

포병부대 사격순서결정을 위한 분지한계 알고리즘 연구

윤상흠* · 황원식** · 전재호*** · 이익선****†

*영남대학교 경영학부

**육군3사관학교 운영분석학과

***충주대학교 행정학부 행정정보시스템학전공

****동아대학교 경영학부

A Branch-and-Bound Algorithm on the Fire Sequencing for Planned Artillery Operations

Sang Hum Yoon* · Won-Shik Hwang** · Jaeho Juhn*** · Ik Sun Lee****†

*School of Business, Yeungnam University

**Operations Analysis Department, Korea Army Academy

***Division of Public Management Information System, Chungju National University

****School of Business, Dong-A University

This paper considers the simultaneously firing model for the artillery operations. The objective of this paper is to find the optimal fire sequence minimizing the final completion time of the firing missions of multiple artillery units for multiple targets. In the problem analysis, we derive several solution properties to reduce the solution space. Moreover, two lower bounds of objective are derived and tested along with the derived properties within a branch-and-bound scheme. Two efficient heuristic algorithms are also developed. The overall performances of the proposed branch-and-bound and heuristic algorithms are evaluated through various numerical experiments.

Keywords : Fire Sequencing, Artillery Operations, Branch-and-Bound, Heuristic

1. 서 론

국방분야의 다양한 무기체계중 화력은 적에게 투사되는 타격형태를 의미하며 원거리에서 적의 주요 표적을 타격함으로써 적의 전투력을 약화시키는 결정적인 역할을 하게 된다. 최근 이라크 전쟁 등의 사례에서도 확인되었듯이 현대전쟁의 승패는 고지 또는 지역의 확보에 앞서 화력에 의해 적의 전투수행능력과 C4I체계(Command,

Control, Communication, Computer, Intelligence)를 와해시킬 수 있는 능력여부에 의해 좌우된다고 해도 과언이 아니며 향후에도 이러한 화력의 중요성은 더욱 확대될 전망이다[2].

최근 포병 무기체계의 비약적인 발전으로 인해 포병 사격의 화력효과와 정밀도는 지속적으로 향상되고 있다. 예를 들어 K-9 자주포의 경우 3발 TOT(time on target)가 가능하여 화포 1문을 사각과 장약을 달리하여 포

논문접수일 : 2010년 07월 20일 논문수정일 : 2010년 09월 13일 게재확정일 : 2010년 09월 13일

† 교신저자 lis1007@dau.ac.kr

※ 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음.

탄 3발을 발사할 경우 하나의 표적에 대해 동일한 시점에 타격을 가할 수 있으며 이는 과거의 화포 3문이 동시에 사격을 하는 것과 동일한 타격효과를 가지게 되었음을 의미한다. 따라서 타격효과를 동일하게 유지하면서도 운용 무기체계의 수를 줄임으로써 관리대상이 단순화되는 장점이 생긴다.

하지만, 개별 무기체계의 운용모드가 다양해짐에 따라 타격효과가 지휘관의 의사결정에 따라 달라질 수 있는 문제점이 발생하게 된다. 즉, 과거에는 관리대상인 화포의 수는 많으나, 개별적인 화포에 대해 표적이 일단 할당되면 발사와 미발사의 두 가지 의사결정만이 존재하지만, 현재는 하나의 무기체계와 한 개의 표적에 대해서도 여러 개의 운용방식이 존재하게 되는 것이다. 더욱이 단일표적에 단일무기체계가 아니라 실전과 같이 동시에 타격해야 하는 표적의 수가 증가하고 이러한 복수 표적에 대해 타격을 가할 무기체계의 수가 증가하게 되면 무기체계별 표적과 운용방식의 선택, 사격순서 등의 조합에 의해 지휘관이 선택 가능한 의사결정의 수는 지수적으로 증가하게 되고 그 중 가장 신속한 타격을 가능하게 하는 조합을 선택하는 것은 매우 복잡하고 어려운 일이다.

이와 같이 화력성능을 극대화하기 위해서는 표적을 정확하게 탐지하고 명중할 수 있는 무기체계의 하드웨어적인 능력뿐 아니라 가용한 화력들을 어떤 조합으로 어떤 표적에 대해 어떠한 타격절차(탐지-결심-타격)에 따라 타격을 가할 것인가를 결정하는 운용방식에 대한 의사결정이 매우 중요하다 하겠다. 더욱이 이러한 의사결정이 지휘관의 개인능력에 의해 좌우되는 것이 아니라 무기체계에 자동화된 형태로 내재될 경우 그 효과는 극대화 될 것이다.

이러한 포병 무기체계운용방식에 주요한 부분을 차지하는 것이 사격모형이다. 사격모형은 크게 표적할당(targeting problem)와 사격순서결정(fire sequencing problem)의 두가지 의사결정을 포함하고 있다. 이 중 표적할당은 각 부대별로 어떤 표적에 대해 사격을 가할 것인가를 결정하는 문제이며 표적할당의 결과 결정된 복수의 사격임무들을 어떤 순서로 실시할 것인가를 결정하는 것이 사격순서결정이다. 표적할당과 사격순서결정은 통합하여 동시에 결정하는 것이 이상적이나 의사결정의 복잡성과 해법의 부재로 인해 두 개를 단계별로 구분하여 결정하는 것이 일반적이다. Kwon et al.[7]의 연구에서도 언급된 바와 같이 표적할당은 과거부터 매우 많은 연구가 진행되어 왔으나 사격순서결정에 대한 연구는 상대적으로 미진한 상태이다. 그러나, 전시가 아닌 평시의 훈련과정에서는 표적할당을 위한 충분한 시간이 사전에 주어지며 계획표적의 형태로 훈련전에 정해지는

경우가 대부분이다. 더욱이 정해진 계획표적에 대한 사격순서결정은 그 자체의 중요성뿐 아니라 일상으로 반복되는 사격훈련에서 훈련운영비용을 줄인다는 측면에서 그 중요성이 적다고 할 수 없다.

따라서, 본 연구에서는 포병부대의 사격순서결정문제를 고려하고 있으며 이에 대한 효과적인 해법을 제시하는 것을 목적으로 하고 있다. 사격순서결정문제는 복수의 표적을 복수의 무기체계를 통해 타격을 가할 경우 전체 발사임무를 모두 완료하기 위한 총 소요시간을 최소화하기 위한 사격순서를 결정하는 의사결정모형으로 정의될 수 있다. 여기서 무기체계란 한 대의 화포 또는 여러 대의 화포를 포함하는 부대를 의미하게 된다. 사격순서결정문제에는 실제 포병사격운영의 현실적인 요구사항을 그대로 반영하여 다음과 같은 몇 가지 제약이 존재한다.

- (1) 한 부대는 동일시점에 하나의 표적에 대해서만 사격이 가능하다.
- (2) 일단 사격이 시작되면 사격임무가 완료될 때까지 중간에 중단될 수 없다.
- (3) 동일 표적에 대해 여러 부대에 사격임무가 주어진 경우 해당 표적에 대한 사격들의 시작시점은 동일하다. 그러나, 각 부대의 사격임무가 다르기 때문에 종료되는 시점은 다를 수 있다.

특히, 위의 조건들 중 세 번째는 표적에 대한 타격효과를 극대화하기 위한 것으로 그동안 수많은 연구가 진행된 민간 산업현장에서의 병렬기계일정계획(parallel machine scheduling)과 동시자원 일정계획[5]의 연구결과를 그대로 사용할 수 없도록 만드는 사격순서결정문제만의 고유한 제약이다.

Kwon et al.[7, 8]은 사격모형에 대한 기존연구들을 상세히 기술하고 있으며 각 문제에 대한 효과적인 수리계획(mathematical formulation)들을 제안하였다. 또한, 표적할당에 대한 라그랑지안 접근법과 사격순서결정을 위한 휴리스틱(heuristic algorithm)을 최초로 제안하였다. 김태현과 이영훈[1]은 사격순서결정을 위한 페어알고리즘이란 이름의 새로운 휴리스틱을 제안하였으며, Kwon[6]은 유전알고리즘을 통한 해법을 제안한 바 있다. 최근에는 동시사격 모델 뿐 아니라 개별사격, 혼합사격의 보다 복잡한 현실적인 상황을 고려한 사격모형들도 제안되고 있다[3, 4].

본 연구의 대상 문제는 Strongly NP-complete 영역의 문제(Lee et al.[9]에 의해 증명됨)로써, 표적수와 부대수의 증가에 따라 문제의 복잡도가 급격히 증가하게 된다. 따라서 본 연구에서는 작은 사이즈의 문제에 대한 최적해를 효율적으로 도출할 수 있는 분지한계법(branch-and-bound)과 큰 사이즈의 문제에 대해서는 빠른 시간 안에

효과적인 해를 제공하는 휴리스틱을 함께 제시한다.

본 연구의 구성은 다음과 같다. 제 2장에서는 본 연구에서 고려하고 있는 동시사격모형에 대한 엄격한 정의와 함께 수리계획모델이 제시되었다. 제 3장에서는 문제에 대한 분석을 통해 몇 가지 우월성질들을 제시하고, 목적함수에 대한 2가지 하한값(lower bound)을 활용한 분지한계법과 효율적인 해를 제공할 수 있는 2가지 휴리스틱들이 제안된다. 제 4장에서는 다양한 수치실험을 통해 제안된 해법들의 성능을 평가하였다. 마지막으로 제 5장에서는 연구의 기여점과 추후연구과제에 대한 논의가 이루어진다.

2. 문제정의와 수리모형

포병사격은 일반적으로 표적정보수집 → 타격요망효과 결정 → 사격발수결정 → 사격순서결정 → 사격의 순서로 진행된다. 그 중 사격발수결정은 지휘관이 결정한 표적에 대해 사전에 결정한 표적타격요망효과에 맞추어 적에게 사격이 가능한 각 포병부대의 화포특성을 고려하여 사격발수를 결정하는 것이다. 이러한 사격발수는 피해요망효과를 고려하여 JMEM(joint munition effectiveness manual)이란 합동탄약효과 교범을 이용하여 결정한다. JMEM에 의해 사격발수가 결정되면 아군의 화기제원표에 기록된 곡사포의 지속 발사속도를 확인하여 이를 시간으로 변환하여 각각 표적에 대해 사격수행시간을 알 수 있다.

이러한 현실적인 상황을 고려하여 본 연구의 대상이 되는 동시사격모형을 사용할 수리적 기호와 함께 엄격하게 다시 정의하면 다음과 같다; 타격해야 할 총 n 개의 계획표적이 존재하며 이들 표적에 대해 m 개의 사격부대가 사격 임무를 수행한다. 표적 $i(i=1,2,\dots,n)$ 에 대한 부대 $j(j=1,2,\dots,m)$ 의 사격수행시간이 p_{ij} 로 사전에 지정되어 있다. 동시사격모형에서는 계획표적에 대한 사격운용에 있어 다음과 같은 몇 가지 현실적인 제약을 가지게 된다.

- (1) 각 표적에 대한 사격은 동시에 시작된다. 즉, 표적 i 에 대한 사격수행시간이 $p_{ij} > 0$ 인 모든 부대들은 해당 표적에 대한 사격을 동시에 시작한다.
- (2) 일단 사격이 시작되면 중간에 중단될 수 없으며 p_{ij} 의 사격수행시간이 지난 후에 사격이 종료된다.
- (3) 특정 표적 i 와 부대 j 간의 사격수행시간이 $p_{ij} = 0$ 인 경우에는 부대 j 를 제외하고 나머지 부대들($p_{ik} > 0$, $p_{ik} > 0$ 인 모든 k)이 동시사격을 수행한다.

이러한 제반 환경에서 동시사격모형의 목적은 모든

계획표적에 대한 사격임무를 완수하기 위한 총 사격완료시간을 최소화할 수 있는 계획표적에 대한 최적의 사격순서를 결정하는 것이다.

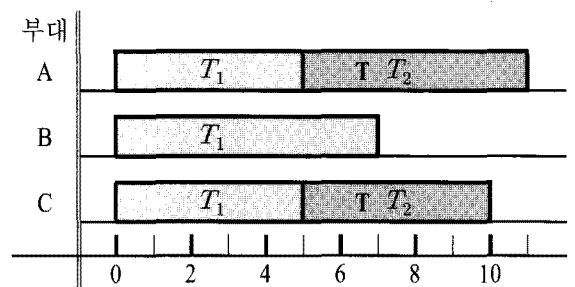
동시사격모형에 대한 보다 정확한 이해를 돕기 위한 예제는 다음과 같다.

예제 1 : 두 개의 표적을 세 개의 부대가 동시사격한다. 각 표적에 대한 부대별 사격수행시간은 다음 표와 같이 계획되어 있다.

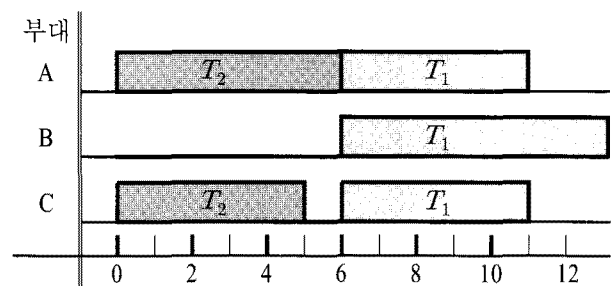
<표 1> 동시사격모형의 사격수행시간의 예

표적	T_1			T_2		
	A	B	C	A	B	C
사격부대						
사격수행 시간(분)	5	7	5	6	•	5

가능한 사격순서는 T_1T_2 또는 T_2T_1 의 두 가지 경우이며 T_1T_2 의 사격순서의 총 완료시간은 다음의 <그림 1-a>에서 알 수 있듯이 11분이 된다. 이에 비해 만약 사격순서가 T_2T_1 의 순서로 수행된다면 <그림 1-b>에 의해 13분이 되어 위 예제의 최적해는 T_1T_2 의 사격순서가 된다.



a : T_1T_2 의 사격순서인 경우



b : T_2T_1 의 사격순서인 경우

<그림 1> 동시사격모형의 총 완료시간

동시사격모형은 다음과 같은 정수계획법(Mixed Integer

Programming)으로 표현될 수 있다.

Min C_{max}
subject to

$$C_{max} \geq tx_{ijt} + p_{ij} \text{ for all } i, j, t$$

$$\sum_{t=1}^T z_{it} = 1 \text{ for all } i$$

$$\sum_{t=1}^T x_{ijt} = 1 \text{ for all } i, j$$

$$x_{ijt} \geq z_{it} \text{ for all } i, j, t$$

$$\sum_{t=1}^T tx_{ijt} \geq \sum_{t=1}^T tx_{ikt} + p_{ik} - My_{ik} \text{ for all } i, j \neq k$$

$$\sum_{m=1}^n y_{mi} \geq \sum_{m=1}^n y_{mk} - My_{ik} \text{ for all } i \neq k$$

$$y_{ik} + y_{ki} = 1 \text{ for all } i \neq k$$

$$x_{ijt}, z_{it}, y_{ik} \in \{0, 1\} \text{ for all } i \neq k, j, t$$

위의 수리모형에서 사용된 변수에 대한 설명은 다음과 같다.

- x_{ijt} : 표적 i 에 대한 j 부대의 사격이 시점 t 에서 시작되면 값은 1, 그렇지 않으면 0,
- y_{ik} : 표적 i 를 표적 k 보다 먼저 타격하는 경우 값은 1, 그렇지 않으면 0,
- z_{it} : 표적 i 에 대한 동시사격을 시점 t 에 시작하면 값은 1, 그렇지 않으면 0

위에서 제시한 수리계획은 기존연구에서 제시된 수리모형[1, 3, 8]들과는 다른 것이며 비록 제약식과 변수의 수가 많아 그 자체로써 강점을 가지기는 힘드나 라그랑지안과 같은 수리적 접근법에 유리한 구조를 가지고 있으며 사격순서결정문제에 대한 정확한 이해를 도울 수 있는 구조를 가지고 있다.

3. 분지한계법(Branch-and-Bound Algorithm)

3.1 분지규칙(branching rule)

본 연구에서 제안하는 분지한계법의 각 노드는 전체 사격순서중에 부분순열(partial sequence)에 대응된다. 분지단계에서는 분지할 하나의 노드(부분순열)를 선택하고 이 부분순열에 포함되지 않은 표적들 중 한 개를 부분순열의 뒤에 붙임으로써 새로운 노드(확장된 부분순열)가 생성된다.

분지한계트리에서의 탐색은 깊이우선탐색(depth-first search)을 사용한다. 이는 탐색트리에서 가장 큰 부분순열을 가진 노드가 분지노드로 선택됨을 의미한다. 이러한 선택이 여러 노드에서 발생할 경우에는 그중에서 하한 값이 가장 작은 노드가 선택된다.

3.2 노드제거규칙(fathoming rules)

노드제거규칙은 다음에서 제시되는 몇 가지 성질들의 조건이 성립할 경우 해당되는 노드들이 더 이상 탐색될 필요가 없이 제거될 수 있음을 의미한다. 본 연구에서는 다음에 제시될 세 가지 노드제거규칙을 사용한다. 이에 대한 원활한 설명을 위해 다음과 같은 기호를 사용한다.

- π : 현재 노드에서 이미 사격순서가 결정된 표적들의 부분순열,
- n_π : π 에 포함된 표적의 수,
- $C_j(\pi)$: π 에 대한 부대 j 의 사격완료시간,
- $\tilde{\pi}$: π 에 속하지 않는 나머지 표적들의 집합,
- s_{ij} : 표적 i 에 대한 부대 j 의 사격 시작시간,
- f_{ij} : 표적 i 에 대한 부대 j 의 사격 완료시간,
- Ω : 모든 부대의 사격수행시간이 0보다 큰 표적들의 집합, 즉 $\Omega = \{i : p_{ij} > 0, 1 \leq j \leq m\}$.

첫 번째 성질은 총 n 개의 표적간의 모든 가능한 순열 $n!$ 개를 모두 고려할 필요가 없이 Ω 에 속한 표적들을 중심으로 표적들을 그룹화하고 그룹간의 순열만을 고려하면 된다는 것을 보여준다.

성질 1 : 임의의 사격순서 S 에서 S 에 속한 런간에 순서를 바꾸는 것은 총 사격완료시간에 변화를 주지 않는다.

증명 : 증명의 편의를 위해 어떤 사격순서에서 두 개의 연속된 Ω 표적 사이에 위치한 부분순열을 런(run)이라고 정의하고 런간의 순서를 고려하여 i 번째 런을 R_i 로 표시한다. 예를 들어, 6개의 표적에 대한 어떤 사격순서가 $S = T_3T_1T_5T_2T_6T_4$ 이고, Ω 표적이 $\Omega = \{T_2, T_3, T_4\}$ 라면 3개의 런($R_1 = T_3T_1T_5$, $R_2 = T_2T_6$, $R_3 = T_4$)이 존재하게 되고 전체 사격순서는 $S = R_1R_2R_3$ 가 된다. 성질 4는 런단위로 사격순서를 바꾸는 경우에는 전체 사격완료시간에 변함없음을 의미한다. 예를 들어 위에서 S 에서 R_1 , R_2 와 R_3 에 대해서 어떠한 방식으로 위치를 바꾸더라도, 기존의 S 와 동일한 사격완료시간을 가지게 된다. 예를 들어, 새로운 사격순서 $R_3R_2R_1$ 은 기존 S 와 동일한 사격완료시간을 가진다.

위의 성질 1에 의해 다음과 같이 분지한계법에서 활용할 수 있는 첫 번째 노드제거규칙이 제안될 수 있다.

부속성질 1 : 분지한계트리에서 부분순열 π 에 대한 노드에서 아직 순서가 정해지지 않은 $\bar{\pi}$ 에 속한 모든 표적들이 Ω 표적인 경우에 $\bar{\pi}$ 에 속한 표적들은 임의의 순서로 사격해도 사격완료시간은 동일하다.

또한, 성질 1에 의해 다음의 부속성질 2의 조건이 만족하는 경우에는 분지한계의 최초노드에서 해를 찾을 수 있다.

부속성질 2 : 모든 표적이 Ω 표적인 경우에는 모든 사격순서의 사격완료시간은 동일하다.

증명 : 위의 성질 1에 의해 모든 표적이 Ω 표적인 경우에는 각 표적에 대한 모든 사격이 동시에 시작되는 특성에 의해 임의로 선택된 사격순서에 대한 총 완료시간은 $C_{\max} = \sum_{i=1}^n \max_{1 \leq j \leq m} \{p_{ij}\}$ 으로 상수이다.

다음은 특정 부대의 사격이 다른 부대의 사격시간에 비해 항상 클 경우에 대한 특수한 경우에 대한 성질이다.

성질 2 : 부분사격순열 π 뒤에 연속하여 사격되는 두 개의 표적 i 와 k 가 있을 때 모든 부대 j 에 대해 다음의 조건을 만족할 경우 i 를 k 보다 먼저 사격하는 것이 그 반대의 경우보다 우월하다.

$$C_j(\pi ik) \leq C_j(\pi ki)$$

증명 : 인접작업교환(pair-wise job interchange)방법으로 증명이 가능하다.

3.3 한계규칙(bounding rule)

한계규칙(bounding rule)은 각 노드의 상한과 하한을 이용하여 현재 탐색중인 노드의 제거 여부를 결정하고, 해의 수렴을 촉진시킨다. 본 연구에서는 초기 상한값으로 제 3.4절에서 제시될 두 가지 휴리스틱들의 목적함수 값을 구하고 그 중에서 해가 더 우수한 것을 사용한다. 즉, UB_1 와 UB_2 를 각각 휴리스틱 H1과 H2를 활용하여 얻은 목적함수값이라고 할 때, 초기노드의 상한값 UB 는 다음과 같다.

$$UB = \min\{UB_1, UB_2\}$$

또한, 이후의 탐색에서 더 좋은 가능해(feasible solution)가 얻어지면 상한 값은 계속 갱신된다.

분지한계 트리의 각 노드에는 부분순열 π 가 일대일로 대응되며, 각 노드에 대해 두 개의 하한값 LB_1 과 LB_2 를 사용한다. 첫 번째 하한값 LB_1 은 다음과 같이 표현된다.

$$LB1_1 = \max_{1 \leq j \leq m} \left\{ C_j(\pi) + \sum_{i \in \bar{\pi}} p_{ij} \right\}$$

또한, 두 번째 하한값 LB_2 는 다음과 같다.

$$LB_2 = C_{\max}(\pi) + \sum_{i \in \Omega_{\pi}} \max_{1 \leq j \leq m} \{p_{ij}\} + \max_{1 \leq j \leq m} \left\{ 0, \sum_{i \in \bar{\Omega}_{\pi}} p_{ij} - I_j \right\}$$

여기서, Ω_{π} 는 π 에 속하지 않는 나머지 표적들 중 Ω 표적들의 집합을 의미하며, $\bar{\Omega}_{\pi}$ 는 Ω 표적이 아닌 표적들의 집합을 나타낸다. 또한, I_j 는 $\bar{\Omega}_{\pi}$ 에 속한 표적들에 대한 부대 j 에서의 총 유휴시간(idle time)의 하한값을 의미하며, 다음과 같이 표현된다.

$$I_j = C_{\max} - C_j(\pi) + \sum_{i \in \bar{\Omega}_{\pi}} \{ \max_{1 \leq k \leq m} (p_{ik}) - p_{ij} \}$$

본 연구의 분지한계법에서는 분지한계 트리의 각 노드에서 도출된 두 가지 하한 LB_1 , LB_2 를 계산하여 보다 우수한 것을 각 노드별 최종 하한으로 사용한다. 즉 각 노드에서의 최종 하한값은 다음과 같다.

$$LB = \max\{LB_1, LB_2\}$$

3.4 휴리스틱(heuristic algorithm)

본 절에서는 그리디(Greedy) 타입의 두 가지 휴리스틱 H1, H2를 제안한다. 첫 번째 휴리스틱은 할당되지 않은 표적들 중에서 사격완료시간의 증가분을 최소화시켜주는 표적들을 우선적으로 할당해나가는 방식의 알고리즘이다.

휴리스틱 H1

단계 1 : 현재까지 할당된 표적들의 집합을 σ 이라 한다.

단계 2 : σ 에 속하지 않는 표적들을 대상으로 $T_z = \max_j C_j(\sigma z)$ 값을 계산한다. 단, $C_j(\sigma z)$ 는 일정 σ

의 바로 뒤에 표적 z 를 할당하는 경우의 부대 j 에서의 사격완료시간을 의미한다.

단계 3 : 가장 작은 T_z 값을 가지는 표적 z 를 할당하고, 사격완료시간을 업데이트하고, 단계 2로 간다.

본 연구가 제시하는 두 번째 그리디 휴리스틱은 H1 휴리스틱과 유사하지만, T_z 값을 계산하는 방식에서 차이점을 가진다. H1 휴리스틱은 표적들의 최종완료시간을 최소화시키는 것이 주관심사이지만, H2는 표적들을 처리해나가는 과정에서 발생하는 유휴시간(idle time)을 줄이는 것이 주관심사이다. 따라서, 어떤 표적을 할당할 때, 부대별로 사격완료의 최대시간값과 최소시간값의 차이를 최소화되는 표적들을 우선적으로 처리하도록 한다.

휴리스틱 H2

단계 1 : 현재까지 할당된 표적들의 집합을 σ 이라 한다.
 단계 2 : σ 에 속하지 않는 표적들을 대상으로 다음의 값을 계산한다.

$$T_z = \max_j \{C_j(\sigma z)\} - \min_j \{C_j(\sigma z)\}$$

단계 3 : 가장 작은 T_z 값의 표적 z 를 할당하고, 완료시간의 값들을 업데이트하고, 단계 2로 간다.

4. 성능 평가

본 장에서는 제 2장에서 소개된 성질들과, 제 3장에서 제안된 분지한계 알고리즘, 제 4장에서 소개된 휴리스틱들의 성능을 평가하고 분석한다. 먼저 각 목표에 대한 사격시간(p_{ij} , $i=1, \dots, n$, $j=1, \dots, m$)는 $U(1, 100)$ 으로부터 발생시킨다. 여기서, $U(a,b)$ 는 a 와 b 를 모수로 가지는 이산형 균등분포(discrete uniform distribution)를 의미한다.

본 연구는 p_{ij} 들 중 일부는 0의 값을 가질 수 있음을 감안하고 있다. 기호 ρ 를 전체에서 양수 값을 가지는 p_{ij} 들의 비율이라고 한다면, 분지한계 알고리즘과 휴리스틱들의 성능은 ρ 값에 영향을 받을 것으로 예상된다. 따라서, 본 연구는 알고리즘들의 성능분석을 위해서 ρ 값으로 0.6, 0.7, 0.8, 0.9의 네 가지 값을 고려하고 있다. 본 수치실험은 표적의 수(n)를 9종류(6, 8, 10, 12, 14, 16, 18, 20, 22), 부대의 수(m)를 3종류(2, 3, 4)를 대상으로 하고, ρ 값으로 4종류를 고려하여 총 108가지의 문제 조합에 대해 각각 20문제씩, 총 2,160문제를 임의로 생성하였다. 실제 군사문제에서는 부대의 수는 4개를

초과하는 경우는 거의 일어나지 않는다고 한다. 따라서, 본 실험에서 4개 이하의 부대의 수를 고려하는 것은 타당하다고 말할 수 있다.

먼저 분지한계 알고리즘의 성능을 평가한다. ρ 값이 0.6, 0.7, 0.8, 0.9에 대해서 성능평가의 결과요약은 <표 2>~<표 5>에 각각 정리되어 있다. 각 표에서 npl 은 20개 문제들 중에서 1,000초 이내에 최적해를 찾아낸 문제들의 개수를 나타내고 있다. 시간 내에 최적해를 찾아낸 문제들을 대상으로 분지한계 알고리즘의 평균 수행시간과 최대 수행시간, 평균 탐색노드 수, 최대 탐색노드 수를 제시하고 있다. (*)표시는 1000초(CPU)안에 문제를 풀 수 없는 문제들이 생겼기 때문에, 통계치를 얻을 수 없었음을 의미한다.

<표 2>~<표 5>까지의 분지한계 알고리즘의 성능 테스트 결과를 보면, ρ 값이 0.6이고 부대수가 2인 경우에는 20개까지 타겟들의 경우에 모두 최적해를 찾음을 알 수 있었다. 부대의 수가 4개로 증가되면 알고리즘의 성능은 떨어져서, 14개 정도까지 최적해를 찾을 수 있었다. 또한 실험결과표를 보면, ρ 의 값이 작을수록 알고리즘의 성능은 좋아짐을 알 수 있다. 또한, 부대의 수가 작을수록 성능이 더 좋아짐을 알 수 있다.

<표 2> 분지한계알고리즘의 성능테스트($\rho = 0.6$)

부대수	표적수	계산시간		탐색 노드 수		npl
		aver.	max.	aver.	max.	
2	6	0.01	0.01	6	35	20
	8	0.01	0.01	12	209	20
	10	0.01	0.01	20	173	20
	12	0.01	0.08	755	9933	20
	14	0.01	0.01	167	1900	20
	16	0.00	0.01	199	1053	20
	18	0.95	6.67	19159	2149617	20
	20	13.05	162.06	112247	9684535	20
	22	(*)	(*)	(*)	(*)	15
3	6	0.01	0.01	39	116	20
	8	0.01	0.01	246	1022	20
	10	0.02	0.11	1878	11540	20
	12	0.29	1.22	26736	109157	20
	14	17.70	181.17	1371888	14446197	20
	16	77.16	545.02	5992151	86304055	20
	18	(*)	(*)	(*)	(*)	6
4	6	0.01	0.01	88	152	20
	8	0.02	0.03	1063	4320	20
	10	0.08	0.34	7434	36546	20
	12	2.38	14.38	195131	1247995	20
	14	39.48	302.66	2526006	21998894	20
	16	(*)	(*)	(*)	(*)	12

<표 3> 분지한계알고리즘의 성능테스트($\rho = 0.7$)

부대수	표적수	계산시간		탐색 노드 수		npl
		aver.	max.	aver.	max.	
2	6	0.00	0.01	10	48	20
	8	0.01	0.01	76	296	20
	10	0.01	0.03	384	3074	20
	12	0.04	0.19	4039	23541	20
	14	0.52	5.25	50101	528369	20
	16	8.44	131.16	778145	11961400	20
	18	(*)	(*)	(*)	(*)	16
3	6	0.00	0.01	61	247	20
	8	0.01	0.03	881	3360	20
	10	0.16	1.00	18627	127683	20
	12	3.25	16.42	313522	1759957	20
	14	71.30	208.30	5382416	18318395	20
	16	(*)	(*)	(*)	(*)	10
4	6	0.01	0.01	111	183	20
	8	0.02	0.05	1906	4787	20
	10	0.31	0.73	28103	75998	20
	12	14.31	49.81	1140321	4654349	20
	14	(*)	(*)	(*)	(*)	18

<표 5> 분지한계알고리즘의 성능테스트($\rho = 0.9$)

부대수	표적수	계산시간		탐색 노드 수		npl
		aver.	max.	aver.	max.	
2	6	0.00	0.01	16	52	20
	8	0.01	0.01	44	119	20
	10	0.01	0.03	497	3117	20
	12	0.05	0.17	6462	22316	20
	14	0.50	1.76	62249	229671	20
	16	28.27	118.63	3034029	12816353	20
	18	(*)	(*)	(*)	(*)	17
3	6	0.01	0.01	24	56	20
	8	0.01	0.01	295	478	20
	10	0.02	0.05	1574	4533	20
	12	0.51	1.25	57054	151930	20
	14	29.66	132.83	2805561	12371282	20
	16	(*)	(*)	(*)	(*)	15
4	6	0.01	0.01	60	103	20
	8	0.01	0.02	867	1884	20
	10	0.08	0.20	9990	24882	20
	12	2.26	8.89	230074	850146	20
	14	155.37	789.22	12516724	64360399	20
16	(*)	(*)	(*)	(*)	3	

<표 4> 분지한계알고리즘의 성능테스트($\rho = 0.8$)

부대수	표적수	계산시간		탐색 노드 수		npl
		aver.	max.	aver.	max.	
2	6	0.00	0.01	12	39	20
	8	0.01	0.01	95	299	20
	10	0.02	0.05	1298	4188	20
	12	0.32	1.16	38436	148517	20
	14	8.24	30.56	835519	3246000	20
	16	198.25	561.11	17162855	50583400	20
	18	(*)	(*)	(*)	(*)	10
3	6	0.01	0.01	54	134	20
	8	0.01	0.03	722	2922	20
	10	0.11	0.41	13543	49013	20
	12	4.50	18.77	430316	1983864	20
	14	217.50	780.92	18117103	87854019	20
	16	(*)	(*)	(*)	(*)	7
4	6	0.01	0.01	100	158	20
	8	0.02	0.03	1685	3510	20
	10	0.46	1.30	46537	150443	20
	12	13.08	38.47	1066226	3499739	20
	14	(*)	(*)	(*)	(*)	10

<표 6> 노드제거규칙 활용하지 않은 경우($\rho = 0.6$)

부대수	표적수	계산시간		탐색 노드 수		npl
		aver.	max.	aver.	max.	
2	6	0.00	0.03	11	72	20
	8	0.01	0.03	245	1741	20
	10	0.03	0.22	3440	25784	20
	12	0.77	6.47	75688	663899	20
	14	42.52	414.38	3860810	39361807	20
	16	163.42	673.84	13718625	91716814	18
	3	6	0.00	0.01	105	411
8		0.02	0.05	2363	8685	20
10		0.61	4.03	76333	538953	20
12		23.26	120.06	2294454	13114007	20
14		467.55	717.38	34122527	83552565	15
4		6	0.01	0.01	170	344
	8	0.03	0.09	4347	11993	20
	10	0.69	2.56	71418	326830	20
	12	37.82	139.76	3116292	12091902	20
	14	669.89	913.94	41511397	78490955	10

이제는 3.3점에서 제시된 노드제거규칙들이 분지한계 알고리즘의 성능에 얼마나 큰 도움을 주었는지 평가하기 위한 실험을 실시하였다. <표 6>는 노드제거규칙들을 적용하지 않았을 경우의 분지한계 알고리즘의 성능평가 결과를 보여주고 있다. 앞의 <표 2>의 결과와 비교해 볼 때 성질들이 분지한계 알고리즘의 성능을 향상시켜준다고 말할 수 있다.

다음으로 제 3.4절에서 제안된 휴리스틱들의 성능평가를 위한 수치실험을 수행하였다. 결과는 <표 7>에 요약되어 있다. <표 7>에서 $Gap(\%) = \frac{(Heu - Opt)}{Opt} \times 100$ 으로 계산된다. 여기서, Heu는 해당 휴리스틱 알고리즘을 통해 구해진 해의 목적식 값을 의미한다. 또한, "no"는 해당 휴리스틱이 최적해를 발견한 횟수를 의미한다(20개의 실험문제 중에서).

성능분석결과 휴리스틱 H2가 보다 우수한 성능을 보임을 알 수 있었다. 또한 ρ 값이 증가할수록 휴리스틱 알고리즘의 성능은 저하되는 것을 확인할 수 있었다. 휴리스틱 H2는 평균적으로 5% 안팎의 오차를 가지는 것으로 보이는데, 이는 매우 우수한 성능을 가지는 것으로 결론 내려질 수 있다.

<표 7> 휴리스틱 알고리즘 성능평가(부대수 = 2)

ρ	표적수	H1		H2	
		Aver. GAP(%)	no	Aver. GAP(%)	no
0.6	6	8.12	8	3.29	12
	8	9.11	4	4.68	5
	10	8.14	3	4.15	3
	12	10.24	2	6.53	2
	14	15.29	0	8.68	0
0.7	6	4.33	11	1.89	12
	8	6.29	7	3.32	12
	10	6.45	6	2.42	9
	12	9.13	4	5.48	5
	14	4.67	8	1.96	7
0.8	6	10.53	2	6.59	4
	8	9.98	3	6.72	3
	10	10.50	0	5.01	3
	12	10.44	0	7.85	1
	14	10.82	0	7.29	0

5. 결 론

사격순서결정문제에 대한 최적화 기법을 활용한 연구는 단순히 수리계획의 제안[1, 8, 9]과 ILOG Cplex와 같은 상용패키지를 통한 접근[1, 3]에 국한되어 있었다. 따라서, 본 연구에서 제시한 분지한계법을 활용하면 보다 큰 사이즈(부대수-표적수)의 문제에 대한 최적화된 해를 찾게 될 뿐 아니라 향후에도 진행될 다양한 발견적 해법에 대한 성능비교의 도구로도 활용될 수 있을 것이다.

또한, 본 연구결과는 동시사격상황을 확장한 개별사격이나 혼합사격과 같은 보다 현실적인 복잡성을 가미한 다양한 사격모형에 대한 연구에도 도움이 될 것으로 사료된다.

참고문헌

- [1] 김태현, 이영훈; "공유표적을 포함한 사격순서 결정에 관한 연구", 한국경영과학회지, 28(3) : 123-134, 2003.
- [2] 정금연, 백귀철; "화력계획 교리개선", 육군교육사령부 교리개선, 2008.
- [3] 황원식, 이재영; "스케줄링을 이용한 계획표적 사격순서의 최적화에 관한 연구", 한국국방경영분석학회지, 33(1) : 105-115, 2007.
- [4] 황원식, 천윤환, 박현수, 윤상홍; "포병부대 계획표적 사격시간 단축을 위한 표적할당에 관한 연구", 한국국방경영분석학회지, 36(1) : 15-28, 2010.
- [5] Dobson, G., Karmarkar, U.; "Simultaneous Resource Scheduling to Minimize Weighted Flow Times," *Operations Research*, 37 : 592-600, 1989.
- [6] Kwon, O. J.; "A Genetic Algorithm Approach to the Fire Sequencing Problem," *Journal of the Military Operations Research Society of Korea*, 29(2) : 61-80, 2003.
- [7] Kwon, O. J., Kang, D. H., Lee K. S., and Park, S. S.; "Lagrangian Relaxation Approach to the Targeting Problem," *Naval Research Logistics*, 46 : 640-653, 1999.
- [8] Kwon, O. J., Lee K. S., and Park, S. S.; "Targeting and Scheduling Problem for Field Artillery," *Computers and Industrial Engineering*, 33 : 693-696, 1997.
- [9] Lee K. S., Kwon, O. J., and Park, S. S.; "Complexity of the Fire Sequencing Problem," *International Journal of Management Science*, 5 : 55-59, 1999.