

## 공유표적을 포함한 사격순서 결정에 관한 연구

김태헌\* · 이영훈\*\*

### Fire Sequencing Problem with Shared Targets

Tae Heon Kim\* · Young Hoon Lee\*\*

#### ■ Abstract ■

Fire Sequencing Problem (FSP) is to find a sequence of targets, where there exist a number of targets with different time units required to fire. Because of the weapon's specifications and the size of the targets, several weapons may fire on the same targets, and the time units required on firing for each weapon may be different from each other. The objective is to minimize the completion time of firing for given number of targets. Mathematical formulation is given, and the heuristic algorithm based on the pairing of targets in advance is suggested. Performance of the heuristic is evaluated by comparison of heuristic appeared in the literature through the computational experiments.

Keyword : Fire Sequencing Problem, Simultaneous Processing, Pairing

### 1. 서 론

현대전은 대량의 화력, 우수한 기동력 및 이에 대한 과학적인 운용이 요구된다. 걸프전과 같은 최근의 전례에서 입증된 것처럼 대량의 화력전으로 특징되는 현대전에서 전승을 보장하기 위해서는 우수한 화력과 이에 대한 효율적인 운용이 요구되

며, 이를 위해 각기 다른 무기체계까지 포함한 통합 화력운용의 중요성이 증가되고 있다[3, 4]. 화력전에서 대표적인 전투의 방식은 사격을 통하여 이루어진다. 사격은 작전개시 전에 미리 계획하여 수행하게 되는 경우 표적별 사격 필요량에 따라 사격 부대와 사격량을 할당하고 할당된 결과에 따라 사격순서를 결정하며 이에 필요한 전체사격시간을

논문접수일 : 2002년 12월 31일 논문게재확정일 : 2003년 9월 1일

\* 연세대학교 컴퓨터과학·산업시스템공학과

\*\* 연세대학교 정보대학원 부교수

최소화함으로써 사격의 효과를 높일 수 있다[2, 5, 9]. 전체 사격시간을 최소화 하는 것은 동일한 사격량을 장시간에 걸쳐 사격하는 것보다 단시간에 완료함으로써 적이 회복하거나 반격할 여지를 최소화하여 전투의 효과를 증대시키는 결과를 가져올 수 있다. 계획된 사격을 수행할 때의 표적은 하나의 사격부대가 단독으로 공격하는 단일표적과 여러 부대가 동시에 한 표적에 공격하는 공유표적의 두 종류가 있다. 공유표적의 경우 사격개시는 동시에 시작하지만 부대마다 상이한 무기체계 등의 이유로 사격에 소요되는 시간이 다를 수 있어 사격 완료시간은 각 부대별로 다를 수 있다. 본 연구는 여러 부대가 다종의 표적에 대해 사격을 수행할 때 발생하는 사격순서 결정문제(Fire Sequencing Problem)에 대한 것으로 다음과 같이 정의한다.

- 사격부대의 수와 사격목표인 표적 모두 여러 개이다.
- 사격부대는 한번에 하나의 표적에만 사격할 수 있다.
- 사격부대별로 사격을 수행할 표적이 할당되어 있고 표적별 소요시간도 정해져 있다. 사격 소요시간은 정수(Integer)이다.
- 동일한 표적에 대해 여러 개의 부대가 사격하도록 할당되어 있는 경우 동일 표적에 대한 사격 개시시간은 동일하다.
- 목적함수는 할당된 표적에 대한 사격완료시간의 최소화이다.

이 문제는 일반적인 스케줄링 문제로 전환할 수 있는데 부대는 설비에 해당하고 표적은 수행할 작업에 해당하며 사격소요시간은 각 작업이 설비에서 이루어지는 작업소요시간에 해당한다. 동일한 문제의 사례로 외과 수술의 경우 필요한 의사들이 동시에 참여하여 수술이 시작되나 마취과 의사는 마취에만 참여하고 다른 수술에 참여하거나 다른 업무를 수행할 수 있다. 또한 제조현장에서 하나의 작업이 수행되기 위하여 동시에 여러 개의 설비에서

필요한 작업을 수행하지만 설비별로 작업에 소요되는 시간이 다르고 해당 작업이 완료된 경우 다른 종류의 작업을 수행할 수 있는 형태의 스케줄링 문제도 동일한 구조를 가지고 있다고 할 수 있다. 사격순서 결정문제는 권오정[1]과 Kwon *et al.*[8, 9]에 의하여 처음 제안되었으며 이는 Dobson and Karmarkar[7]가 제시한 다중처리 일정계획 문제의 일반화 모형이라고 할 수 있는데 Dobson and Karmarkar[7]는 하나의 작업이 여러 대의 설비를 공유할 때 시작시간과 완료시간이 동일하다는 조건을 만족하는 경우에 국한하는 것으로 일반적인 사격순서 결정문제에 있어서 사격을 수행하는 부대마다 소요시간이 다를 수 있다는 점이 다르다. 또한 이와 유사한 문제로 세트 다중처리 일정계획 문제가 소개되고 있는데 특정한 작업에 대하여 여러 대의 설비가 동시에 작업을 수행 할 수 있는데 선택된 설비의 조합에 따라 작업시간이 달라지며 전체 작업의 완료시간을 최소화할 수 있는 설비의 세트를 구하는 것도 의사결정의 하나이며 Dobson and Karmarkar[7]과 마찬가지로 하나의 작업이 여러 대의 설비에 의해 작업이 이루어질 경우 소요시간은 선택된 설비에 대해 동일한 경우이다. Chen and Lee[6]는 이 문제에 대해 기계를 일에 할당하는 문제와, 전체작업시간을 최소화하는 스케줄링 문제로 단계화하고 각각 최적해와 하한값을 산출하는 동적계획법 알고리즘을 제시하였다.

공유표적과 단일 표적이 혼합되어 있는 경우 사격순서에 따라 전체적인 사격시간이 달라지며, 전체 사격시간을 최소화하는 사격순서 결정문제는 통합화력 운용이 활발해 질수록 더욱 복잡한 문제로 대두된다. 본 연구의 목적은 사격순서 결정문제에 대한 효율적인 발견적 기법을 제시하고 성능을 평가하는 것으로 예제 실험을 통하여 권오정[1]과 Kwon *et al.*[8, 9]이 제시한 발견적 기법과 성능을 비교하였다. 2장에서 사격순서 결정문제에 대한 정수 계획법 수리모형을 제시하고 3장에서 개선된 발견적 기법을 소개하며 실험결과와 결론을 4장과 5장에 기술하였다.

## 2. 수리모형

사격순서 결정문제를 수리모형으로 표현하면 다음과 같다.

< Notation >

- $W$  : 사격부대의 집합
- $T$  : 표적의 집합
- $P_{ij}$  : 부대  $i$ 가 표적  $j$ 를 사격하는 사격소요 시간
- $W(j)$  : 표적  $j$ 를 사격하는 부대의 집합
- $W(j, k)$  : 표적  $j$ 와  $k$ 를 모두 사격하는 부대의 집합, 즉  $W(j, k) = W(j) \cap W(k)$

< 결정변수 >

- $X_j$  : 표적  $j$ 의 사격 시작시간
- $C_{max}$  : 전체의 사격을 완료하는 시간
- $Y_{jk}$  : 이진변수 - 표적  $j$ 가 표적  $k$ 에 선행할 경우 1, 아니면 0

Min  $C_{max}$

$$\text{s.t. } C_{max} \geq X_j + \max_{i \in W(j)} (P_{ij}) \quad \forall j \in T \quad (1)$$

$$X_j + \max_{i \in W(j, k)} (P_{ij}) \leq X_k + M \cdot (1 - Y_{jk}) \quad \text{for } j < k \quad (2)$$

$$X_k + \max_{i \in W(j, k)} (P_{ik}) \leq X_j + M \cdot Y_{jk} \quad \text{for } j < k \quad (3)$$

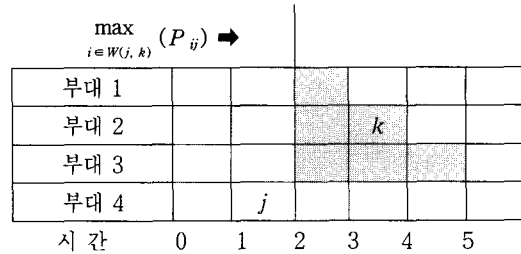
$$X_j \geq 0 \quad \forall j \in T \quad (4)$$

$$Y_{jk} \in \{0, 1\} \quad \forall j, k \in T, j < k$$

$M$ 은 큰 상수

목적함수는 전체의 사격완료시간을 최소화하는 것이다. 식 (1)은 전체의 사격완료시간이 모든 표적의 사격완료시간 보다 큰 것을 나타내며, 식 (2)와 식 (3)은 표적의 선후행 관계를 나타내는 식이고, 식 (4)는 비음조건을 나타내는 식이다. 식 (2)에서 표적  $j$ 의 사격시간이  $\max_{i \in W(j, k)} (P_{ij})$ 로 정해지

는 이유는 다음과 같다.



[그림 1] 표적  $j-k$  순서에서 표적  $k$ 의 사격시작시간

[그림 1]은 표적  $j$ 와 표적  $k$ 의 두 개 표적을 스케줄링할 경우의 한 예를 나타내고 있다. 표적  $j$ 는 부대 2, 3, 4에 의해 각각 1, 2, 4 단위의 시간이 소요되고, 표적  $k$ 는 부대 1, 2, 3에 의해 각각 1, 2, 3 단위의 시간이 소요될 경우,  $(j, k)$ 의 사격완료시간은 5임을 나타내고 있다. 표적  $j-k$  순서로 스케줄링할 경우 표적  $j$ 와  $k$ 를 모두 사격하는 부대( $i \in W(j, k)$ , 부대 2, 부대 3이 해당됨) 중 표적  $j$ 의 가장 긴 사격 소요시간( $\max_{i \in W(j, k)} (P_{ij})$ , 부대 3이 2로 가장 큼)보다는 표적  $k$ 의 사격 시작시간이 늦어야 하기 때문이다. 위 문제는 대표적인 혼합정수계획법 문제로서 변수의 증가에 따라 최적해를 구하는데 소요되는 계산시간이 급격히 증가하여 최적해를 구하기보다는 발견적 기법을 통해 근사해를 구함으로써 좀더 유용하게 도출해를 적용할 수 있다.

## 3. 발견적 기법

사격순서 결정문제에 대해 권오정[1]과 Kwon et al.[8, 9]은 발견적 기법을 제시하였다. 발견적 기법의 기본적인 개념은 동일 표적에 대해 할당된 사격부대의 수가 많은 표적에 우선 순위를 두고 할당하는 근시안적인 방법으로 본 논문에서는 이를 그리디 알고리즘(Greedy Algorithm)이라고 명한다. 발견적 기법의 구성과정은 시간축을 설정하고 한 단위

씩 증가시키면서 할당이 완료되지 않은 표적에 대해 해당 사격부대가 많이 남은 표적을 우선으로 할당하게 된다. 그리디 알고리즘은 대체적으로 우수한 성능을 나타내지만 우선 할당할 수 있는 개별 표적 중에서 선택하기 때문에 좋지 않은 할당과정을 생성할 수 있다. <표 1>과 같이 주어진 문제에 대해 [그림 2]는 그리디 알고리즘의 단점을 보여준다. 시간단위 1에서 사격부대 3이 수행할 수 있는 목표는 표적 2로서 표적 2를 할당하게 되면 표적 2의 사격시간 2를 완료하고 표적 3을 할당하게 되어 시간단위 6에서 사격이 완료된다. 그러나 표적 2와 표적 3은 서로 순서를 바꿈으로써 사격시간 5에 완료하는 최적해를 만들 수 있다. 이와 같이 그리디 알고리즘은 시간단위 축에서 사격 가능한 표적에 대해 선택을 하게 됨으로서 근시안적으로 스케줄링하게 된다. [그림 2]는 표적 2와 표적 3을 사전에 조합하여 하나의 표적처럼 간주하고 이를 스케줄링함으로써 그리디 알고리즘보다 더 좋은 해를 얻을 수 있는 사례를 보여주고 있다.

<표 1> 사격부대 3, 표적 3의 예제 ; 사격시간

	표적 1	표적 2	표적 3
사격부대 1	4	.	.
사격부대 2	2	.	3
사격부대 3	1	2	1

사격부대 1	1	1	1	1	.	.	.
사격부대 2	1	1	.	3	3	3	.
사격부대 3	1	2	2	3	.	.	.
시 간	0	1	2	3	4	5	6

(그리디 알고리즘에 의한 해)

사격부대 1	1	1	1	1	.	.	.
사격부대 2	1	1	3	3	3	.	.
사격부대 3	1	.	3	2	2	.	.
시 간	0	1	2	3	4	5	6

(최적해)

[그림 2] 그리디 알고리즘에 의한 스케줄생성 예제

본 논문에서 제시하는 발견적 기법은 그리디 알고리즘에서 나타나는 문제점을 해결하여 개선된 성능을 얻을 수 있도록 고안되었다. [그림 2]는 특정한 두 개 이상의 표적을 사전에 조합하여 하나의 표적과 같이 간주하고 스케줄링하면 각각의 표적이 개별적으로 스케줄링될 때 보다 사격소요시간을 감소시킬 수 있음을 보여준다. 이를 응용하여 할당되지 않은 표적에 대해 미리 우수한 표적간의 조합을 구하고 사격순서의 스케줄링 과정에 고려함으로써 전체 사격소요시간을 최소화하고자 한다. 두 개의 표적의 경우 서로 인접하여 스케줄링할 경우 각각의 사격소요시간의 합보다 적은 사격소요시간에 사격을 마칠 경우가 있을 수 있는데 이 때 절약가능한 시간을 절약시간이라고 정의한다. 이 개념에 근거하여 하나의 표적처럼 간주되어 스케줄링하는 것이 더 효율적이라고 판단되는 스케줄링 순서의 집합을 구하여 페어(Pair)라고 정의되는 순서쌍을 구하게 된다. 페어에 근거한 사격순서 결정방법을 페어 알고리즘이라고 명명한다.

### 3.1 페어와 절약시간에 대한 정의

페어 알고리즘은 기본적으로 가로축을 시간, 세로축을 사격부대로 하는 간트 차트에서 가로와 세로가 만나 이루는 셀을 기준으로 전개된다. 알고리즘을 구현하기 위한 각각의 단계에서 페어를 구하고 페어에 대한 절약시간을 기준으로 표적을 시간축에 할당한다. 페어는 중복되지 않도록 선정하는 과정을 통하여 결정된다. 이를 정량화하기 위한 기호 정의는 다음과 같다.

$TS_{jk}$  : 표적  $j, k$  순서로 구성할 때 절약되는 시간

$CS_{jk}$  : 표적  $j, k$  순서로 구성할 때 절약되는 셀의 수

$Pair(j, k)$  : 표적  $j, k$  순서로 구성된 짝

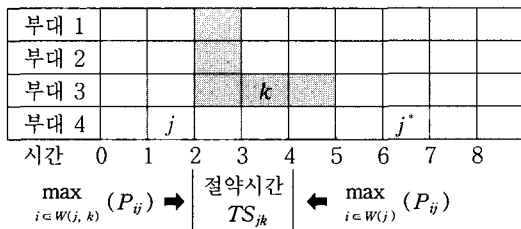
$TS_{jk}$ 는 각각의 사격 소요시간의 합에서  $(j, k)$  순서로 스케줄링하였을 때 필요한 사격 소요시간을

뺀 값으로, 절약할 수 있는 사격소요시간을 의미한다. 이는 표적  $j$ 를 사격하는 가장 긴 시간에서 표적  $j, k$ 를 모두 사격하는 부대 중 표적  $j$ 를 사격하는 가장 긴 시간을 뺀 시간과 표적  $k$ 를 사격하는 가장 긴 시간 중 최소값으로 계산된다.

$$TS_{jk} = \min \left\{ \max_{i \in W(j)} (P_{ij}) - \max_{i \in W(j,k)} (P_{ij}), \max_{i \in W(k)} (P_{ik}) \right\}$$

최소값을 선택하는 이유는 표적  $j$ 와  $k$ 를 모두 사격하는 부대가 존재하지 않을 경우를 보정하기 위한 것이다.  $TS_{jk}$ 의 값이 양(Positive)이 되는 ( $j, k$ )의 조합 중에서 페어가 선택된다.  $CS_{jk}$ 는 절약 시간 ( $TS_{jk}$ )에서 절약될 수 있는 셀의 수를 의미하며 수식 표현은 다음과 같다.

$$CS_{jk} = \sum_{i \in W(k)} \min(P_{ik}, TS_{jk})$$



[그림 3] 절약시간의 계산 사례

[그림 3]의 예에서 표적  $j$ 와  $k$ 를 연결하여 스케줄링할 때  $TS_{jk}$ 와  $CS_{jk}$ 의 계산 예는 다음과 같다.

$$TS_{jk} = \min \left\{ \max_{i \in W(j)} (P_{ij}) - \max_{i \in W(j,k)} (P_{ij}), \max_{i \in W(k)} (P_{ik}) \right\}$$

$$\left( \begin{array}{l} \max_{i \in W(j)} (P_{ij}) = 4 \quad (i = \text{부대 } 4) \\ i \in W(j, k) = \{ \text{부대 } 2, \text{부대 } 3 \} \\ \max_{i \in W(j, k)} (P_{ij}) = 2 \\ \max_{i \in W(k)} (P_{ik}) = 3 \end{array} \right)$$

$$\begin{aligned} \therefore TS_{jk} &= \min \{ 2, 3 \} = 2 \\ CS_{jk} &= \sum_{i \in W(k)} \min(P_{ik}, TS_{jk}) \\ (i \in W(k) &= \{ \text{부대 } 1, \text{부대 } 2, \text{부대 } 3 \}) \\ \therefore CS_{jk} &= 1 + 1 + 2 = 4 \end{aligned}$$

반대로  $k$  다음에  $j$ 가 연결하여 스케줄링하면  $TS_{jk}$ 와  $CS_{jk}$ 는 0이 된다. [그림 3]에서  $j^*$ 는  $j$  표적이  $k$  표적 다음에 스케줄링 되었을 경우를 나타내고 있다. 이와 같은 방법으로 각 표적별 전후 연결 관계에 따라 셀의 절약 개수를 계산하고 이를 정리한 셀 절약도표를 작성할 수 있다. 셀 절약도표를 이용하여 다음과 같은 규칙에 따라 페어를 선정한다.

1. 표적 사격부대 수 1인 표적은 제외한다.
2. 절약되는 셀의 개수 ( $CS_{jk}$ )를 계산한다.
  - while(더 이상의 페어가 존재하지 않을 때까지)
    - 3. 구한 절약 셀의 개수는 행과 열에서 동시에 최대값을 가질 경우 페어로 선정한다.
    - 4. 동일표적이 다른 페어에도 선정되었다면 절약 셀의 개수가 많은 페어를 선정하며, 셀의 절약 개수가 같다면, 임의로 1개만 선정한다.
    - 5. 동일한 표적 2개가 순서가 바뀐 상태로 각각 페어로 구성되었다면, 셀 절약이 큰 것을 선택한다.
    - 6. 페어로 선정된 표적은 이후의 페어구성에서 제외한다.

셀 절약 도표작성 과정에서 표적 사격부대 수가 1인 표적은 사격순서의 할당과정에서 페어로서의 의미가 없으므로 제외하고 그 이외 나머지 표적은 절약 셀의 개수를 계산한다. <표 2>와 같은 셀 절약도표가 만들어졌다면 규칙 3에 의해 선정되는 페어는 Pair(1, 2), Pair(2, 3), Pair(4, 5), Pair(5, 4)이다. 규칙 4는 <표 2>에서 표적 2가 Pair(1, 2), Pair(2, 3)이 중복되어 선정되는데 이를 방지하기 위한 것으로, 이 경우에는 셀 절약 개수가 많은 Pair(1, 2)만 선정하고 Pair(2, 3)은 제외한다. 규칙 5도 마찬가지로 동일 표적 두 개가 순서가 바뀐 상태로 동

시에 페어로 선정될 경우 셀 절약이 크게 되는 순서로 페어를 구성하게 한다. 이와 같은 방법으로 페어가 존재하지 않을 때까지 실행한다. <표 2>에서 표적 1, 2, 4, 5를 제거한 후 이 과정을 통해 Pair(3, 6)을 추가로 구할 수 있다. 따라서 <표 2>에서 최종적으로 선택되는 페어는 Pair(1, 2), Pair(4, 5), Pair(3, 6)이다.

<표 2> 셀 절약 도표 및 페어 선정과정

	표적 1	표적 2	표적 3	표적 4	표적 5	표적 6
표적 1	0	(4)	1	0	0	0
표적 2	1	0	3	1	1	1
표적 3	0	1	0	0	2	(2)
표적 4	0	0	0	0	(4)	3
표적 5	1	1	2	3	0	2
표적 6	1	0	2	2	3	0

### 3.2 페어 알고리즘

페어 알고리즘은 정수로 주어지는 사격 소요시간 기준으로 시간축 상에서 한 단위씩 증가하면서 표적이 선택되는 과정을 통하여 사격순서가 결정된다. 이 때 페어로 선정된 표적의 조합은 하나의 표적인 것처럼 간주되어 순서가 결정된다. 페어 알고리즘은 우선 페어로 선정된 표적을 통합한 후 새로 구성된 표적에 대해 사격 소요시간과 최초사격부대수를 계산함으로써 스케줄링 과정에 대한 준비과정을 수행한다. 사격순서를 결정하는 스케줄링 과정은 시간축 상에서 스케줄링이 이루어지지 않은 표적 중에서 하나씩 선정하여 시간축에 할당하는 과정을 모든 표적이 할당될 때까지 반복된다. 일반적인 스케줄링 과정과는 달리 스케줄링이 이루어지는 현재 시간( $T$ )과 이미 할당이 완료된 표적의 스케줄링 완료시간( $S$ )이 동시에 운영되며, 할당하기로 선정된 표적은 시간축의  $[T, S]$ 사이에서 스케줄링될 수 있는 가장 빠른 시간( $V_j$ )에 할당된다. 시작 가능시간( $V_j$ )은 이미 할당이 완료된 표적과 고려되고 있는 표적간의 셀의 모양에 달려있다.

할당이 완료되면 할당된 표적의 모양에 따라  $T$ 와  $S$ 가 수정된다.

스케줄링이 이루어지지 않은 표적 중에서 할당하고자 하는 표적을 선정하는 방법은 기본적으로 페어의 구성과정에서 사용되었던 절약 셀의 수에 따른다. 페어를 구성할 때는 두 개의 표적에 대해 계산하였지만 스케줄링 과정에서 사용되는 절약 셀의 수는 이미 할당이 완료된 표적의 집합( $A-U$ )을 하나의 표적으로 간주하고, 고려되는 표적과의 연결과정에서 절약되는 셀의 수를 구하면 된다. 선정의 우선순위가 높은 표적은 이미 할당된 표적의 셀에 가장 적합하게 부합되는 표적을 선정하는데 이는 페어를 선정할 때와 마찬가지로 절약되는 셀의 수를 기준으로 선정한다. 현재까지 스케줄링이 완료된 표적의 집합에 연결하여 표적  $j$ 를 스케줄링하였을 때 절약할 수 있는 시간을  $TS_j$ 라 하면 절약되는 셀의 수,  $CS_j$ 는 다음과 같이 계산된다.

$$CS_j = \sum_{i \in W(j)} \min(P_{ij}, TS_j)$$

스케줄링이 진행되는 과정에서 절약 셀의 수를 구하는 과정은 페어의 구성과정에서 사용한 개념과 동일한 것으로 본 알고리즘을 페어 알고리즘으로 명명한 이유이기도 하다. 순차적으로 할당될 표적을 선정할 때 절약되는 셀의 수가 양수(Positive)일 경우에 한하여 진행하고 절약 셀의 수가 0일 경우는 최대한 할당을 뒤로 미루다가 스케줄링이 완료되지 않은 모든 표적만 남았을 때 각 표적의 셀의 크기 순으로 할당한다. 페어 알고리즘을 요약하면 다음과 같다.

- 단계 1) 페어 구성
- 1.1) 셀 절약도표를 이용한 페어 구성 : 페어를 새로운 표적으로 구성
  - 1.2) 각 표적에 대한 사격소요시간 및 최초 사격부대수 계산
- 단계 2) 스케줄링
- 2.1) 현재시간( $T$ )=0, 스케줄링완료시간( $S$ )=0, 스케줄링미완료 표적집합( $U$ )=A

- 2.2) 각 표적( $j \in U$ )에 대한 절약 셀의 수( $CS_j$ )와 시작가능시간( $V_j$ ) 계산
- 2.3) 절약 셀의 수가 양수인 표적의 집합( $R$ ) 계산,  $R = \{k : CS_k > 0\}$
- 2.4)  $R = \emptyset$ 이면 2.10) 스텝으로, 아니면 2.5) 스텝으로 진행
- 2.5) 최대 절약 셀의 수를 가진 표적( $k^*$ ) 선정, 다수 있으면 그 중 최소 사격 소요시간을 가진 표적을 선정
 
$$CS_{k^*} = \max_{j \in R} (CS_j)$$
- 2.6) 표적  $k^*$ 를 시간  $[T, S]$  내에서 시작가능한 빠른 시간으로 스케줄링.
- 2.7) 스케줄링 완료시간( $S$ ) 수정 및  $R = R - \{k^*\}$ ,  $U = U - \{k^*\}$
- 2.8) 현재시간( $T$ ) 수정 ;  $T = \max[T, \min_{j \in R}(V_j)]$ ,
- 2.9)  $U = \emptyset$ 이면 스케줄링 종료, 아니면 2.2) 스텝으로 진행.
- 2.10) 각 표적( $j \in U$ )에 대해 셀의 수 크기 순으로 스케줄링 및 종료

단계 1에서 페어와 나머지 표적을 새로운 표적으로 구성하고 표적 사격부대 수를 수정한다. 페어로 구성된 표적은 표적 두 개가 결합된 형태로 사격 시작시간이 동일하지 않아 문제의 성격이 바뀌므로 본 알고리즘에서 사용하기 위한 표적 사격부대 수로 수정되는 표적 사격부대 수는 각 표적별 최초 사격하는 부대의 수로 정의한다. [그림 4]는 페어가 형성되는 과정에서 최초 사격하는 부대수의 수정되는 과정을 보여준다. [그림 4](a)는 최초 표적을 나타내며 [그림 4](b)는 페어로 구성된 후의 표적을 나타낸 것으로 [그림 4](a)의 표적 (1, 2, 3)은 최초 사격부대수가 각각 (2, 3, 4)이었다. 페어가 형성된 이후의 표적은 이전의 (1, 2)가 합쳐서 표적 1이 되고 표적 3은 표적 2로 수정되며 표적 (1, 2)의

최초 사격 부대수도 각각 (2, 4)로 수정된다.

단계 2는 스케줄링이 진행되는 주요 단계이다. 2.1)은 초기화 과정이며 2.2)에서 2.9)까지의 과정은 절약 셀의 수가 양수가 되는 한 하나의 표적이 할당될 때마다 반복되는 일련의 단계들이다. 만약 셀 절약이 동일한 표적이 다수 존재한다면, 이 표적들 중 최소 사격시간을 갖는 표적을 선택한다. 이 과정을 [그림 5]를 통하여 설명하면 [그림 5](a)에서 현재 시간 ( $T$ )는  $S-3$ 이며 스케줄링 완료시간은  $S$ 이다. 또한 스케줄링 미완료표적의 집합,  $U = \{1, 2, 3\}$ 이며  $V_1 = S-2$ ,  $V_2 = S-3$ ,  $V_3 = S-2$ 이다. 표적 1, 2, 3이 시간  $V_1, V_2, V_3$ 에 할당되면 셀 절약은 각각 3, 2, 3이 된다. 셀 절약이 가장 큰 표적 1, 표적 3이 선정되며, 그 중 최소 사격시간을 갖는 표적 3이 최종 선택된다. [그림 5](b)는 셀 절약이 적은 표적이 선정된 경우를 나타내고 있으며 [그림 5](c)는 셀 절약이 같은 표적 중 최소 사격시간을 갖는 표적을 선택하는 이유를 설명하고 있으며 [그림 5](d)는 알고리즘에서 제시하는 규칙대로 진행되었을 때 효율적인 결과를 도출하는 사례를 보여주고 있다.

단계 2.10)에서는 남아 있는 표적들 중 셀 개수가 가장 많은 표적을 선정한다. 이 때는 정의에 따라  $T$ 와  $S$ 의 값이 같을 때며 절약 셀의 수가 많은 표적을 우선 할당하는 것이 반드시 좋은 스케줄링 일 것이라는 증거는 없으나 셀 개수가 많은 표적을 선정하는 것은 그 표적이 페어로 구성되어 있는 표적이거나 만약, 페어가 존재하지 않더라도 표적 사격부대 수가 많은 표적이 선정될 가능성이 크기 때문이다. [그림 6]은 셀 개수가 많은 표적을 선정하

부대 1				2			3		
부대 2	1	1					3		
부대 3	1			2			3	3	
부대 4				2	2		3	3	3

(a)

			1				2		
	1	1					2		
	1	1					2	2	
			1	1			2	2	2

(b)

[그림 4] 최초 사격부대수의 수정 이유

부대 1								1									2	2			3				
부대 2																									
부대 3																					3	3			
부대 4																									
시 간		S-2		S		(a)																			
부대 1		2	2			1																3			
부대 2																									
부대 3						1	1	1														3	3		
부대 4																									
시 간		S-2		S		(b)																			
부대 1			1																			3			
부대 2																									
부대 3			1	1	1																	3	3		
부대 4																									
시 간		S-2		S		(c)																			
부대 1				3		1																			
부대 2																									
부대 3			3	3	1	1	1																		
부대 4																									
시 간		S-2		S		(d)																			

[그림 5] 단계 2.5) 및 단계 2.6)의 표적 할당 이유

(a)	부대 1																								
	부대 2																								
	부대 3																								
	부대 4																								
시 간		S																		남 은 표 적					
(b)	부대 1					1	1	2	2	3															
	부대 2					1	2	2																	
	부대 3						2	2	4	4															
	부대 4					1					3	3	3	3											
시 간		S																		S+8					
(c)	부대 1					3	2	2		1	1														
	부대 2					2	2			1															
	부대 3					2	2	4	4																
	부대 4					3	3	3	3	1															
시 간		S																		S+6					

[그림 6] 단계 2.10)의 표적 할당 이유



는 이유를 설명하고 있다. [그림 6](a)와 같이 시간  $T(=S)$ 에서 할당되지 않은 표적 (1, 2, 3, 4)가 있을 때 각각의 최초 사격 부대 수는 (3, 2, 2, 1)이며, 셀의 개수는 각각 (4, 6, 5, 2)이다. [그림 6](b)는 최초 사격 부대 수가 많은 표적을 선정한 경우로서 표적 1이 우선 선정이 되고 셀 절약이 있는 표적 (2, 4, 3)순서로 결정되게 되므로 최종 시간은  $(S + 8)$ 이 된다. [그림 6](c)는 셀의 개수가 많은 표적을 선정한 경우로서 표적 2를 우선 선정한다면 표적 (3, 4, 1)이 순서대로 기록되고 최종시간은  $(S + 6)$ 이 된다.

#### 4. 실험 결과

제시된 알고리즘의 성능을 평가하기 위한 실험으로 실제의 상황에 근접한 경우의 예제를 생성하였다. 단위 사격부대의 수는 대대급 부대인 3, 사단급 부대인 12, 군단급 부대인 20개로 하였으며, 표적의 수는 3개 사격부대의 경우 (10, 15, 20)개의 3종류, 12개 부대의 경우는 (30, 50, 70)의 3종류, 20개 부대의 경우는 (5, 10, 15, 60, 80, 100)에 대하여 실험하였다. 표적의 수는 사격부대의 3~5배 수준에서 결정하였으나 최적해와 비교를 통한 성능 실험을 위해 20개 부대의 경우는 일정한 시간 내에 최적해의 계산이 가능한 적은 수의 표적에 대하여

실험을 실시하였다. 표적별 표적 사격부대 수는 1, 2(부대수 3인 경우), 3, 6, 9, 12(부대수 3, 12, 20인 경우), 20(부대수 20인 경우)을 발생토록 하였는데, 통상적으로 3개 단위의 사격부대규모로 사격이 실시되는 특성을 고려하여 이와 같이 생성하였다. 사격 부대수가 20개일 경우 표적 사격부대 수가 20인 표적을 표적 사격부대 수 3이상인 표적 중 임의의 표적에서 1개를 대신 생성하고, 나머지는 사격부대 수가 12개인 경우와 동일하게 생성하였다.

부대수와 표적 수에 따라 3가지 타입으로 나누어 실험하였다. 타입 1은 표적 사격부대 수가 1인 표적(단일표적)을 표적수 대비 1/2 비율로 생성하고 나머지는 동일한 비율로 생성하였으며, 타입 2는 표적 사격부대 수를 동일 비율로, 타입 3은 부대수에 따라 적정값의 표적 사격부대 수에 전체 표적수의 1/2 비율로 할당하고 2개 또는 4개의 표적 사격부대의 수에 균등하게 할당하였다. 다만 20개 부대의 예제의 경우 20개 부대 모두가 동시에 사격하는 표적을 반드시 1개씩 포함되도록 하였다. 예를 들면 사격부대 수가 3이고 표적수가 10일 때 타입 1에서는 표적 사격부대 수가 1인 표적이 5개, 표적 사격부대 수가 2와 3인 표적이 각각 2~3개 발생하며, 타입 2의 경우 표적 사격부대 수가 각각 1, 2, 3인 표적은 모두 동일하게 3~4개의 비율로, 타입 3에서는 표적 사격부대 수가 각각 2~3, 5, 2~3개의

〈표 3〉 실험예제의 생성 기준

예 제	사격부대	표적 사격부대 수						
		1	2	3	6	9	12	20
타입 1	3개 부대	1/2	1/4	1/4	·	·	·	·
	12개 부대	1/2	·	1/8	1/8	1/8	1/8	·
	20개 부대	1/2	·	1/8	1/8	1/8	1/8	(1개)
타입 2	3개 부대	1/3	1/3	1/3	·	·	·	·
	12개 부대	1/5	·	1/5	1/5	1/5	1/5	·
	20개 부대	1/5	·	1/5	1/5	1/5	1/5	(1개)
타입 3	3개 부대	1/4	1/2	1/4	·	·	·	·
	12개 부대	1/8	·	1/8	1/2	1/8	1/8	·
	20개 부대	1/8	·	1/8	1/2	1/8	1/8	(1개)

비율로 발생토록 한 것이다. 타입에 따라 표적별로 갖는 표적 사격부대 수와 할당되는 사격부대를 전체적으로 나타내면 <표 3>과 같이 비율로 나타낼 수 있으며 사격 소요시간을 4 이하의 정수로 임의로 생성하였다.

최적해의 산출은 ILOG OPL Studio 3.1을 이용하였으며 페어 알고리즘과 그리디 알고리즘은 C++로 구현하여 PC(Pentium IV 1.6GHz)상에서 실험하였다. 최적해는 표적수가 20개 이상인 표적은 30분 이내에 해를 구할 수 없어 하한값을 구하여 비교함으로써 성능을 평가하였다. 하한값은 사격부대별로 사격시간의 합이 가장 많은 부대의 사격시간의 합이며 다음의 식으로 계산할 수 있다.

$$LB(\text{하한값}) = \max_{i \in W} \sum_{j \in T} P_{ij}$$

<표 4>는 각 예제별로 서로 다른 Seed Number를 사용하여 생성된 데이터에 대해 30회씩 반복 실험하여 평균한 결과로서 페어 알고리즘과 그리디 알고리즘으로 구한 전체사격 소요시간의 결과와 최적해(또는 하한값)와의 차이를 최적해(또는 하한값)으로 나눈 값을 성능 지수라고 정의하고 이를 정리한 것이다.

성능지수=

$$\frac{\text{발견적기법에 의한 해} - \text{최적해(또는 하한값)}}{\text{최적해(또는 하한값)}}$$

최적해의 산출은 표적수가 20개 이하인 예제들에 대해서 산출할 수 있었으며, 최적해 산출 시간은 표적수가 많아질수록 증가하는데 각 문제별로 평균 0.9~300초가 소요되었다. 비교우위의 횟수는 각 문제별로 30회 실험하였을 때 두 알고리즘이 산출한 결과 중 보다 우수한 해를 산출한 횟수이며, 두 알고리즘의 횟수를 더했을 때 30회가 되지 않는 것은 동일한 결과를 나타냈을 경우이다.

최적해를 구한 문제에서 최적해와의 성능지수가 타입 1에서는 페어 알고리즘이 1.66~7.51%, 그리디 알고리즘은 1.61~8.88%를 보여 최적해 대비 평

균에서 페어 알고리즘이 평균적으로 0.24%의 근소한 차이로 앞서며, 마찬가지로 타입 2에서 각각 4.29%, 5.00%, 타입 3에서도 5.37%, 5.76%로 페어 알고리즘이 근소한 차이로 앞섰다. 최적해를 구할 수 없었던 문제의 예제는 하한값으로 성능지수를 비교하였는데 타입 1에서 페어 알고리즘의 평균이 25.45%, 그리디 알고리즘의 평균은 32.74%였으며, 타입 2에서는 각각 33.66%, 35.94%, 타입 3에서는 34.43%와 34.79%로 페어 알고리즘이 각각 7.29%, 2.28%, 0.36% 앞섰다. 또한, 두 알고리즘 중 보다 좋은 해를 구한 비교우위 횟수의 평균은 타입 1에서 페어 알고리즘이 25.2회, 그리디 알고리즘이 2.7회로 페어 알고리즘이 압도적으로 많았고, 타입 2는 각각 17.6회와 8.6회로 페어 알고리즘이 약 2배가 많으며, 타입 3에서는 14.6회와 10.0회로 페어 알고리즘이 약 5회 많았다. 계산 시간에 있어서는 그리디 알고리즘보다 페어 알고리즘이 더 소요되지만 최적화방법으로 30분 이내에 해를 구하지 못하는 문제에 대해서도 페어 알고리즘은 최대 1.8초 이내에 결과를 도출함으로써 실용적인 측면에서 효용성이 높다고 할 수 있다.

전체적으로 볼 때, 타입 1에서 문제의 크기가 클수록 페어 알고리즘이 우수한 해를 도출하며, 성능지수가 최고 9.5% 앞서는 우수한 결과를 나타내었으며, 타입 2, 타입 3에서는 성능지수의 차이가 적었다. 즉, 단일표적의 전체적인 발생 비율이 크고 표적의 수가 많아질수록 페어 알고리즘이 우수한 성능을 나타내게 되는데, 이 이유는 단일표적이 많아질수록 공유표적의 개수는 상대적으로 적어지고, 적은 수의 공유표적들이 짝으로 구성되고, 셀 절약 개수가 많은 순으로 결과가 정렬이 되면서 발생하는 빈 공간에 상대적으로 많은 단일표적이 그 공간을 채우게 되어 결과적으로 전체사격 소요시간이 줄어들게 되기 때문이다. 반대로 단일표적의 발생 비율이 작고, 공유표적의 발생 비율이 많아질수록 이러한 가능성은 줄어들게 되어 타입 1과 같은 우수한 결과를 기대하기는 힘들다. 그러나 공유표적만 발생한다거나, 단일표적만 발생하는 경우보다는,

<표 4> 실험 결과

구 분		성능지수(%)			비교 우위 횟수(회)		최적해 찾은 횟수(회)		계산시간(초)			
		페어	그리디	차 이	페어	그리디	페어	그리디	최 적	페어	그리디	
1	타입 최적해 대비	(3, 10)	1.66	1.61	0.05	3	4	22	25	16.7	0.001	0.000
		(3, 15)	1.99	1.91	0.09	5	8	19	24	264.2	0.002	0.001
		(3, 20)	2.70	2.14	0.56	5	10	13	20	300.0	0.003	0.001
		(20, 5)	2.87	1.78	1.09	3	4	24	25	1.0	0.001	0.001
		(20, 10)	4.04	5.89	-1.85	7	5	20	19	1.1	0.002	0.001
		(20, 15)	7.51	8.88	-1.37	15	8	8	8	17.5	0.005	0.001
		평 균	3.46	3.70	-0.24	6.3	6.5	17.7	20.2	100.1	0.002	0.001
	타입 하한값 대비	(12, 30)	18.77	23.45	-4.68	18	6	-	-	-	0.023	0.003
		(12, 50)	20.14	25.90	-5.75	24	2	-	-	-	0.108	0.014
		(12, 70)	21.72	29.39	-7.66	30	0	-	-	-	0.279	0.035
		(20, 60)	30.22	37.38	-7.16	24	3	-	-	-	0.284	0.021
		(20, 80)	29.63	39.11	-9.48	26	4	-	-	-	0.666	0.042
		(20, 100)	32.21	41.22	-9.01	29	1	-	-	-	1.326	0.075
		평 균	25.45	32.74	-7.29	25.2	2.7	-	-	-	0.448	0.032
2	타입 최적해 대비	(3, 10)	3.91	2.84	1.07	5	8	17	17	32.6	0.001	0.000
		(3, 15)	2.99	3.23	-0.24	5	4	17	15	160.7	0.002	0.001
		(20, 5)	1.44	3.18	-1.74	4	1	26	23	0.9	0.001	0.000
		(20, 10)	6.13	7.61	-1.48	14	7	12	4	3.2	0.002	0.001
		(20, 15)	6.98	8.13	-1.15	12	10	3	6	175.8	0.006	0.001
			평 균	4.29	5.00	-0.71	8.0	6.0	15.0	13.0	74.6	0.002
	타입 하한값 대비	(3, 20)	5.55	5.59	-0.04	8	11	-	-	-	0.003	0.001
		(12, 30)	28.65	29.20	-0.55	13	10	-	-	-	0.031	0.007
		(12, 50)	34.39	36.90	-2.51	23	5	-	-	-	0.151	0.026
		(12, 70)	33.84	34.55	-0.71	15	14	-	-	-	0.428	0.065
		(20, 60)	44.48	48.46	-3.97	20	8	-	-	-	0.368	0.036
		(20, 80)	45.48	49.58	-4.10	17	9	-	-	-	0.916	0.076
		평 균	33.66	35.94	-2.27	17.6	8.6	-	-	-	0.519	0.049
	3	타입 최적해 대비	(3, 10)	3.97	3.92	0.05	7	10	14	15	48.5	0.001
(3, 15)			3.93	4.57	-0.64	10	7	12	12	288.8	0.001	0.001
(20, 5)			1.39	5.11	-3.71	8	2	27	20	0.9	0.000	0.001
(20, 10)			7.88	7.98	-0.10	9	14	7	4	5.1	0.002	0.001
(20, 15)			9.66	7.22	2.44	4	17	0	5	233.3	0.006	0.002
			평 균	5.37	5.76	-0.39	7.6	10.0	12.0	11.2	115.3	0.002
타입 하한값 대비		(3, 20)	5.10	5.75	-0.65	11	7	-	-	-	0.003	0.001
		(12, 30)	35.23	35.64	-0.41	13	11	-	-	-	0.033	0.007
		(12, 50)	37.58	37.79	-0.22	13	13	-	-	-	0.165	0.027
		(12, 70)	36.54	37.45	-0.91	21	5	-	-	-	0.455	0.067
		(20, 60)	42.31	42.68	-0.37	16	10	-	-	-	0.402	0.035
		(20, 80)	41.84	42.70	-0.85	16	10	-	-	-	0.916	0.073
		평 균	34.43	34.79	-0.36	14.6	10.0	-	-	-	0.540	0.049

주) 1. 차이 : (페어 알고리즘 결과의 성능지수) - (그리디 알고리즘 결과의 성능지수).  
 2. 예제의 표기 : (A, B) = (사격부대의수, 표적 수).

공유표적과 단일표적이 비슷한 비율로 발생하는 것이 보다 현실적이며, 공유표적이거나 단일표적의 어느 한쪽이 많이 발생하게 된다면 문제의 성격도 어느 한쪽으로 치우치게 되어 포괄적이지 못하여 비슷한 비율로 발생하는 것이 보다 현실적인 경우라 할 수 있다. 이 경우에 해당하는 타입 1에서 페어 알고리즘은 특히 우수한 성능을 나타냈으며, 공유표적이 대부분 발생하게 되는 타입 3에서도 페어 알고리즘은 최소한 그리디 알고리즘보다는 좋은 성능을 나타내었다. 따라서 보다 현실적인 상황에서 더욱 우수한 성능을 나타내는 것이 페어 알고리즘이라 할 수 있다.

## 5. 결 론

다수의 사격부대가 동시에 동일표적을 사격하고, 각 사격부대의 사격완료시간이 다른 상태에서 전체 사격시간을 최소화 하는 본 문제는 연구가 거의 진행되지 않은 문제이다. 이러한 문제는 현재 군에서 강조되고 있는 무기체계가 다른 부대까지 화력을 집중하는 통합 화력운용 상황에서 더욱 빈번한 발생이 예상된다. 또한, 일반 생산현장에서도 여러 대의 설비가 공동의 작업을 동시에 실행하지만 각 설비별로 작업소요시간이 다른 경우에서 발생할 수 있는 문제이다. 본 연구는 공유표적과 단일표적이 혼합되어 나타나는 상황에서 전체 사격시간을 최소화하기 위한 일정계획의 알고리즘을 제시한 연구이다. 제시한 페어 알고리즘의 성능 검증을 위해 세 가지 타입의 예제로 구분하여 단일표적과 공유표적의 발생 비율을 달리 하였으며, 타입별로 사격부대수와 표적수를 또 다시 여러 경우로 하여 다양한 상황을 부여하였다. 그 결과 페어 알고리즘이 기존에 발표되었던 그리디 알고리즘보다 우수한 결과를 나타내었으며, 특히 공유표적과 단일표적이 비슷한 비율로 발생하게 되는 상황을 반영하는 보다 현실적인 상황인 타입 1에서 가장 우수한 성능을 나타내었다. 현재 야전부대에서는 표적 공격에 필요한 사격제원 계산은 자동화가 진행되었으나 전술

적인 사격순서 결정문제는 자동화가 구현되지 않은 상태로 향후 전술분야 자동화 체제 구축에 본 알고리즘이 사용된다면 유용할 것이다.

본 문제의 확장된 형태로서 특정계열의 표적에 특정 부대가 사격할 경우 준비교체시간이 필요한 경우와 특정시간에 특정표적이 사전에 선정되어 있는 상태에서 나머지 표적들의 사격순서를 결정하는 경우 등 보다 현실에 접근한 문제가 앞으로의 좋은 연구과제가 되리라 생각된다.

## 참 고 문 헌

- [1] 권오정, "포병사격순서 결정에 관한 연구", KAIST 박사학위 논문, 1997.
- [2] 김성환, "차세대구축함 대공무기체계의 최적 탄약 적재량 결정에 관한 연구", 국방대학원 석사논문, 1999.
- [3] 육군본부, 「포병 운용」, 야전교범 2-2, 2002.
- [4] 육군본부, 「화력 운용」, 야전교범 6-1, 1997.
- [5] 장상국, 강성진, "포병 표적할당에 관한 개선 모형", 「한국 군사운영분석학회지」, 제20권, 제2호(1994), pp.62-78.
- [6] Chen, J. and C-Y. Lee, "General Multi-Processor Task Scheduling," *Naval Research Logistics*, Vol.46(1999), pp.57-74.
- [7] Dobson, G. and U. Karmarkar, "Simultaneous Resource Scheduling to Minimize Weighted Flow Times," *Operations Research*, Vol.37(1989), pp.592-600.
- [8] Kwon, O-J., D-H. Kang, K-S. Lee and S-S. Park, "Lagrangian Relaxation Approach to the Targeting Problem," *Naval Research Logistics*, Vol.46(1999), pp.640-653.
- [9] Kwon, O-J., K-S. Lee and S-S. Park, "Targeting and Scheduling Problem for Field Artillery," *Computers and Industrial Engineering*, Vol.33(1997), pp.693-696.