBS
	Mean	Standard deviation	
LER	0.744	0.196	0
LER/D	0.732	0.199	0
MinC	0.555	0.253	4
MinC/D	0.470	0.231	0
MNEH	0.408	0.234	6
IS	0.101	0.074	18
IC	0.085	0.068	21
ISIC	0.073	0.065	21
ICIS	0.027	0.040	64

5. 결 론

본 연구는 사격 중 아군 위협도를 최소로하는 실시간 사격 순서 결정문제(RFSP; Real time Fire Scheduling Problem)를 다루었다. 아군 무기를 효율적으로 운영하고, 신속하게 적의 움직임에 대응할 수 있도록, 휴리스틱 알고리즘을 개발하였다. 작은 크기의 문제에 대하여 비선형 정수 계획 모델의 최적해와 비교하였고, 큰 크기의 문제에 대해서 Dispatching rule 간에 해의 효율을 비교하였다. 본 연구에서 제안한 구성 알고리즘 중 사격 시간이 적고 위협도 함수값이 큰 표적을 먼저 사격하는 방식이 실시간 사격 문제에서 좋은 결과를 보이는 것으로 나타났다. 추가로 제안한 개선 알고리즘들에서도 위협도가 개선됨을 보였다. 모든 알고리즘에 대해서 아군 포대와 표적이 다수 존재하는 문제에 대해서 0.001초 이내에서 해가 도출 되어, 사격 부대간의 중앙 통제 운영이 가능한 경우, 적 표적을 효율적으로 제거하여 표적의 위협도를 최소화할 수 있는 사격 작전이 가능하다.

참 고 문 헌

[1] 김동현, 이영훈, “표적 할당과 사격 순서의 동시 결정 문제를 위한 발견적 기법”, 「한국경영과학회지」, 제35권, 제1호(2010), pp.47-65.

[2] 윤상흠, 황원식, 전재호, 이익선, “포병부대 사격순서결정을 위한 분지한계 알고리즘 연구”, 「산업경영시스템학회지」, 제33권, 제3호(2010), pp.154-161.

[3] 이대력, 양재환, “혼합정수계획법을 이용한 요격미사일의 할당 및 교전 일정계획에 관한 연구”, 「경영과학」, 제32권, 제4호(2015), pp.109-133.

[4] 정치영, 이재영, 이상현, “격추확률 최대화를 위한 미사일 최적배치 문제”, 「경영과학」, 제27권, 제1호(2010), pp.75-90.

[5] 차영호, 방준영, “최소 종료시간 사격 스케줄을 위한 분지계획법 알고리즘 연구”, 「산업경영시스템학회지」, 제38권, 제4호(2015), pp.132-141.

[6] 최용백, 김경섭, “포병부대 비계획 사격순서 결정 알고리즘”, 「산업경영시스템학회지」, 제35권, 제2호(2012), pp.37-44.

[7] Amoura, A.K., E. Bampis, C. Kenyon, and Y. Manoussakis, "Scheduling independent multiprocessor tasks," *Algorithmica*, Vol.32, No.2 (2002), pp.247-261.

[8] Bianco, L., P. Dell'Olmo, and M.G. Speranza, "Nonpreemptive scheduling of independent tasks with prespecified processor allocations," *Naval Research Logistics,* Vol.41, No.7(1994), pp.959-971.

[9] Blazewicz, J., P. Dell'Olmo, M. Drozdowski, and M. Speranza, "Scheduling multiprocessor tasks on three dedicated processors," *Information Processing Letters*, Vol.41, No.5(1992), pp.275-280.

[10] Bozoki, G. and J.P. Richard, "A branch and bound algorithm for the continuous-process job-shop scheduling problems," *AIIE Transactions,* Vol.2, No.3(1970), pp.246-252.

[11] Brucker, P. and A. Krämer, "Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems," *Euro-*

pean Journal of Operational Research, Vol.90, No.2(1996), pp.214-216.

[12] Chen, J. and A. Miranda, "A polynomial time approximation scheme for general multiprocessor job scheduling," *SIAM Journal on Computing,* Vol.31, No.1(2001), pp.1-17.

[13] Chen, J. and C.-Y. Lee, "General multiprocessor tasks scheduling," *Naval Research Logistics,* Vol.46, No.1(1999), pp.57-74.

[14] Dell'Olmo, P., M.G. Speranza, and Z. Tuza, "Efficiency and effectiveness of normal schedules on three dedicated processors," *Discrete Mathematics,* Vol.164, No.1(1997), pp.67-79.

[15] Drozdowski, M., "Scheduling multiprocessor tasks-An overview," *European Journal of Operational Research,* Vol.94, No.2(1996), pp.215-230.

[16] Garey, M.R. and D.S. Johnson, *Computers and intractability : a guide to the theory of NP-completeness*, San Francisco : Freeman, 1979.

[17] Goemans, M.X., "An approximation algorithm for scheduling on three dedicated machines," *Discrete Applied Mathematics*, Vol.61, No.1 (1995), pp.49-59.

[18] Hall, L.A., *Approximation algorithms for scheduling*, PWS Publishing Company, (1997), pp. 1-45.

[19] Hoogeveen, J.A., S.L. Van de Velde, and B. Veltman, "Complexity of scheduling multiprocessor tasks with precified processor allocations," *Discrete Applied Mathematics* Vol.55, No.3(1994), pp.259-272.

[20] Huang, J., J. Chen, S. Chen, and J. Wang, "A simple linear time approximation algorithm for multi-processor job scheduling on four processors," *Journal of Combinatorial Optimization*, Vol.13(2007), pp.33-45.

[21] Jansen, K. and L. Porkolab, "General multiprocessor task scheduling : approximate solutions in linear tim," *SIAM Journal on Computing*, Vol.35, No.5(2005), pp.519-530.

[22] Kim, D.-H. and Y.-H. Lee, The heuristic algorithm for the fire target allocation and sequencing problem, *Proceedings of the 2008 spring KIIE conference*, Pohang, Korea, 2008.

[23] Kim, T.-H. and Y.-H. Lee, Fire sequencing problem with shared targets, *Korean Operations Research and Management Society*, Vol.28, No.4(2003), pp.123-134.

[24] Krämer, A., "Branch and bound methods for scheduling problems with multiprocessor tasks on dedicated processors," *OR Spektrum,* Vol.19, No.3(1997), pp.219-227.

[25] Kubale, M., "The complexity of scheduling independent two-processor tasks on dedicated processors," *Information Processing Letters*, Vol.24, No.3(1987), pp.141-147.

[26] Kwon, O.-J., K.-S. Lee, and S.-S. Park, "Targeting and scheduling problem for field artillery," *Computers and Industrial Engineering*, Vol.33, No.3-4(1997), pp.693-696.

[27] Lee, C.-Y., L. Lei, and M. Pinedo, "Current trends in deterministic scheduling," *Annals of Operations Research*, Vol.70(1997), pp.1-41.

[28] Nawaz, M., E.E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, Vol.11, No.1 (1983), pp.91-95.