J. Soc. Korea Ind. Syst. Eng Vol. 38, No. 4 : 132-141, December 2015
http://dx.doi.org/10.11627/jkise.2015.38.4.132
ISSN : 2005-0461(print)
ISSN : 2287-7975(online)

A Branch-and-Bound Algorithm to Minimize the Makespan in a Fire Scheduling Problem

Young-Ho Cha* · June-Young Bang**†

*50th Division, Republic of Korea Army, Korea
**Sungkyul University, Gyeonggi-do, Korea

최소 종료시간 사격 스케줄을 위한 분지계획법 알고리즘 연구

차영호* · 방준영**†

*대한민국 육군 50사단
**성결대학교 산업경영학부

We focus on the fire scheduling problem (FSP), the problem of determining the sequence of targets to be fired at, for the objective of minimizing makespan to achieve tactical goals. In this paper, we assume that there are m available weapons to fire at n targets (> 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 develop several dominance properties and a lower bound for the problem, and suggest a branch and bound algorithm implementing them. Also, In addition, heuristic algorithms that can be used for obtaining an initial upper bound in the B&B algorithm and for obtaining good solutions in a short time were developed. Computational experiments are performed on randomly generated test problems and results show that the suggested algorithm solves problems of a medium size in a reasonable amount of computation time. The proposed lower bound, the dominance properties, and the heuristics for upper bound are tested in B&B respectively, and the result showed that lower bound is effective to fathoming nodes and the dominance properties and heuristics also worked well. Also, it is showed that the CPU time required by this algorithm increases rapidly as the problem size increases. Therefore, the suggested B&B algorithm would be limited to solve large size problems. However, the employed heuristic algorithms can be effectively used in the B&B algorithm and can give good solutions for large problems within a few seconds.

Keywords : Military, Fire Scheduling, Branch and Bound

1. Introduction

This research considers the issue of scheduling enemy targets which are assigned to different artillery units under the situation of planned artillery attack operations. In the planned artillery attack operation, it is assumed that information on a group of enemy targets such as the locations and characteristics is already on hand and the operation plan will be prepared in advance before the execution of operation. In combat situation, identified enemy targets should be fired and destroyed by various friendly fire power such as combat aircrafts, field artillery units, and missiles, etc. In this research, among these fire power assets, we mainly focus on the field artillery units. A planned artillery attack operation is executed with two sequential procedures. First, at assignment phase,

Received 8 October 2015; Finally Revised 16 November 2015; Accepted 7 December 2015
† Corresponding Author : jybang@sungkyul.ac.kr

enemy targets are assigned to a weapon or a group of weapons. Here, a weapon represents an artillery battery, which is the basic unit to execute artillery attack operations and is typically composed of six howitzers. Note that when there are more targets than available weapons, one weapon may have to be assigned to more than one target. Second, at scheduling phase, scheduling firing operations against the targets that are assigned to each weapon is calculated. Note that a target can be assigned to more than one weapon and the prespecified set of weapons should start firing at the target simultaneously to achieve surprise effect. In this paper, we assume that the assignment is given, thus we only consider the second phase which is called fire scheduling problem (FSP).

In the following, we will explain that the FSP is the generalization of multiprocessor task scheduling problem (MTSP). In MTSP, a collection of n tasks has to be executed by m processors. Task Jj (j = 1, ⋯, n) requires processing during a given uninterrupted time p_j [13]. Each task requires the simultaneous use of a set of pre-specified processors for its execution; each processor can execute at most one task at a time. Note that, in FSP the processing time of each target (task) may be different among prespecified weapons (processors) due to the assignment results. Thus, the FSP is a generalized version of the MTSP.

Since most research on the weapon management is mainly focused on the quality of weapon firing [1, 2, 25], the research on the FSP is very rare. First introduced the FSP, Kwon et al. [22] consider the FSP with the objective of minimizing makespan under the assumption that the targets are fixed (do not move). Note that they focus on achieving surprise attack effect through the quickest completion of the firing operation. Kim and Lee [19] also consider the FSP and they provide a heuristic algorithm. Kim and Lee [18] combined the fire target allocation problem and sequencing problem in one algorithm and suggest a heuristic approach.

Due to the lack of literature on the FSP, we mainly review the multiprocessor task scheduling problem which is different but share common characteristics comparing with traditional machine scheduling theory. Bozoki et al. [6] first considered the multiprocessor task scheduling problem. However, the research area did not attract much attention until the computing power tremendously increases in 1990's. Along with survey papers by Drozdowski [12] and Lee et al. [25], there have been a number of research articles on the multiprocessor task scheduling problems. Bozoki et al. [6], Bianco et al. [4] and Kramer [20] provide branch and bound algorithms for the problem.

The multiprocessor task scheduling problem is denoted as Pk|fix|Cmax in the literature [14]. Feasibility and approximability of the Pk|fix|Cmax problem have been studied by many researchers. The P2|fix|Cmax problem is a generalized version of the classical job scheduling problem on a 2-processor system [13], thus it is NP-hard. Kubale [21] studied the complexity of the particular problem of scheduling tasks that require at most two prespecified processors. Chen et al. [9] developed a fully polynomial time approximation scheme for the P2|fix|Cmax problem. Hoogeveen et al. [15] showed that the P3|fix|Cmax problem is NP-hard in the strong sense thus it does not have a fully polynomial time approximation scheme unless P = NP. Note that firing scheduling problem is also NP-hard since FSP is generalized version of P3|fix|Cmax problem. Blazewicz et al. [5] developed a polynomial time approximation algorithm of ratio 4/3 for the problem P3|fix|Cmax, which was improved by Dell'Olmo et al. [11], who gave a polynomial time approximation algorithm of ratio 5/4 for the same problem. Both algorithms are based on the study of a special type of scheduling called normal scheduling. Brucker et al. [7] present polynomial algorithms for multiprocessor task problems with unit processing times and more general objective functions. More recently, Amoura et al. developed a polynomial time approximation scheme for the problem Pk|fix|Cmax for every fixed integer k. Polynomial time approximation schemes for a more generalized version of the Pk|fix|Cmax problem have also been developed recently [10, 17]. Huang et al. [16] derive an approximation algorithm of ratio 1.5 for the P4|fix|Cmax problem, which significantly improves the best previous ratio of 2 for practical algorithms for the problem. Recently, Yoon et al. [26] use the basic concept of a branch-and-bound algorithm to find the optimal fire sequence minimizing the completion time of firing operations and Cha and Kim [8] propose the branch-and-bound algorithm for minimizing the threat rate during the firing operation. In this research, we extend the problem described by Yoon et al. [26] and develop several dominance properties and a lower bound for the FSP and present a branch and bound (B&B) algorithm using them. Due to the complexity of the problem and the requirement of prompt decisions in practice, we also suggest heuristic algorithms that can give reasonably good solutions in a short time. The heuristic algorithms are also used in the B&B algorithm for an initial upper bound. This paper is organized as follows. In section 2, we describe the FSP considered in

this study in more detail. Sections 3 and 4 present dominance properties and the branch and bound algorithm, respectively. For evaluation of the performance of the suggested algorithms, a series of computational tests is performed on randomly generated problem instances and results are reported in section 5. Finally, section 6 concludes the paper with a short summary and recommendations for further research.

2. Problem Description

The *fire scheduling problem* (FSP) considered in this research is the problem of scheduling a set of targets for the objective of minimizing makespan. To describe the branch and bound algorithm clearly, we first define terms that will be used in this research.

firing operation (i, j) :
an operation which is executed by weapon i to target j, and the duration of the operation is d_{ij};

job : a
set of firing operations against one target;

compatible targets :
targets which are assigned to two disjoint set of weapons;

incompatible targets :
targets which are assigned to at least one common weapon.

period :
the unit time of length, that is, a period is the minimum time duration for a firing operation. The firing duration for a firing operation is given as an integer multiple of the length of a period.

Note a job for a single-weapon target, which is assigned to only one weapon, is composed of only one firing operation, while a job for a multiple-weapon target, which is assigned to more than one weapon, is composed of two or more firing operations. In this research, a pair (i, j) represents a firing operation in which weapon i is assigned to target j. For example, if weapon 1, 2, and 4 are assigned to a target 1, then job 1 (or target 1) is composed of three firing operations (1, 1), (2, 1) and (4, 1).

In this study, the following assumptions are made.

1) The information on targets such as locations and sizes are known in advance.
2) The weapon-target allocation result and the durations of the firing operations are given.
3) The firing duration for a firing operation is given as an integer multiple of the length of a period.
4) Each weapon can fire at most one target in a single period.
5) Preemption of a job is not allowed.
6) Jobs assigned to more than one weapon should be started simultaneously to achieve surprise effect. However, the finish times of the jobs from different weapons may be different.

Solution representation scheme

Schedules are represented with a sequence of jobs. On each weapon, a firing operation of a job in an earlier position of the sequence is scheduled earlier. If two targets require firing operations that are assigned to two disjoint sets of weapons, we only consider a sequence in which the job with an operation assigned to the lowest-indexed weapon is placed earlier.

While there may be various solution representation schemes for this problem, we represent the schedules (partial schedules) with a sequence of jobs. This implies that the sequence of jobs determine a unique schedule. Remind that a job is set of firing operations against one target. On each weapon, a firing operation of a job in an earlier position of the sequence is scheduled earlier than that of a job placed in a later position. In other words, the sequence of firing operations on each weapon is obtained from the sequence of jobs. Note that a job in an earlier position in the sequence is not necessarily started earlier than another job if the two jobs are compatible. When a schedule is constructed from a sequence, one should consider the constraint that firing operations of the same job should be started simultaneously.

In the following we present a mathematical formulation of the FSP. In the formulation the following notation is used and the other notation follows the same with previous definition.

i	index for weapon systems ($i = 1, \cdots, W$)
j	index for targets ($j = 1, \cdots, T$)
$W(j, k)$	set of weapons that are needed for firing at both targets j and k
x_{jk}	binary variable that is equal to 1 if firing operation for target j precedes that for target k in a given firing sequence, for all $j, k \in R, j < k$
M	a large number

Now, a nonlinear integer programming formulation is given.

Minimize $C_{\max}$

Subject to

$$C_{\max} \geq t_j + \max_i (d_{ij}) \qquad \forall j \qquad (1)$$

$$t_j - t_k \leq -\max_{i \in W(j,k)} (d_{ij}) + M(1 - x_{jk}) \qquad \forall j, k (j < k) \qquad (2)$$

$$t_k - t_j \leq -\max_{i \in W(j,k)} (d_{ik}) + M(x_{jk}) \qquad \forall j, k (j < k) \qquad (3)$$

$$t_j \geq 0 \qquad \forall j \qquad (4)$$

$$x_{jk} \in \{0, 1\} \qquad \forall j, k (j < k) \qquad (5)$$

Equation (2) and (3) are the disjunctive constraints which represent the starting time relationship between any two targets. x_{jk} is a binary variable which indicates whether task j precedes task k in the sequence and M is a large number. Note that if x_{jk} = 1, then the start of job k, $t_k \geq t_j$+max (d_{ij}) and equation (2) becomes nonbinding because the right hand side becomes a large number. If x_{jk} = 0, then the start of job j, $t_j \geq t_k$+*max* (d_{ik}) and equation (3) becomes non-binding. It is possible that several targets can be fired at simultaneously if they do not share a common weapon that is supposed to fire at them. If W(j, k) is empty, then max(d_{ij}) and max (d_{ik}) will be 0 and, by constraints (2) and (3), t_j = t_k can be accomplished.

For a better description of the problem we present a simple example in which a schedule is represented by a sequence of jobs. For example, in <Figure 1>, three jobs (T1, T2, and T3) are scheduled on three weapons. T1 and T3 represent set of firing operations against multiple-weapon targets 1 and 3, respectively, while T2 represents a firing operation against a single-weapon target 2. A job sequence (1-2-3) represents the schedule as shown in the figure. With this solution representation method, one can always generate a unique non-delay schedule from a sequence of the jobs.

3. Dominance Properties

In this section, we present dominance properties that can be used in a branch and bound (B&B) algorithm for the fire scheduling problem (FSP) considered in this research. We use the following additional notation.

$W(j)$ set of weapons which are assigned to target j

$W(j, k)$ set of weapons which are assigned to both target j and k

d_{ij} firing duration from weapon i to target j

TS_{jk} saving period when target k is immediately sequenced after target j

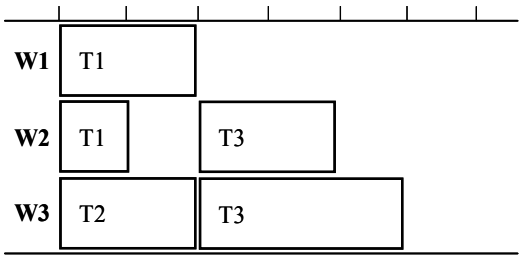
$$TS_{jk} = \min \left\{ \max_{i \in W(j)} (d_{ij}) - \max_{i \in W(j,k)} (d_{ij}),\ \max_{i \in W(k)} (d_{ik}) \right\}$$

σ partial sequence

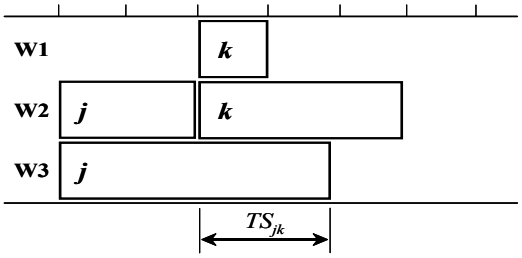
σ_k partial sequence appending job j and k in this order at partial sequence σ

w_{jk} weapon which decides the completion time of partial sequence σ_{jk}

In the notation, TS_{jk} means the saving period [19] when target j and k are scheduled in this order. Clearly, that implies value of (sum of longest firing duration of target j and that of target k)-(the firing duration needed when target j and k are combined). In <Figure 2>, longest firing duration of target j is 4 and that of target k is 3. If target j and k are sequenced as shown in the figure, the firing duration of combined targets j and k is 5. Thus, the saving period of this example is 2 (4+3-5). Obviously, the TS_{kj} which is the saving period of combined target of target k first and j second is 0. Also, the saving period is calculated with the formulation as in notation. That is $\max_{i \in W(j)} (d_{ij})$ is 4, $\max_{i \in W(j,k)} (d_{ij})$ is 2, and $\max_{i \in W(k)} (d_{ik})$ is 3. Thus TS_{jk} is equivalent to min{4-2, 3}.



<Figure 1> A Simple Example for Solution Representation



<Figure 2> A Simple Example for Saving Period

Now, we present an optimal solution property of two-weapon problem in which there are two weapons.

Proposition 1 *Consider two targets in two-weapon problem : a two-weapon target j and single-weapon target k. If* $TS_{jk} = \max_{i \in W(j)}(d_{ij}) - \max_{i \in W(j,k)}(d_{ij}) = \max_{i}(d_{ik})$ *is satisfied, then in an optimal solution of the problem, target k is immediately sequenced after target j.*

Proof : Suppose there are two single-weapon targets k and k' which are assigned to weapon i. Let S be a sequence that satisfy the sequencing rule, and S' be a sequence that is identical to S except that the position's of targets k and k' are interchanged. We will show that the makespan of the sequence S is at least equal to that of the sequence S'. We consider two cases related to the durations of two targets : (Case 1) $d_{ik} < d_{ik'}$ and (Case 2) $d_{ik} > d_{ik'}$.

(Case 1) Note that, $TS_{jk} = TS_{jk'}$ and comparing to the contribution of the targets j and k to the makespan in sequence S, that of targets j and k' in sequence S' increases by the difference of two durations ($d_{ik} - d_{ik'}$). In addition, if the makespan of sequence S' is determined by weapon i, then there is no makespan difference between sequence S and S'. However, if the makespan of sequence S' is determined by the other weapon, then makespan of S' is increased by ($d_{ik'} - d_{ik}$).

(Case 2) Note that, $TS_{jk} > TS_{jk'}$ due to $d_{ik} > d_{ik'}$ and the contribution of the targets j and k to the makespan in sequence S is equal to that of targets j and k' in sequence S'. In addition, if the makespan of sequence S' is determined by weapon i, then makespan of S' is increased by ($d_{ik} - d_{ik'}$). However, if the makespan of sequence S' is determined by the other weapon, then there is no makespan difference between sequence S and S'. This completes the proof. ■

The following proposition gives a dominance property related to a target which is supposed to be appended in partial sequence σ.

Proposition 2 Considering partial sequence σ, if an unscheduled target j can be inserted idle times between two sequenced targets (implying that the target j can be positioned at earlier period than the time according to the schedule representation scheme) without changing other sequenced targets' schedule, then the current partial solution is dominated.

Proof : Consider a sequence, S, in which job k is scheduled at t_k. according to the schedule representation scheme. Consider a sequence, S', that is identical to S except that the job k is inserted between two sequenced jobs (positioned earlier period). We will prove this proposition by showing that the completion time of sequences oriented from sequence S' is at least as good as that of sequences oriented from sequence S. Observe that sequences originated from S can be exactly replicated with sequences originated from S' except job k. Furthermore, after insertion of job j at earlier position, the completion time of sequences originated from S' may be decreased. ■

This dominance condition can be applied when an unscheduled job can be inserted idle times between two sequenced targets of partial sequence σ. If there exists such an unscheduled job, then σ can be deleted from further consideration in the B&B algorithm.

Proposition 3 *Consider two unscheduled targets, targets j and k. Assume that they are incompatible,* $TS_{jk} > TS_{kj}$, *and firing start time of target j of* σ_{jk} *is same as that of k of* σ_{kj}. *If all unscheduled targets except target j and k are assigned to weapon* w_{kj}, *then* σ_{jk} *dominates* σ_{kj}.

Proof : Since $TS_{jk} > TS_{kj}$, the latest completion time of σ_{jk} is strictly less than that of σ_{kj}. In any schedule generated from partial schedule σ_{kj}, if the weapon w_{kj} is commonly assigned to all unscheduled targets, then the earliest start time of any unscheduled targets is the latest completion time of partial schedule σ_{kj}. However, in any schedule generated from partial schedule σ_{jk}, the earliest start time of any unscheduled targets is the completion time of σ_{jk} which is less than that of σ_{kj}. Thus, σ_{jk} *dominates* σ_{kj}. ■

This dominance condition can be applied when the target (or job) k is placed at the last position of the current partial sequence, σ. If there is another target j not in σ which is incompatible with the target k and target j and k satisfy the conditions in the proposition 2.2, then a sequence composed of σ followed by the target j can be deleted from further consideration in the B&B algorithm. The following proposition is applied when two unsequenced targets are assigned to all weapons.

4. Branch and Bound Algorithm

In the branch and bound (B&B) algorithm described next, each node of the tree corresponds to a sub-problem, which is defined by a partial sequence of jobs. We call the set of jobs not included in partial sequence as un-sequenced jobs. At the beginning of the B&B algorithm (root node), there is an empty sequence of jobs and when the algorithm terminates we gain a complete sequence of jobs such that the latest completion time is minimized. a node at level l in the branching tree represents a partial sequence in which l jobs are scheduled. At level 1, n nodes are created, which n is the number of jobs. For each node in this level, a specific job is sequenced to the 1st position in the partial sequence. At level 2, $n(n\text{-}1)$ nodes are created. This means that for each level 1 node there are $(n\text{-}1)$ un-sequenced jobs. In this way, the lower level nodes are generated and the maximum number of possible nodes is $n+n(n\text{-}1)+n(n\text{-}1)(n\text{-}2)+\cdots+ n(n\text{-}1)(n\text{-}2)\cdots(2)$. According to the solution representation scheme used in this study, the B&B tree is constructed in detail as follows. A job included in a partial sequence corresponding to a node closer to the root node are placed in an earlier position in the sequence than those corresponding to its child nodes farther from the root node. The selection of a node to branch from follows the depth first rule, which is a node with the most jobs included in the associated partial sequence is selected for branching. In case of ties, a node with the minimum lower bound is selected. When a node is branched, one or more nodes are generated by appending one more job at the end of the partial sequence associated with the node being branched. When child nodes are generated from a selected parent node, we check redundancy of (partial) sequences as described in the solution representation scheme.

<Table 1> Results of the CPLEX and B&B Algorithm

		CPLEX		B&B	
W[a]	T[b]	CPUT[c]	NPNS[d]	CPUT	NPNS
2	10	48.98	0	7.84	0
	11	406.44	0	19.65	0
	12	1843.52	4	56.89	0
	13	3175.71	11	989.71	3
	14	3600.02	14	634.35	2

[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.

That is, if two targets are compatible, we only consider a sequence in which the job with an operation assigned to the lowest-indexed weapon is placed earlier. If a node is related with a sequence that does not satisfy this rule, the node is deleted from consideration (pruned) in the B&B procedure. When a node is generated, the dominance conditions presented earlier are checked, and nodes corresponding to dominated sequences are also pruned. For each child node that is not pruned, a lower bound on the makespan is calculated. A node is fathomed if its lower bound is greater than the current incumbent solution.

***Upper bound* (*Heuristic-H*)**

To create a sequence in the branch and bound (B&B) algorithm, four local search algorithms are suggested in this study. In these algorithms, we construct initial solution with NEH algorithm proposed by Nawaz et al. [24], and a set of neighborhood solutions are explored for the current solution and the best neighborhood solution is selected as the new current solution. To explore neighborhood solutions (sequences) from the current solution, we use two methods, insertion and interchange. The insertion method starts by selecting a job in the current sequence and inserting this job into all possible positions in the sequence. On the other hand, the interchange method starts by selecting a job in the current sequence and exchanging its position with each of all other jobs in the sequence. We employed four local search algorithms associated with insertion and interchange: *IS*, *IC*, *ISIC*, and *ICIS*. The *IS* and *IC* algorithm generate neighborhood solutions with insertion and interchange method, respectively. The ISIC and ICIS algorithms in which *IS* and *IC* are executed sequentially and *IC* and *IS* are executed sequentially, respectively. Since it is expected to take very short time to solve a problem with the heuristic algorithm, we execute both of *ICIS* and *ISIC* and select the better solution as the initial upper bound in our B&B algorithm.

A lower bound

A lower bound is obtained for each node that cannot be eliminated by the dominance properties. The lower bound associated with partial sequence σ is maximum value of summing, for each weapon, (1) completion time of sequenced jobs in σ and (2) total duration of each firing operation of un-sequenced jobs (jobs not included in partial sequence σ). The lower bound is computed as

$$\max_{\forall i}\{c_i + \sum_{j \in U} d_{ij}\}$$

where c_i is the completion time of weapon i with partial sequence σ and U is the set of un-sequenced targets. The sum of durations of un-sequenced jobs can be regarded as the minimum total firing duration of un-sequenced jobs.

5. Computational Experiments

First, to test the performance of the proposed algorithm, we use the branch-and-bound algorithm built in ILOG CPLEX 11.0. For the test, we generated 75 problems, 5 problems for each of all combination of one level for the number of weapon (2), 5 levels for the number of targets (from 10 to 14), three levels for the percentage of single-weapon target among all targets (20%, 50%, and 80%). All tests were performed on a personal computer with a Intel® CoreTM i5 CPU at 3.2GHz clock speed.

Results of the test are given in <Table 1>, which shows the average computational time and number of problems that were not solved to the optimality in time limit. As can be seen from the table, the B&B algorithm performs better than CPLEX with respect to both CPUT and NPNS. Thus, we can conclude that ILOG CPLEX takes much more computational time to solve a problem than the suggested B&B algorithm.

Before evaluating the performance of the B&B algorithm, we test the effectiveness of the dominance properties, the lower bound, and the heuristic algorithm used in the B&B algorithm. For this test, we compare four B&B algorithms : BB with all of these included; BB-DP, the B&B algorithm without dominance properties, i.e., the one with the lower bound and the heuristic algorithm only; BB-LB, the one without the lower bound; and BB-H, the one without the heuristic algorithm. Since it takes very short time to solve a problem with the heuristic algorithm, we execute both of ICIS and ISIC and select the better solution as the initial upper bound in our B&B algorithm. It was tested on 30 randomly generated test problems, 5 problems for each of all combinations of two levels for the number of weapons (3 and 6), three levels for the number of targets (10, 12, and 14). The number of weapons needed for a target was generated from the uniform distribution with range [40%, 80%] of weapon number and rounded down for the integrality. The assigned number of targets to each weapon is balanced among different weapons. The durations (d_{ij}) of the firing operations against targets were generated from the discrete uniform distribution with rang [2, 4]. The execution time for each problem was limited to 3,600 seconds.

Results of the test are shown in <Table 2>, which shows the average CPU time, the number of problems that were not solved to the optimality in 3600 seconds, and the average ratio of the numbers of nodes generated to that of nodes that would have been generated if none of the dominance properties, the lower bound and the heuristic algorithm had been used in the B&B algorithm. If the algorithm could not solve a problem within 3,600 seconds, we calculate the CPU time of the problem as 3,600 seconds. For a better exposition of the results, we prepared another table, <Table 3>, which shows the ratios of the CPU times and the ratios of the numbers of nodes generated in the B&B procedures. The lower bound was more effective than the dominance properties and the heuristic in reducing the computation time

<Table 2> Performance of the B&B Algorithms

W[a]	T[b]	BB			BB-DP			BB-LB			BB-H		
		CPUT[c]	NPNS[d]	RNN[e]	CPUT	NPNS	RNN	CPUT	NPNS	RNN	CPUT	NPNS	RNN
3	10	0.2	0	3.8×10^{-7}	0.2	0	3.8×10^{-7}	22.4	0	2.3×10^{-2}	0.2	0	2.4×10^{-5}
	12	0.2	0	2.9×10^{-9}	0.2	0	2.9×10^{-9}	901.6	0	7.7×10^{-3}	0.2	0	2.2×10^{-7}
	14	0.2	0	1.6×10^{-11}	0.2	0	1.6×10^{-11}	3600.0	5	1.5×10^{-4}	0.8	0	1.1×10^{-8}
6	10	0.8	0	2.1×10^{-4}	0.4	0	3.1×10^{-4}	12.2	0	1.1×10^{-2}	0.8	0	2.4×10^{-4}
	12	7.0	0	1.4×10^{-5}	2.9	0	2.8×10^{-5}	263.1	0	1.7×10^{-3}	6.8	0	1.4×10^{-5}
	14	119.4	0	9.3×10^{-7}	97.4	0	3.9×10^{-6}	3186.6	3	1.1×10^{-4}	124.4	0	1.0×10^{-6}

[a,b] number of weapons/number of targets.
[c] (lower bound on the) average CPU time (in seconds) required to solve a problem.
[d] number of problems (among 5 problems) that were not solved to the optimality in 3,600 seconds.
[e] average ratio of the number of nodes considered in the B&B procedure to that of nodes that would have been generated if no nodes had been fathomed.

<Table 3> Results of the Comparison of the Algorithms

W[a]	T[b]	Average ratio of CPU times[c]			Average ratio of nodes generated[d]		
		T_{BB-DP}/T_{BB}	T_{BB-LB}/T_{BB}	T_{BB-H}/T_{BB}	N_{BB-DP}/N_{BB}	N_{BB-LB}/N_{BB}	N_{BB-H}/N_{BB}
3	10	1.034	125.978	1.034	1.000	6.044×10^{4}	62.800
	12	1.065	4899.978	1.120	1.000	2.644×10^{6}	74.200
	14	1.039	17647.108	4.039	1.000	9.590×10^{6}	713.400
6	10	1.951	29.727	0.976	1.515	5.109×10^{1}	1.136
	12	2.414	90.710	1.034	2.042	1.281×10^{2}	1.018
	14	1.226	32.717	1.009	4.233	1.139×10^{2}	1.072

[a,b] See the footnotes of <Table 2>.
[c] TX denotes the CPU time required to solve a problem using algorithm X
[d] NX denotes the number of nodes considered for a problem in algorithm X

as well as in fathoming nodes in the B&B algorithm. However, use of the dominance properties and the heuristic algorithm reduced the CPU time although the reduction was not very impressive. This means that the benefit of using these, i.e., pruning dominated partial solutions, exceeds the cost of using them, i.e., the time required for checking the dominance conditions or for obtaining a good initial upper bound. In the following, we further test BB, the B&B algorithm with the dominance properties and heuristic algorithm as well as the lower bound.

For this main test, 60 problems were additionally generated, 5 problems for each of all combination of three levels for the number of weapons (3, 6, and 9), four levels for the number of targets (16, 18, 20, and 22). As in the previous test, the upper limit of 3600 seconds is given for each problem. Results of the test are given in <Table 4>. The table shows the average CPU time and the number of problems which were not solved to the optimality within the time limit as well as average percentage gap of the heuristic solutions from the BB solutions and the number of problems the heuristic algorithm gave the same solutions with BB solutions.

The B&B algorithm solved to the optimality 42 problems out of 60 within 3,600 seconds. The CPU time was affected more by the number of weapons than the number of targets. This may be because, if there are more weapons (when the number of target is given), the weapon-target allocation graph becomes sparse in general. This results in weaker lower bound and takes longer time when the number of weapons is larger. The heuristic algorithm worked relatively well. In 20 problems it gave the same solutions as those obtained from the B&B algorithm, and the overall average percentage gap of the heuristic solution (from the solutions of BB) was 5.10%.

<Table 4> Results of the Main Test

W[a]	T[b]	B&B algorithm		Heuristic algorithm	
		CPUT[c]	NPNS[d]	APG[e](%)	NOBS[f]
3	16	720.2	1	0.95	4
	18	0.3	0	1.25	4
	20	0.3	0	0.00	5
	22	0.4	0	0.00	5
6	16	599.4	0	10.56	1
	18	2225.5	2	5.58	1
	20	952.1	0	8.96	0
	22	2205.2	3	8.05	0
9	16	815.9	0	5.55	0
	18	2327.5	2	5.98	0
	20	3600.0	5	8.66	0
	22	3600.0	5	5.61	0
Average		1420.6	1.5	5.10	1.7

[a,b] See the footnotes of <Table 2>.
[c] (lower bound on the) average CPU time (in seconds) required to solve a problem.
[d] number of problems (among 5 problems) that were not solved to the optimality in 3,600 seconds.
[e] average percentage gap of the heuristic solutions from the solutions of the branch and bound algorithm.
[f] number of problems (among 5 problems) for which the heuristic algorithm gave the optimal or best solutions, i.e., the same solutions as those from the BB algorithm.

6. Summary

This research considered the problem of scheduling fire operations with the objective of minimizing makespan. Several dominance properties and a lower bound on the latest completion time for a partial schedule were found and used to develop a branch and bound algorithm. In addition, heu-

ristic algorithms that can be used for obtaining an initial upper bound in the B&B algorithm and for obtaining good solutions in a short time were developed. The proposed lower bound, the dominance properties, and the heuristics for upper bound are tested in B&B respectively, and the result showed that lower bound is effective to fathoming nodes and the dominance properties and heuristics also worked well. Also, it is showed that the CPU time required by this algorithm increases rapidly as the problem size increases. Therefore, the suggested B&B algorithm would be limited to solve large size problems. However, the employed heuristic algorithms can be effectively used in the B&B algorithm and can give good solutions for large problems within a few seconds.

References

[1] Ahn, N.-S., Jeong, J.-S., Jeong, W.-K., Hwang, W.-Y., and Park, S.-W., Sampling Procedures Enhancement in Government Defense Quality Assurance Procedures : Case Studies in Combat Force Support Material and Ammunition Areas. *Journal of the Korean Society for Quality Management*, 2012, Vol. 40, No. 3, pp. 245-258.

[2] Ahn, N.-S., Park, S.-W., Chae, J.-M., Lee, Y.-W., and Oh, B.-I., Suggestion of Using Defense Quality Score based on Taguchi Loss Function in Korea Defense Area. *Journal of the Korean Society for Quality Management*, 2013, Vol. 41, No. 3, pp. 443-456.

[3] Amoura, A.K., Bampis, E., Kenyon, C., and Manoussakis, Y., Scheduling independent multiprocessor tasks. *Algorithmica*, 2002, Vol. 32, pp. 247-261.

[4] Bianco, L., Dell'Olmo, P., and Speranza, M.G., Nonpreemptive scheduling of independent tasks with prespecified processor allocations. *Naval Research Logistics*, 1994b, Vol. 41, pp. 959-971.

[5] Blazewicz, J., Dell'Olmo, P., Drozdowski, M., and Speranza, M., Scheduling multiprocessor tasks on three dedicated processors. *Information Processing Letters*, 1992, Vol. 41, pp. 275-280.

[6] Bozoki, G. and Richard, J.P., A branch and bound algorithm for the continuous-process job-shop scheduling problems. *AIIE Transactions*, 1970, Vol. 2, pp. 246-252.

[7] Brucker, P. and Kramer, A., Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems. *European Journal of Operational Research*, 1996, Vol. 90, pp. 214-216.

[8] Cha, Y.-H. and Kim, Y.-D., Fire scheduling for planned artillery attack operations under time-dependent destruction probabilities. *Omega*, 2010, Vol. 38, No. 5, pp. 383-392.

[9] Chen, J. and Lee, C.-Y., General multiprocessor tasks scheduling. *Naval Research Logistics*, 1999, Vol. 46, pp. 57-74.

[10] Chen, J. and Miranda, A., A polynomial time approximation scheme for general multiprocessor job scheduling. *SIAM Journal on Computing*, 2001, Vol. 31, pp. 1-17.

[11] Dell'Olmo, P., Speranza, M.G., and Tuza, Z., Efficiency and effectiveness of normal schedules on three dedicated processors. *Discrete Mathematics*, 1997, Vol. 164, pp. 67-79.

[12] Drozdowski, M., Scheduling multiprocessor tasks-An overview. *European Journal of Operational Research*, 1996, Vol. 94, pp. 215-230.

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

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

[15] Hoogeveen, J.A., Van de Velde, S.L., and Veltman, B., Complexity of scheduling multiprocessor tasks with prespecified processor allocations. *Discrete Applied Mathematics*, 1994, Vol. 55, pp. 259-272.

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

[17] Jansen, K. and Porkolab, L., General multiprocessor task scheduling : approximate solutions in linear time. *SIAM Journal on Computing*, 2005, Vol. 35, pp. 519-530.

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

[19] Kim, T.-H. and Lee, Y.-H., Fire sequencing problem with shared targets. *Korean Operations Research and Management Society*, 2003, Vol. 28, No. 3, pp. 123-134.

[20] Kramer, A., Branch and bound methods for scheduling problems with multiprocessor tasks on dedicated processors. *OR Spektrum*, 1997, Vol. 19, pp. 219-227.

[21] Kubale, M., The complexity of scheduling independent two-processor tasks on dedicated processors. *Information*

Processing Letters, 1987, Vol. 24, pp. 141-147.

[22] Kwon, O.-J., Lee, K.-S., and Park, S.-S., Targeting and scheduling problem for field artillery. *Computers and Industrial Engineering*, 1997, Vol. 33, pp. 693-696.

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

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

[25] Shin, S.-S., Kim, B.-K., and Sim, C.-B., A Study on Crack Formation in the K11 Objective Individual Combat Weapon Fire Control System using Analysis of Variance. *Journal of the Korean Society for Quality Management*, 2015, Vol. 43, No. 3, pp. 67-73.

[26] Yoon, S.-H., Hwang, W.-S., Juhn, J.-H., and Lee, I.-S., A branch-and-bound algorithm on the fire sequencing for planned artillery operations. *Journal of the Society of Korea Industrial and Systems Engineering*, 2010, Vol. 33, No. 3, pp. 154-161.

ORCID

Young-Ho Cha | http://orcid.org/0000-0003-3301-9587

June-Young Bang | http://orcid.org/0000-0003-4999-9161