

## 조선 공정계획에서 탑재 순서 생성\*

홍윤기\*\* · 정은경\*\*\* · 전 진\*\*\* · 김세영\*\*\*

eneration of Erection Sequence in Shipbuilding Process Planning

oon Gee Hong · Eun Kyung Jung · Jin Jun · Se Young Kim

### 〈ABSTRACT〉

This paper explains a way of describing the erection sequence using the connecting relationship among blocks of a hull in shipbuilding industry. The information of the adjacent blocks such as upper, lower, right, left, front, rear, and their erection status, etc., is defined and the proper use of this allows us to determine one of the possible erection sequences. The methodology suggested also includes some useful benefits of avoiding insert blocks or maintaining stability in erection process. Further, an algorithm for generating the initial erection network is shown and this network may be a basis for designing daily erection scheduling. Some examples illustrate the procedure developed in this study and the results support the model's fidelity.

### 1. 서론

조선업은 많은 부분이 숙련된 기술자와 고부가가치를 수반하는 첨단 설비를 총동원하는 노동 및 기술 집약적 산업[1]으로 국민 경제에 미치는 영향이 지대한 산업 중 하나이다[2].

오늘날 조선업이 국가 경쟁력 우위를 차지하기 위해서는 각 조선공정의 자동화가 요구되고 있는데 이에 대한 연구로는 다음과 같다. CAD/CAM정보 및 기술을 이용하여 설계 단계에서부터 가능한 완전히 통합된 시스템의 제안[6]과 컴퓨터 지원에 의한 조립 공정에 관한 연구[7]를 찾아볼 수 있다. 또한, 수주에서 인도에 이르기까지 모든 기업 활동의 계획, 실행, 관

리에 대한 정보를 통합하여 필요한 정보를 적시 적소에 능률적으로 공급하는 방법으로 선박을 건조하고자 하는 조선 CIM구축에 대한 연구도 최근 발표되었다 [1].

본 연구는 선박 탑재 공정 계획에서 초기 탑재 네트워크 구성 및 탑재 순서 계획에 관한 내용을 그 골자로 하고 있다.

선박 공정 중에서 탑재 공정은 조립장소나 크레인 능력을 고려하여 선체를 여러 개의 블록으로 나누어 작업을 한다[3]. 즉, 내업에서 소조립, 중조립, 대조립 과정을 거쳐 하나의 블록 단위를 제작하고, 블록들을 선행의장, 선행도장 및 P.E.(Pre Erect)를 한 후 Dock 및 선대에서 순서대로 탑재하는 것은 외업공정에서

\* 이 연구는 한국기계연구원 위탁 과제임.

\*\* 한성대학교 산업시스템공학부

\*\*\* 고려대학교 산업공학과

이루어진다[1]. 이 탑재 공정은 타 건조계획과의 관계에서 보면 장기 부하 평준화와 조업계획에 근거한 Dock 사용계획인 기본선표를 작성하고 각 선의 착공 및 Dock에서의 기공, 진수일자, 안벽 의장기간 등을 정하면 Dock 사용기간에 준하여 선별 블럭 탑재 계획이 수립되는데 이 탑재 계획은 일반적인 타 건조공정의 기준이 되는 것으로 그 일정을 기준으로 전후 각 공정이 계획된다.

이처럼 탑재 공정이 기타 공정에 미치는 영향이 과중하기 때문에 블럭의 탑재 순서 결정은 선박건조계획의 중일정계획에서 가장 먼저 수립되어야 할 중요한 요소이다. 지금까지 현업에서는 탑재 순서를 결정하는데 경험을 토대로 한 표준치를 이용하거나 선형/선종별로 기존의 탑재 순서를 그대로 사용하는 경우가 많았다. 이런 경우 Dock상황 또는 Batch건조 상황 등 변화하는 상황에서 실제 블럭 탑재시 바람직한 결과를 얻으리라는 보장이 없게 된다.

탑재 순서를 결정하는데 있어서 이처럼 기존의 Data나 경험을 이용하는 가장 큰 이유는 가능한 탑재 순서를 모두 고려하기에 그 경우의 수가 너무 많기 때문이라는 지적이다. 여기서 주관적인 요소를 배제한 탑재 순서를 수작업으로 생성하기에 곤란한 점이 많기 때문에 시뮬레이션 기법을 이용하여 다양한 상황에서 가능한 한 다양한 탑재 순서를 자동으로 생성하는데 본 연구의 목적이 있다.

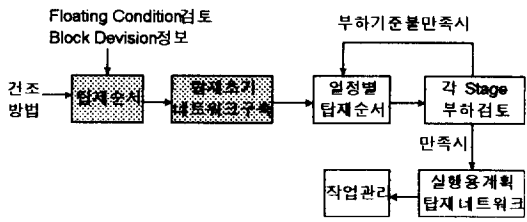
본 연구의 범위는 선박 건조에 관한 전체 업무 Flow를 나타낸 <그림 1> 가운데 빗금으로 표시된 부분에 해당한다.

기존 연구 중에는 Goliath Crane의 제약으로 인하여 블럭이 한번에 하나씩만 탑재 가능하다는 점과 가능한 탑재 Sequence의 수가 매우 많다는 점, 그리고 가능한 탑재 Sequence를 도출하는 과정에 탑재 기술적인 측면에서의 전문가 지식이 존재한다는 사실에 착안하여 제약조건하의 그래프 탐색기법(Constraint Directed Graph Search)을 이용한 탑재 일정계획 시스템을 제안하였다[4,8,9].

또한, 블럭간 접합 정보를 가지고 초기 순서(Sequence)를 생성하고, 중복된 순서를 삭제한 후, 탑재 공법 데이터를 가지고 가상 액티비티를 추가해서 탑재 순서를 생성하는 연구[5]와 블럭간 접속관계에 따라 탑재 방향을 생성하는 Path Generator와 사용자 규칙에 따라 탑재 순서를 만드는 Adder, 사용자 규칙에 따라 탑재 순서를 삭제하는 Deletor를 가지고 탑재 순서를 생성하는 방안[1]을 제안하고 있다.

본 연구는 pitch 개념을 배제한 상태에서 각 블럭의 접속 관계만으로 탑재 순서를 생성하는 단계와 탑재 순서 생성 방안의 결과를 약간 변형하여 초기 탑재 네트워크 생성 방법을 제시하는 두 단계로 이루어져 있다.

첫번째 단계인 탑재 순서 생성의 기본 개념은 탑재 조건에 맞는 블럭을 찾아 탑재하고 기준에 의해 탑재 상태 여부를 나타내는 블럭 VALUE를 수정한 후, 이어서 탑재가 가능한 후보 블럭을 찾아내는 과정을 반복하여 탑재 순서를 생성한다. 두번째 단계인 탑재 초기 네트워크와 관련하여 현업에서는 주로 1) 기존 표준 데이터와 경험을 이용하는 방법 또는 2) 사용자가 몇 개의 블럭간 순서를 지시해 줌으로써 생성하게 하는 방법을 사용하고 있다. 본 연구에서는 탑재 순서 생성시와 마찬가지로 개념으로 매 단계마다 탑재 가능 블럭을 node로 추가하여 네트워크를 확장해 나가는 방법을 적용한다. 또한 탑재 순서 생성과 초기 탑재 네트워크 구축 방법에 대한 간단한 적용례를 살펴보기로 한다.



<그림 1> 선박 건조 계획 업무 Flow

## II. 탑재 순서 생성

### 1. 전체 구성

블럭의 탑재 순서 생성은 블럭간 선후 관계를 고려하여 인접 블럭의 탑재 상태에 대한 정보를 가지고 가능 조건을 만족하는 블럭을 찾아 탑재 순서를 생성한다. 인접 블럭의 정보를 나타내기 위하여 블럭에 관한 정보 입력시 인접 블럭과 인접 블럭의 탑재 여부를 표시하는 블럭 VALUE의 개념을 이용한다. 선박의 구조는 3차원 즉, 상하, 좌우, 전후의 블럭 형태를 이루고 있기 때문에 일반적으로 인접 블럭 중 하측 블럭, 좌우블럭 중 한쪽 블럭, 전후블럭 중 한쪽 블럭 등의 세 가지 탑재 상태를 만족하는 블럭이 탑재 가능한 블럭이 된다. 탑재 조건을 가능하는 형식을 벗어난 경우에는 가상 블럭 VALUE의 개념을 도입해서 일반적인 탑재 가능 조건을 만족시키도록 재조정하는 절차가 필요하다.

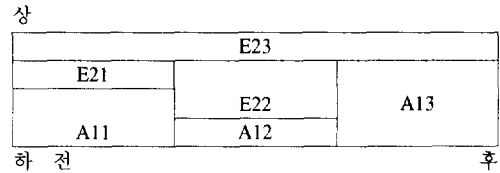
탑재 순서 결정 방법의 기본 골격은 K/L(Keel Laying)블럭을 탑재한 후, 블럭 VALUE 수정 기준에 의해 블럭의 탑재 상태를 나타내는 선후 블럭의 VALUE를 수정하고 현 시점에서 탑재 조건을 만족하는 탑재 가능한 블럭을 찾게 된다. 탑재 가능한 블럭이 여러 개인 경우에는 입력순으로 블럭을 선택하여 차례로 탑재 순서를 생성하도록 하였다. 이와 같이 하나의 완전한 탑재 순서가 생성된 후에 첫번째 복수 대안이 발견될 때까지 backtracking(역추적)을 실시하여 이 시점까지 탑재된 블럭들을 기반으로 해서 이전 경로와는 다른 블럭을 탑재함으로써 이전과 다른 탑재 순서의 생성이 가능하게 된다. 이런 과정을 반복하여 가능한 모든 탑재 순서를 생성하는 것을 기본으로 하고 있다.

### 2. 블럭 입력 내용 및 세부 사항

#### (1) 블럭 명칭

블럭의 명칭을 부여하는데 있어서 기존 현장에서 사용하는 블럭 명칭을 그대로 사용하고 P.E블럭은 하나의 블럭으로 간주하여 블럭 명칭을 표기한다. 예를

들어서, <그림 2>는 선박 구조의 일부 형태를 나타낸 것으로 블럭 명칭은 임의로 정한 것이다. 여기서 블럭의 명칭은 탑재 순서 결정에 아무런 영향을 미치지 않는다.



<그림 2> 블럭 CODE

#### (2) 입력 내용

##### 1) 블럭 형태

임의의 단위 블럭은 상하, 좌우, 전후에 여섯 방향의 인접 블럭을 가질 수 있다. 기준 블럭의 위쪽에 해당하는 상측 블럭, 아래에 해당하는 하측 블럭, 왼쪽에 해당하는 좌블럭, 오른쪽에 해당하는 우블럭, 앞에 해당하는 전블럭, 뒤쪽에 해당하는 후블럭이 자리하게 된다.

##### 2) 입력 형식

입력 형식의 기본 형태는 해당 블럭명, 블럭의 상하, 좌우, 전후 블럭을 나타내는 선후 블럭, 선후 블럭의 탑재 상태를 나타내는 블럭 VALUE로 표현되어 있다. <표 1>에서 블럭명의 괄호 안의 숫자는 입력 순서를 나타낸다. 입력 순서는 여러 개의 블럭이 탑재 가능할 때 입력순으로 탑재하도록 하는 규칙이 적용되기 때문에 탑재 순서 결정에 직접적인 영향을 미치게 된다. 블럭 VALUE는 해당 선후 블럭의 탑재 상태에 따라 0에서 1 사이의 값을 갖는다.

##### 3) 초기값

가상 좌블럭 VALUE와 가상 전블럭 VALUE가 모든 블럭에 초기값으로 주어졌다. 이것은 하측 블럭만 탑재되면 어떠한 블럭이라도 탑재 가능한 블럭으로 만들어 주기 위함이다. 가상 좌블럭 VALUE와 가상 전블럭 VALUE는 실제로 선후 블럭의 좌블럭과 전블

력이 탑재되지 않았지만 탑재된 것으로 가정해서 탑재 조건을 만족시키게 된다. 이와 같은 탑재 방식을 채택하면서 삽입 블록을 방지하기 위해 탑재 블록의 전후를 따라 가상 전블록 VALUE를 수정하고, 마찬가지로 좌우를 따라 가상 좌블록 VALUE를 수정한 후, 탑재된 블록에 해당하는 인접 블록의 탑재 여부를 나타내는 블록 VALUE를 수정하여 탑재 가능 여부를 결정하게 된다.

〈표 1〉 블록 입력 형태와 초기값

A11블록(1)			A12블록(2)		
블록구분	선후블록	VALUE	블록구분	선후블록	VALUE
상측블록	E21		상측블록	E22	
하측블록	-		하측블록	-	
가상좌블록		1	가상좌블록		1
좌블록	-		좌블록	-	
우블록	-		우블록	-	
가상전블록		1	가상전블록		1
전블록	-		전블록	A11	
후블록	A12,E22		후블록	A13	

A13블록(3)			E21블록(4)		
블록구분	선후블록	VALUE	블록구분	선후블록	VALUE
상측블록	E23		상측블록	E23	
하측블록	-		하측블록	A11	
가상좌블록		1	가상좌블록		1
좌블록	-		좌블록	-	
우블록	-		우블록	-	
가상전블록		1	가상전블록		1
전블록	A12,E22		전블록	-	
후블록	-		후블록	E22	

E22블록(5)			E23블록(6)		
블록구분	선후블록	VALUE	블록구분	선후블록	VALUE
상측블록	E23		상측블록	-	
하측블록	A12		하측블록	A13,E21,E22	
가상좌블록		1	가상좌블록	-	1
좌블록	-		좌블록	-	
우블록	-		우블록	-	
가상전블록		1	가상전블록		1
전블록	A11,E21		전블록	-	
후블록	A13		후블록	-	

3. 탑재 순서 결정 절차

(1) 입력 자료

탑재 순서를 결정하기 위한 입력 자료로는 블록명,

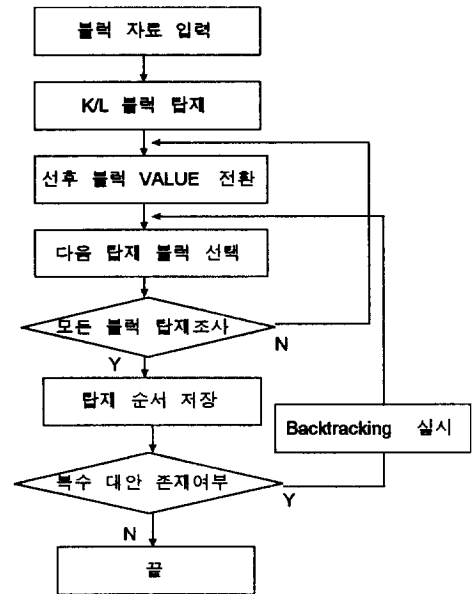
그 블록의 인접 블록 개수와 인접 블록명이 필요하다.

(2) 출력 자료

본 절차에서는 Pitch라 불리는 블록간 소요 기간의 개념 없이 순수하게 블록간 접속 관계만 고려했을 때 모든 가능한 탑재 순서가 얻어지게 된다.

(3) 절차

탑재 순서를 결정하는 방안을 그림으로 나타내면 〈그림 3〉과 같으며 이를 구현하는 Algorithm을 간단히 정리하면 다음과 같다.



〈그림 3〉 탑재 순서 생성 절차

〈탑재 순서 결정 Algorithm〉

단계 0 (K/L 블록 선택)

블록 자료를 입력하고 K/L블록을 선택하여 K/L블록을 탑재시킨다.

단계 1 (블록 VALUE 수정)

블록이 탑재되면 정해진 기준에 의해 블록 VALUE를 수정한다.

단계 2 (답재 순서 결정)

여러 개의 블록이 답재 가능할 때 입력 순서에 의해 단 한개의 블록을 선택한다. 선택된 블록을 답재하고 모든 블록이 답재될 때까지 단계 1과 2를 반복한다.

단계 3 (답재 순서 저장 및 답재 상태 제거)

모든 블록이 답재되면 답재 순서를 저장하고 역으로 첫번째 복수 대안이 나오는 곳까지 backtracking을 실시한다. (depth-first search)

단계 4 (복수 대안)

먼저 선택한 블록을 제외하고 입력 순서가 가장 빠른 블록을 선택하여 답재한 후 단계 1, 단계 2, 단계 3을 반복한다.

단계 5

모든 복수 대안이 찾아진 경우 종료한다.

4. 답재 블록 선택 및 블록 VALUE 수정

(1) 답재 블록 선택 방법

블록 답재시 삽입 블록을 방지하기 위해서는 답재 가능 블록을 선택할 때, 그 블록의 하측블록, 전후블록 중 한 블록, 좌우블록 중 한 블록이 답재되어 있는 블록을 선택해야 한다. 본 연구에서는 답재 상태를 나타내는 아래의 세 가지 조건 모두를 만족하는 경우가 이에 해당한다. 각각의 숫자는 답재 여부를 나타낸다.

- i) 하측 블록의 VALUE: 1  
숫자 1은 하측에 인접한 모든 블록이 답재되었음을 의미한다.
- ii) 좌블록 VALUE + 우블록 VALUE + 가상 좌블록 VALUE: 1  
인접 블록의 개수가 여러 개인 경우가 생길 수 있으므로 인접 블록 모두가 답재되었을 경우 블록 VALUE가 1이 된다. 즉,  
{좌블록 VALUE + 가상 좌블록 VALUE = 1} .OR.  
{후블록 VALUE + 가상 좌블록 VALUE = 1}이 된다.  
답재되지 않은 블록은 본 연구에서 제시한

Algorithm에 의해 선후블록의 답재 상태를 의미하는 좌블록 VALUE와 우블록 VALUE가 동시에 값을 가지면 삽입 블록이 생기게 되므로 이런 경우가 발생되지 않아야 한다. 결국 상호 배타적으로 좌블록 VALUE와 우블록 VALUE중 하나의 블록 VALUE만 값을 가질 수 있다.

- iii) 전블록 VALUE + 후블록 VALUE + 가상 전블록 VALUE: 1

이것은 위 ii)의 내용과 같은 형식이다.

K/L블록을 결정하면 하블록 VALUE에 1이 부여되어 답재 가능한 블록이 되고 답재 가능 블록인 K/L블록이 답재된다.

(2) 블록 VALUE 수정

한 개의 블록이 답재되면 블록 답재 상태를 표시하는 블록 VALUE를 수정하게 되어 마침내 인접 블록의 답재 상태 여부와 관련한 정보를 표시하게 된다.

- i) 가상 전블록 VALUE와 가상 좌블록 VALUE 수정

답재될 블록의 가상 블록 VALUE가 0이 아닌 동안 일반적인 답재 조건을 만족시키기 위해 부여했던 가상 블록 VALUE를 수정하는 과정을 수행한다. 가상 블록 VALUE가 0이면 이미 일반적인 답재 조건을 만족시키기 위해 부여했던 가상 블록 VALUE가 제거되었기 때문에 더 이상 고려할 필요가 없다.

- ① 가상 전블록 VALUE가 0이 아니면 답재 상태 표시에 상관없이 전, 후블록을 따라 선후 관계로 연결된 모든 블록에 가상 전블록 VALUE를  $\frac{1}{\text{인접 블록 개수}}$  만큼 감한다.
- ② 가상 좌블록 VALUE가 0이 아니라면 답재 표시에 상관없이 좌, 우블록을 따라 선후관계로 연결된 모든 블록에 가상 좌블록 VALUE를  $\frac{1}{\text{인접 블록 개수}}$  만큼 감한다.

- ii) 인접 전, 후, 좌, 우, 상블록 VALUE 수정  
인접 블록의 답재상태를 나타내기 위해 사용된 블록의 블록 VALUE 수정이 필요하다. 답재된 블록의 인접 블록에 각각의 블록 VALUE를  $\frac{1}{\text{인접 블록 개수}}$  만큼 더해서 인접 블록의 답재 상

태를 표시한다.

- ① 전으로 인접한 블럭에는 후블럭이 탑재된 경우이므로 후블럭 VALUE에 더한다.
- ② 후로 인접한 블럭에는 전블럭이 탑재된 경우이므로 전블럭 VALUE에 더한다.
- ③ 좌로 인접한 블럭에는 우블럭이 탑재된 경우이므로 우블럭 VALUE에 더한다.
- ④ 우로 인접한 블럭에는 좌블럭이 탑재된 경우이므로 좌블럭 VALUE에 더한다.
- ⑤ 상으로 인접한 블럭에는 하측블럭이 탑재된 경우이므로 하블럭 VALUE에 더한다.

iii) 탑재 블럭이 1층인 경우 하블럭이 존재하지 않으므로 하블럭 VALUE를 일반적인 규칙에 의해 수정할 수 없다. 이런 경우 특별한 규칙을 적용한다. 즉, 전, 후, 좌, 우 인접한 블럭 중 어느 한 블럭만 탑재될 경우 하블럭이 탑재되었음을 의미하는 하블럭 VALUE를 1로 수정하여 이런 문제를 해결한다.

다음으로 가상 블럭 VALUE의 개념을 설명하기 위해 간단한 예를 들기로 하자.

상				
E21	E22	E23	E24	E25
A11	A12	A13	A14	A15
하 전 후				

〈그림 4〉 블럭 CODE

가상 전·좌블럭 VALUE는 선박의 일반적인 구조인 3차원, 즉 기준 블럭에 대해 인접 블럭이 전후블럭, 좌우블럭, 상하블럭 등 6개가 있는 구조를 만족시키기 위해서 사용했을 뿐만 아니라 〈그림 4〉에서 1층의 모든 블럭 A11, A12, A13, A14, A15블럭이 탑재되었을 경우 2층의 모든 블럭이 탑재 가능한 블럭 조건을 만족시키기 위해 가상 전·좌블럭 VALUE개념을 도입했다. 2층의 어느 한 블럭이 탑재되면 삽입 블럭을 방지하기 위해 인접 블럭만 탑재 가능 블럭으로 만들고 나머지 블럭은 가상 전·좌블럭 VALUE를 기준에 의해서 수정하는 과정이 필요하게 된다.

### 5. 특수한 상황

#### (1) 삽입 블럭 방지

##### 1) 상·하블럭 관계

상	
A21	
A11	A12
하 전	후

〈그림 5〉 블럭 CODE

〈그림 5〉에서 보는 바와 같이 A11블럭이 탑재되고 A21블럭이 탑재되면 A12블럭은 삽입 블럭이 되기 때문에 이런 경우에는 A11블럭과 A12블럭이 탑재된 후 A21블럭을 탑재 가능한 조건으로 만들어 주는 과정이 필요하다. 이것은 A21블럭의 VALUE 수정시  $\frac{1}{\text{인접 블럭 개수}}$ 만큼 더하는 방법을 적용하면 해결될 수 있다. 즉 A11블럭이 탑재되면 A21블럭의 하블럭 VALUE는 0에서  $\frac{1}{2}$ 로 수정되었다. A21블럭은 하블럭 VALUE가 1이 아니기 때문에 아직 탑재 가능한 블럭이 아니다. A12블럭이 탑재된 이후에 비로소 A21블럭의 하블럭 VALUE가 1이 되어 탑재 가능한 블럭이 된다.

##### 2) 전·후블럭 관계

상		
A11	A21	A13
	A12	
하 전		후

〈그림 6〉 블럭 CODE

위의 〈그림 6〉에서도 A11블럭, A12블럭, A13블럭 순으로 탑재하면 A21블럭이 삽입 블럭이 되므로 위의 상하 블럭 관계에서와 같은 방법을 적용해야 한다. 즉 A11블럭이 탑재되었을 경우 A13블럭이 탑재되려면 A12블럭과 A21블럭이 탑재된 이후에 탑재될 수 있다.

(2) 탑재시 안정성 고려

상		
A21	A22	A23
A11	A12	A13
하 전		

〈그림 7〉 블록 CODE

〈그림 7〉을 살펴보면, A12블럭, A22블럭순의 탑재는 탑재의 안정성이 보장되지 않는다. 이런 것을 방지하기 위해 탑재의 안정성을 고려하는 규칙의 적용을 시도해 보기로 한다. 블럭 탑재시 안정성을 고려하여 가능한 한 피라미드 형식으로 탑재를 하기 위해 특별히 다음과 같은 규칙을 적용함이 바람직하다. 하블럭의 VALUE 전환시 제약을 주는 방법으로 해결하고자 한다.

IF ((기준 블럭의 선후 관계에서 후블럭 = 탑재)  
.OR. (선후 관계에서 후블럭이 존재하지 않는 경우.))

기준 블럭에 상으로 인접해 있는 블럭에 하블럭 VALUE를 수정한다.

ELSE IF (기준 블럭의 선후 관계에서 전블럭 = 탑재)  
기준 블럭의 전블럭에 상으로 인접한 블럭의 하블럭 VALUE를 수정한다.

ELSE IF ((기준 블럭의 선후관계에서 전블럭 = 탑재) .AND. (선후관계에서 후블럭이 존재하지 않는 경우.))

- ① 기준 블럭에 상으로 인접해 있는 블럭의 하블럭 VALUE를 수정한다.
- ② 기준 블럭의 전블럭에 상으로 인접한 블럭의 하블럭 VALUE를 수정한다.

ELSE

상으로 인접한 블럭에 하블럭 VALUE를 수정하지 않는다.

위의 상황은 전·후블럭의 탑재 상태만을 참고하여 하블럭 VALUE를 전환했지만 좌·우블럭으로 확장해서 적용이 가능하다. 위와 같이 탑재의 안정성을 고려하다 보면 삽입 블럭이 발생하는 경우와 동시에 중

돌하는 경우가 직면할 수도 있게 된다. 이러한 경우가 발생하면 탑재상 안정성 문제를 배제하고 탑재되도록 한다. 본 연구에서 제시한 Algorithm에서는 탑재의 안정성을 적용하지 않았다.

6. 탑재 순서 결정 방법 적용례

(1) 탑재 순서 결정

상		
E23		
E21	E22	A13
A11	A12	
하 전		

〈그림 8〉 블록 CODE

〈그림 8〉에 나타낸 선박의 일부를 예로 들어 설명하기로 하자. K/L블럭을 선택하면 하블럭 VALUE가 수정되어 탑재 가능한 블럭으로 변하게 되어 탑재된다. 〈표 2〉에서 보는 바와 같이 A11블럭을 K/L블럭으로 가정하여 탑재 순서를 결정하기로 한다.

〈표 2〉 K/L 블럭의 블럭 VALUE전환

블럭구분	선후블럭	VALUE
상측블럭	E21	
하측블럭	-	0->1
가상좌블럭		1
좌블럭	-	
우블럭	-	
가상전블럭		1
전블럭	-	
후블럭	A12,E22	

상		
E23		
E21	E22	A13
A11①	A12	
하 전		

〈그림 9〉 A11블럭 탑재 상태

〈표 3〉 A11블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	0->1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		1->0
전블럭	-		전블럭	A11	0->1
후블럭	A12,E22		후블럭	A13	

A13블럭(3)

E21블럭(4)

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-		하측블럭	A11	0->1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1->0	가상전블럭		1
전블럭	A12,E22		전블럭	-	
후블럭	-		후블럭	E22	

E22블럭(5)

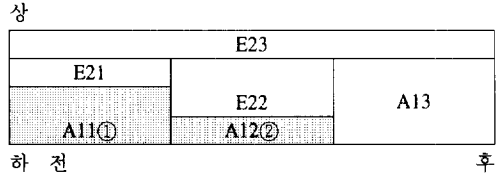
E23블럭(6)

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12		하측블럭	A13,E21,E22	
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1->0.5	가상전블럭		1
전블럭	A11,E21	0->0.5	전블럭	-	
후블럭	A13		후블럭	-	

〈표 3〉에서 보는 바와 같이 A11블럭을 탑재하고 선후 블럭의 블럭 VALUE를 수정한 후 탑재 가능한 블럭들을 찾는다.

- {하측 블럭 VALUE = 1} .AND.
- {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.
- {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 A12블럭과
- {하측 블럭 VALUE = 1} .AND.
- {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.
- {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 E21블럭이 탑재 가능한 블럭이 된다. 두 가지 가능한 대안 중 먼저 입력 순서

가 빠른 A12블럭을 선택하여 탑재하고 복수 대안 LIST {A12\*, E21}에서 탑재 상태를 표시한다. 이를 나타내는 일련의 과정이 〈그림 9〉와 〈그림 10〉에 나타나 있다. (\*: 복수 대안 탑재 블럭 명시)



〈그림 10〉 A12블럭 탑재 상태

〈표 4〉 A12블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-		하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0->0.5	후블럭	A13	

A13블럭(3)			E21블럭(4)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	0->1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A12,E22	0->0.5	전블럭	-0	
후블럭	-		후블럭	E22	

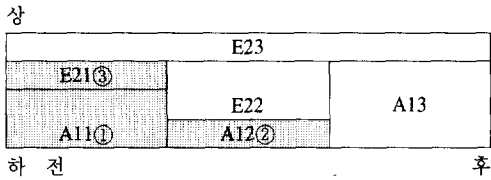
E22블럭(5)			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	0->1	하측블럭	A13,E21,E22	
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0.5	가상전블럭		1
전블럭	A11,E21	0.5	전블럭	-	
후블럭	A13		후블럭	-	

추가로 탑재되는 순서 및 과정, 블럭 VALUE의 변화 등은 〈표 5〉, 〈표 6〉, 그리고 〈표 7〉에서 보는 바



와 같으며 이를 그림으로 설명한 것이 <그림 11>에서 부터 <그림 14>에 나타나 있다. 이들을 체계적으로 설명하면 다음과 같다. A12블럭이 탑재되므로

- {하측 블럭 VALUE = 1} .AND.
- {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.
- {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 E21블럭과
- {하측 블럭 VALUE = 1} .AND.
- {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.
- {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 E22블럭이 탑재 가능한 블럭이 된다. 두 가지 가능한 대안 중 먼저 입력 순서가 빠른 E21블럭을 선택하여 탑재하고 복수 대안 LIST {E21\*, E22}에 탑재 상태를 표시한다.



<그림 11> E21블럭 탑재 상태 표시

- <표 5>에서 보는 바와 같이
- {하측 블럭 VALUE = 1} .AND.
- {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.
- {가상 전블럭 VALUE + 전블럭 VALUE +

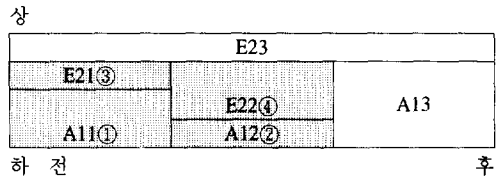
<표 5> E21블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0.5	후블럭	A13	

A13블럭(3)			E21블럭(4): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A12,E22	0.5	전블럭	-	
후블럭	-		후블럭	E22	

E22블럭(5)			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	0->0.3
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0.5->0	가상전블럭		1
전블럭	A11,E21	0.5->1	전블럭	-	
후블럭	A13		후블럭	-	

후블럭 VALUE = 1}인 E22블럭이 탑재 가능한 블럭이 되어 탑재되고 각 선후 블럭의 블럭 VALUE를 수정한다.



<그림 12> E22블럭 탑재 상태 표시

- <표 6>에서 보는 바와 같이
- {하측 블럭 VALUE = 1} .AND.

<표 6> E22블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-		하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0.5->1	후블럭	A13	

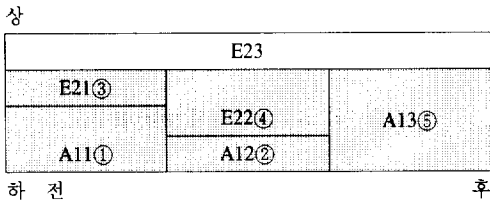
A13블럭(3) E21블럭(4): 탑재

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-		하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-	1	좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭			가상전블럭		1
전블럭	A12,E22	0	전블럭	-	
후블럭	-	0.5->1	후블럭	E22	0->1

E22블럭(5): 탑재 E23블럭(6)

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	0.3->0.7
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A11,E21	1	전블럭	-	
후블럭	A13		후블럭	-	

{가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.  
 {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 A13블럭이 탑재 가능한 블럭이 되어 탑재되고 각 선후 블럭의 블럭 VALUE를 수정한다.



<그림 13> A13블럭 탑재 상태 표시

<표 7>에서 보는 바와 같이  
 {하측 블럭 VALUE = 1} .AND.  
 {가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.  
 {가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 E23블럭이 탑재 가능한 블럭이 되어 탑재되고 각 선후 블럭의 블럭 VALUE를 수정한다.

<표 7> A13블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재 A12블럭(2): 탑재

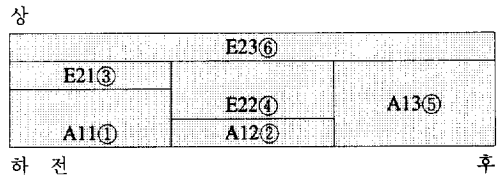
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	1	후블럭	A13	0->1

A13블럭(3): 탑재 E21블럭(4): 탑재

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A12,E22	1	전블럭	-	
후블럭	-		후블럭	E22	1

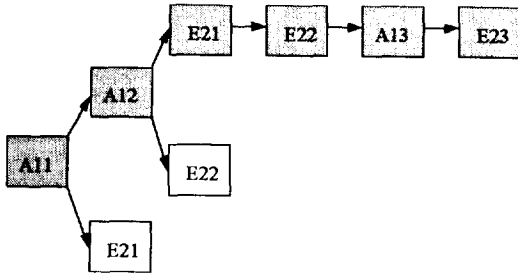
E22블럭(5): 탑재 E23블럭(6)

블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	0.7->1
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A11,E21	1	전블럭	-	
후블럭	A13	0->1	후블럭	-	



<그림 14> E23블럭 탑재 상태 표시

위의 내용에 근거하여 생성된 탑재 순서를 TREE 구조로 나타내면 다음과 같다.



〈그림 15〉 탐색 순서 TREE 구조

(2) Backtracking 실시; 복수 대안 결정

첫번째 탐색 순서가 결정되면 모든 가능한 대안을 찾기 위해서 복수 대안이 발견되는 곳까지 역순으로 자신의 탐색 상태를 제거하고 블록의 탐색시와 반대 개념에 입각해서 블록 VALUE를 수정하고 첫번째 복수 대안이 나오는 곳을 찾아가는 Backtracking의 과정이 필요하다. 먼저 기준 블록의 탐색 상태를 제거하고 다음 방법에 따라 수행한다.

i) 인접 전, 후, 좌, 우블럭 VALUE 수정

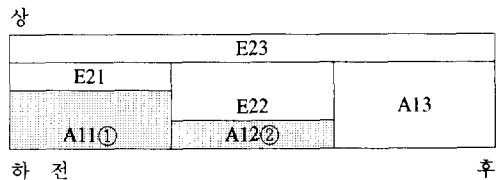
기준 블록의 탐색 상태가 제거되면 이제 기준 블록이 더이상 탐색되지 않았다는 정보를 인접 블록에 인식시켜야 한다. 따라서 블록 탐색시와는 정반대 개념으로 탐색 상태가 제거된 블록에 해당하는 인접 블록의 블록 VALUE를  $\frac{1}{\text{인접 블록 개수}}$ 만큼 감한다.

- ① 인접 전블럭은 후블럭의 탐색 상태가 제거되는 경우이므로 후블럭 VALUE에서 감한다.
- ② 인접 후블럭은 전블럭의 탐색 상태가 제거되는 경우이므로 전블럭 VALUE에서 감한다.
- ③ 인접 좌블럭은 우블럭의 탐색 상태가 제거되는 경우이므로 우블럭 VALUE에서 감한다.
- ④ 인접 우블럭은 좌블럭의 탐색 상태가 제거되는 경우이므로 좌블럭 VALUE에서 감한다.
- ⑤ 인접 상블럭은 하블럭이 탐색 상태가 제

거되는 경우이므로 하블럭 VALUE에서 감한다.

- ii) 1층 블럭의 탐색 상태를 제거할 때는 특별한 규칙을 적용해야 한다. 1층 블럭은 하블럭이 존재하지 않기 때문에 1층 블럭을 탐색할 때 적용했던 규칙과는 역으로 인접 전, 후, 좌, 우블럭 어느 한 블럭이 탐색되어 있지 않을 경우 인접 블럭의 하블럭 VALUE를 0으로 수정하는 방법으로 해결한다.
- iii) 위의 세 가지 블럭 VALUE 수정 방법을 수행하고 나면 첫번째 복수 대안이 나오는 곳에서 탐색 가능한 블럭이라 할지라도 탐색 가능한 조건이 만족되지 않는 경우가 발생할 수 있다. 이런 경우는 블럭의 크기가 동일하지 않기 때문에 가상 VALUE 수정시 생기는 문제이므로 그 차이를 보상해야 한다. 이때 가상 VALUE를 부여해서 탐색 가능한 조건을 만족시키도록 하면 이 문제가 해결된다.

- ① 가상 전블럭 VALUE=1-(전블럭 VALUE+후블럭 VALUE)
  - ② 가상 좌블럭 VALUE=1-(좌블럭 VALUE+우블럭 VALUE)
- ①, ②를 적용하면 첫번째 복수 대안이 나오는 곳에서 탐색 가능 조건을 만족한다. 이 상태에서 다시 앞에서 제시한 Algorithm을 적용하여 이전의 탐색 순서와 상이한 탐색 순서를 생성할 수 있게 된다. 즉 이전에 선택되어 탐색 상태가 표시된 블럭을 제외하고 입력 순서가 가장 빠른 블럭을 탐색하면서 새로운 탐색 순서가 결정된다.



〈그림 16〉 탐색 상태 제거후 상태

〈표 8〉에서 보는 바와 같이 Backtracking 실시한 후

〈표 8〉 Backtracking 실시 후 최종 블록 VALUE

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	1->0.5	후블럭	A13	1->0

A13블럭(3)			E21블럭(4)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1
전블럭	A12,E22	1->0.5	전블럭	-	
후블럭	-		후블럭	E22	1->0

E22블럭(5)			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	1->0
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0->0.5	가상전블럭		1
전블럭	A11,E21	1->0.5	전블럭	-	
후블럭	A13	1->0	후블럭	-	

블럭 VALUE를 수정한 결과

{하측 블럭 VALUE = 1} .AND.

{가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.

{가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 E21블럭과

{하측 블럭 VALUE = 1} .AND.

{가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.

{가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE= 1인 E22블럭이 탑재 가능 블럭이 된다. 두 가지 가능한 대안 중 탑재 상태 표시가 있는 E21블럭을 제외한 E22블럭이 선택되어 탑재된다.

상

E23		
E21	E22③	A13
A11①	A12②	

하 전

후

〈그림 17〉 E22블럭 탑재 상태 표시

〈표 9〉 E22블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0.5->1	후블럭	A13	0

A13블럭(3)			E21블럭(4): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1->0
전블럭	A12,E22	0.5->1	전블럭	-	
후블럭	-		후블럭	E22	0->1

E22블럭(5)			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	
가상좌블럭		1	가상좌블럭	-	0->0.3
좌블럭	-		좌블럭	-	1
우블럭	-		우블럭	-	
가상전블럭		0.5	가상전블럭		1
전블럭	A11,E21	0.5	전블럭	-	
후블럭	A13	0	후블럭	-	

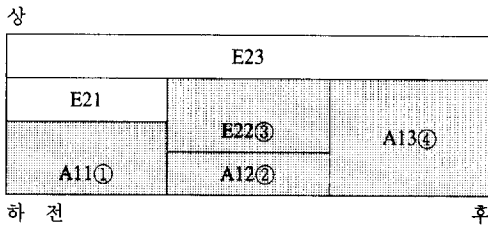
〈표 9〉에서 보는 바와 같이

{하측 블럭 VALUE = 1} .AND.

{가상 좌블럭 VALUE + 좌블럭 VALUE + 우블럭 VALUE = 1} .AND.

{가상 전블럭 VALUE + 전블럭 VALUE + 후블럭 VALUE = 1}인 A13블럭과

{하측 블럭 VALUE = 1} .AND.  
 {가상 좌블럭 VALUE + 좌블럭 VALUE +  
 우블럭 VALUE = 1} .AND.  
 {가상 전블럭 VALUE + 전블럭 VALUE +  
 후블럭 VALUE = 1}인 E21블럭이 탑재 가능한 블  
 록이 된다. 두 가지 가능한 복수 대안 중 입력 순서  
 가 빠른 A13블럭을 선택하여 탑재하고 복수 대안  
 LIST {A13\*, E21}에 탑재 상태를 표시한다.



<그림 18> A13블럭 탑재 상태 표시

<표 10>에서 보는 바와 같이  
 {하측 블럭 VALUE = 1} .AND.  
 {가상 좌블럭 VALUE + 좌블럭 VALUE +  
 우블럭 VALUE = 1} .AND.

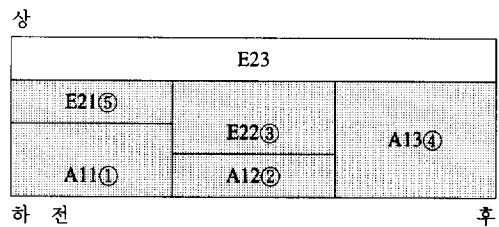
<표 10> A13블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0.5->1	후블럭	A13	0->1

A13블럭(3): 탑재			E21블럭(4)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		1->0
전블럭	A12,E22	1	전블럭	-	
후블럭	-		후블럭	E22	1

E22블럭(5): 탑재			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	0.3->0.7
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0.5	가상전블럭		1
전블럭	A11,E21	0.5	전블럭	-	
후블럭	A13	0->1	후블럭	-	

{가상 전블럭 VALUE + 전블럭 VALUE +  
 후블럭 VALUE = 1}인 E21블럭이 탑재 가능한 블  
 록이 되어 탑재된다.



<그림 19> A13블럭 탑재 상태 표시

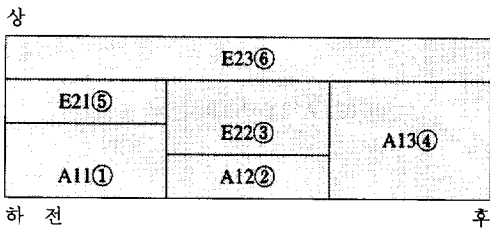
<표 11> E21블럭 탑재시 블럭 VALUE 수정

A11블럭(1): 탑재			A12블럭(2): 탑재		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E21		상측블럭	E22	
하측블럭	-	1	하측블럭	-	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		1	가상전블럭		0
전블럭	-		전블럭	A11	1
후블럭	A12,E22	0.5->1	후블럭	A13	

A13블럭(3): 탑재			E21블럭(4)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	E23	
하측블럭	-	1	하측블럭	A11	1
가상좌블럭		1	가상좌블럭		1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0	가상전블럭		0
전블럭	A12,E22	1	전블럭	-	
후블럭	-		후블럭	E22	1

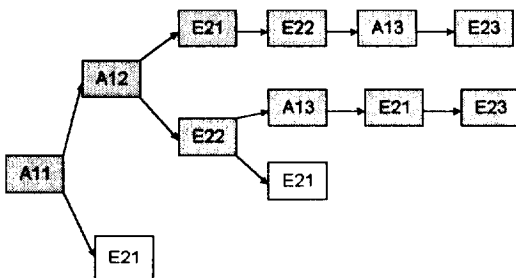
E22블럭(5): 탑재			E23블럭(6)		
블럭구분	선후블럭	VALUE	블럭구분	선후블럭	VALUE
상측블럭	E23		상측블럭	-	
하측블럭	A12	1	하측블럭	A13,E21,E22	0.7->1
가상좌블럭		1	가상좌블럭	-	1
좌블럭	-		좌블럭	-	
우블럭	-		우블럭	-	
가상전블럭		0.5	가상전블럭		1
전블럭	A11,E21	0.5->1	전블럭	-	
후블럭	A13	0->1	후블럭	-	

<표 11>에서 보는 바와 같이  
 (하측 블럭 VALUE = 1) .AND.  
 (가상 좌블럭 VALUE + 좌블럭 VALUE +  
 우블럭 VALUE = 1) .AND.  
 (가상 전블럭 VALUE + 전블럭 VALUE +  
 후블럭 VALUE = 1)인 E23블럭이 탑재 가능 블럭  
 이 되어 탑재된다.



<그림 20> E23블럭 탑재 상태 표시

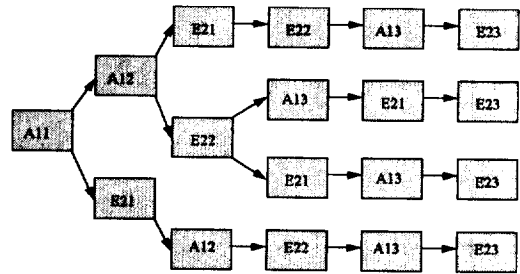
위의 같은 방법으로 Backtracking을 실시한 후, 블럭 탑재를 반복하여 두번째 탑재 순서가 생성된 것을 간단한 TREE 구조로 나타내면 <그림 21>과 같다.



<그림 21> 탑재 순서 TREE 구조

위의 같은 방법으로 Backtracking을 반복하여 모든

가능한 탑재 순서를 생성한다. 앞에서 제시한 Algorithm을 적용한 후 얻어진 모든 가능한 네 개의 탑재 순서 대안에 해당하는 TREE구조는 다음 <그림 22>와 같다.



<그림 22> 모든 가능 탑재 순서 TREE 구조

위의 예제에 대한 프로그램 실행 결과가 <표 12>에 나타나 있다.

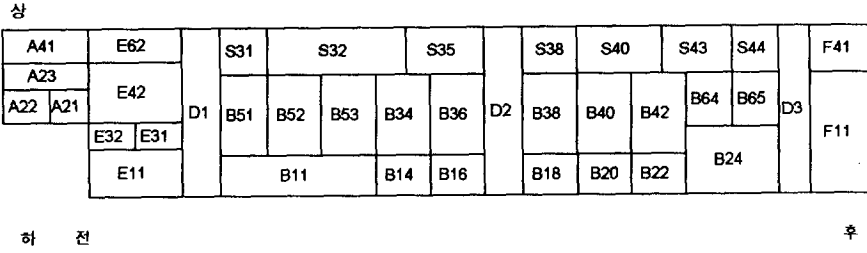
이때 [1]은 탑재 순서 대안 번호, [0:A11]에서 0은 입력 순서, A11은 블럭명울 의미한다.

<표 12> Backtracking실시 결과 출력

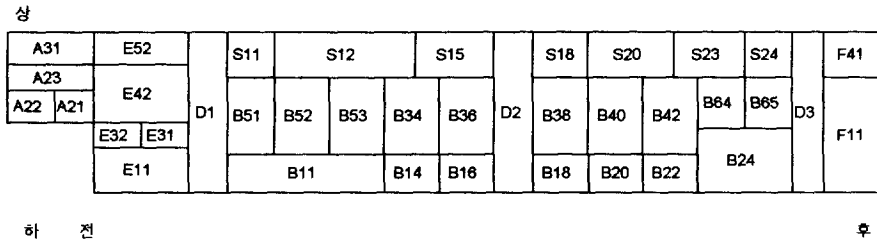
[1] :	[ 0:A11]-[ 1:A12]-[ 3:E21]- [ 4:E22]-[ 2:A13]-[ 5:E23]
[2] :	[ 0:A11]-[ 1:A12]-[ 4:E22]- [ 2:A13]-[ 3:E21]-[ 5:E23]
[3] :	[ 0:A11]-[ 1:A12]-[ 4:E22]- [ 3:E21]-[ 2:A13]-[ 5:E23]
[4] :	[ 0:A11]-[ 3:E21]-[ 1:A12]- [ 4:E22]-[ 2:A13]-[ 5:E23]

### 7. 구획과 구획의 구분과 적용례

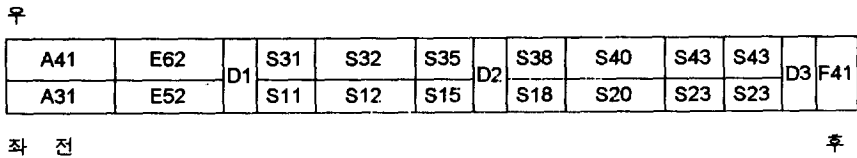
블럭 탑재시 구획과 구획의 구분에 따른 공정계획에 다소 어려움이 있는 것으로 알려져 있다. 앞에서 제시한 Algorithm에서는 블럭 입력시 간단한 절차에 의하여 구획과 구획의 구분은 물론 건조 방식 관점에서 Semi-tandem방식[3], 격벽 등을 고려할 수 있다. 다시 말해서, 구획과 구획 사이에 Dummy Block개념을 도입하여 구획간의 구분을 피할 수 있다.



(a) 좌 측면도



(b) 우 측면도



(c) 평면도

〈그림 23〉 47개 블록을 가정한 표준 도면

이러한 Dummy Block의 도입으로 동일 구획내 블록의 탑재를 끝내고 나서 인접 구획의 블록의 탑재가 진행되어 구획별 탑재를 가능하게 한다.

〈그림 23〉은 47개의 블록으로 구성된 선박의 측면도와 평면도로서 이를 토대로 앞에서 제시한 Algo-

rithm을 적용하여 탑재 순서를 생성해 보기로 하자.

(1) 입력 자료

〈그림 23〉에서 제시한 표준선 도면의 입력 자료는 〈표 13〉과 같다.

〈표 13〉 입력 자료

```

47 <- 전체 블럭수, 아래는 블럭명...
E11 E31 E32 E42 E52 E62
B11 B14 B16 B18 B20 B22 B24
B34 B36 B38 B40 B42 B51 B52 B53 B64 B65
S11 S12 S15 S18 S20 S23 S24 S31 S32 S35
S38 S40 S43 S44
F11 F41
A21 A22 A23 A31 A41
D1 D2 D3
블럭간 접속관계 표현
블럭명 상위접속블럭수 블럭명.. (같은 방법으로) 하
위... 좌측... 우측... 전위... 후위...
E11 2 E31 E32 0 0 0 0 1 D1
E31 1 E42 1 E11 0 0 1 E32 1 D1
E32 1 E42 1 E11 0 0 0 1 E31
E42 2 E52 E62 2 E31 E32 0 0 2 A21 A23 1 D1
E52 0 1 E42 0 1 E62 1 A31 1 D1
E62 0 1 E42 1 E52 0 1 A41 1 D1
B11 3 B51 B52 B53 0 0 0 1 D1 1 B14
B14 1 B34 0 0 0 1 B11 1 B16 B16 1 B36 0 0 0 1
B14 1 D2
B18 1 B38 0 0 0 1 D2 1 B20
B20 1 B40 0 0 0 1 B18 1 B22
B22 1 B42 0 0 0 1 B20 1 B2
B24 2 B64 B65 0 0 0 2 B22 B42 1 D3
B34 4 S12 S15 S32 S35 1 B14 0 0 1 B53 1 B36
B36 2 S15 S35 1 B16 0 0 1 B34 1 D2
B38 2 S18 S38 1 B18 0 0 1 D2 1 B40
B40 2 S20 S40 1 B20 0 0 1 B38 1 B42
B42 4 S20 S23 S40 S43 1 B22 0 0 1 B40 2 B24 B64
B51 2 S11 S31 1 B11 0 0 1 D1 1 B52
B52 2 S12 S32 1 B11 0 0 1 B51 1 B53
B53 2 S12 S32 1 B11 0 0 1 B52 1 B34
B64 2 S23 S43 1 B24 0 0 1 B42 1 B65
B65 2 S24 S44 1 B24 0 0 1 B64 1 D3
S11 0 1 B51 0 1 S31 1 D1 1 S12
S12 0 3 B34 B52 B53 0 1 S32 1 S11 1 S15
S15 0 2 B34 B36 0 1 S35 1 S12 1 D2
S18 0 1 B38 0 1 S38 1 D2 1 S20
S20 0 2 B40 B42 0 1 S40 1 S18 1 S23
S23 0 2 B42 B64 0 1 S43 1 S20 1 S24
S24 0 1 B65 0 1 S44 1 S23 1 D3
S31 0 1 B51 1 S11 0 1 D1 1 S32
S32 0 3 B34 B52 B53 1 S12 0 1 S31 1 S35
S35 0 2 B34 B36 1 S15 0 1 S32 1 D2
S38 0 1 B38 1 S18 0 1 D2 1 S40
S40 0 2 B40 B42 1 S20 0 1 S38 1 S43
S43 0 2 B42 B64 1 S23 0 1 S40 1 S44
S44 0 1 B65 1 S24 0 1 S43 1 D3
F11 1 F41 0 0 0 1 D3 0
F41 0 1 F11 0 0 1 D3 0
A21 1 A23 0 0 0 1 A22 1 E42
A22 1 A23 0 0 0 0 1 A21
A23 2 A31 A41 2 A21 A22 0 0 0 1 E42
A31 0 1 A23 0 1 A41 0 1 E52
A41 0 1 A23 1 A31 0 0 1 E62
D1 0 0 0 0 5 E11 E31 E42 E52 E62 4 B11 B51 S11 S31
D2 0 0 0 0 4 B16 B36 S15 S35 4 B18 B38 S18 S38
D3 0 0 0 0 4 B24 B65 S24 S44 2 F11 F41
    
```

(2) 출력 자료

〈표 13〉의 입력된 자료를 가지고 구현된 프로그램  
을 이용하여 실행한 결과 가운데 두 가지 탑재 순서  
대안은 다음과 같다.

〈표 14〉 출력 자료

```

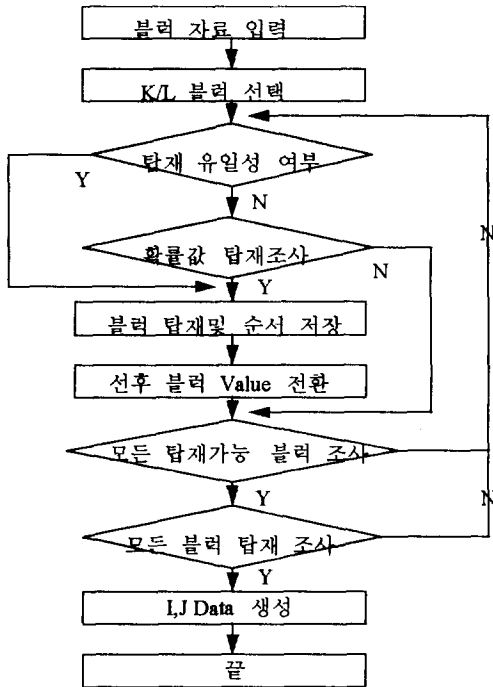
[1]: [0:E11]-[1:E31]-[2:E32]-[3:E42]-
[4:E52]-[5:E62]-[39:A21]-[40:A22]-
[41:A23]-[42:A31]-[43:A41]-[44:D1]-
[6:B11]-[7:B14]-[8:B16]-[18:B51]-
[19:B52]-[20:B53]-[13:B34]-[14:B36]-
[23:S11]-[24:S12]-[25:S15]-[30:S31]-
[31:S32]-[32:S35]-[45:D2]-[9:B18]-
[10:B20]-[11:B22]-[15:B38]-[16:B40]-
[17:B42]-[12:B24]-[21:B64]-[22:B65]-
[26:S18]-[27:S20]-[28:S23]-[29:S24]-
[33:S38]-[34:S40]-[35:S43]-[36:S44]-
[46:D3]-[37:F11]-[38:F41]

[100001]: [0:E11]-[1:E31]-[2:E32]-[3:E42]-
[4:E52]-[5:E62]-[39:A21]-[40:A22]-
[41:A23]-[42:A31]-[43:A41]-[44:D1]-
[6:B11]-[7:B14]-[8:B16]-[18:B51]-
[19:B52]-[20:B53]-[13:B34]-[23:S11]-
[24:S12]-[30:S31]-[31:S32]-[14:B36]-
[25:S15]-[32:S35]-[45:D2]-[9:B18]-
[10:B20]-[15:B38]-[11:B22]-[33:S38]-
[16:B40]-[17:B42]-[12:B24]-[21:B64]-
[26:S18]-[27:S20]-[22:B65]-[34:S40]-
[28:S23]-[29:S24]-[35:S43]-[36:S44]-
[46:D3]-[37:F11]-[38:F41]
    
```

### Ⅲ. 네트워크 생성 Algorithm

앞에서 제시된 블럭간 선후 관계로부터 탑재 순서  
생성하는 방법을 이용하여 초기 탑재 네트워크를 생  
성한다. 탑재 네트워크의 생성은 기준 블럭에 대해 다  
음으로 탑재 가능한 블럭들을 node로 표시하고 node  
들 사이에 arc(선후 관계)를 연결하여 확장시키는 과  
정을 반복하였다. 이때 각 단계별로 탑재 가능한 블  
럭이 다수인 경우 node로 선택 여부는 확률로 표시하  
였다. 초기 탑재 네트워크 Algorithm은 〈그림 24〉와  
같다.





〈그림 24〉 초기 탑재 네트워크 생성 Algorithm

Algorithm을 단계별로 상세하게 설명하면 아래와 같다.

단계 0 (K/L블럭 선택 및 확률값 입력)

K/L블럭을 선택하고, 탑재 가능한 블럭 선택시 탑재될 확률값을 초기값으로 부여한다.

단계 1 (탑재 가능 블럭의 유일 여부 조사)

탑재 가능한 블럭이 하나인 경우는 탑재하고 다수인 경우는 먼저 제시된 Algorithm에 의하면 입력 순서에 따라 블럭을 선택하여 탑재 순서를 결정했지만, 여기서는 초기값으로 부여한 확률값에 따라 탑재 여부를 판단하여 결정한다.

단계 2 (탑재 순서 저장)

블럭을 탑재하고 인접 블럭에 탑재상태 정보를 나타내는 선후 블럭의 블럭 VALUE를 전환한 후, 탑재 순서를 저장한다. 그리고 나서 단계 1로 간다.

단계 3 (I, J Data 생성)

모든 블럭의 탑재 순서가 결정되면 이것이 곧 단계별 탑재 순서가 된다. 단계별 탑재 순서를 가지고 탑재 네트워크 생성을 위한 선후 관계를 나타내는 I, J

Data (I는 선블럭, J는 후블럭)를 생성한다. 우선 기준 블럭을 선택한 후, 바로 전 단계에 탑재된 블럭과 상, 전, 후, 좌, 우 인접 관계가 있는 블럭에 I, J 관계를 표시한다. 전 단계와 인접 관계가 없다면 그 이전 단계로 가서 I, J 관계를 표시한다. 탑재 순서에 의해 모든 블럭의 I, J 관계를 표시한 결과 블럭의 크기가 동일하지 않기 때문에 I, J 관계가 나타나야 하는 곳에 I, J 관계가 나타나지 않는 경우가 발생한다. 그래서 다음과 같은 별도의 규칙을 부여할 필요가 있다. 블럭을 하나 하나 선택하여 그 블럭의 상측 블럭과 I, J 관계 표시가 없고 상측 블럭에서 볼 때 하측 블럭의 모든 블럭 중 어떠한 블럭과도 I, J 관계가 없는 경우 기준 블럭과 그 블럭의 상측 블럭 사이에 I, J 관계를 표시한다. 이상의 과정을 거치지 않는 블럭인 경우에는 진수(L/C)에 I, J 관계를 표시하는 방법으로 탑재 네트워크를 생성한다.

다음은 〈그림 23〉에 대한 단계별 탑재 순서이다.

(1) 출력 자료

- 선택할 확률 = 1.0 인 경우

〈표 13〉입력 자료를 가지고 위의 단계별 탑재 순서 Algorithm를 이용한 블럭간 I, J 관계는 다음과 같다.

〈표 15〉 단계별 탑재 순서

<p>[Solution] E11: 블럭명 0: 단계                  [E11:0]-[E31:1]-[E32:1]-[D1:1]-[E42:2]-                  [B11:2]-[E52:3]-[E62:3]-[B14:3] -[B51:3]-                  [A21:3]-[B16:4]-[B52:4]-[S11:4]-[S31:4]-                  [A22:4]-[B53:5]-[A23:5]-[B34:6]-[A31:6]-                  [A41:6]-[B36:7]-[S12:7]-[S32:7]-[S15:8]-                  [S35:8]-[D2:9]-[B18:10]-[B20:11]-[B38:11]-                  [B22:12]-[B40:12]-[S18:12]-[S38:12]-[B42:13]                  [B24:14]-[S20:14]-[S40:14]-[B64:15]-[B65:16]                  -[S23:16]-[S43:16]-[S24:17]-[S44:17]-[D3:18]                  -[F11:19]-[F41:20]</p>
--

(2) 초기 탑재 네트워크

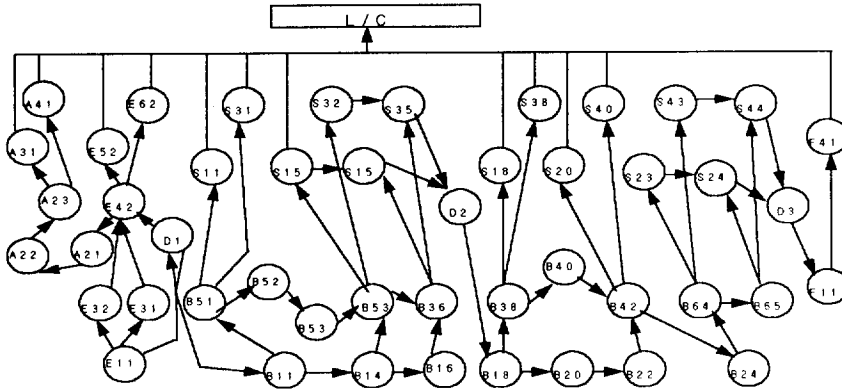
〈표 16〉의 블럭간 I, J 관계를 이용하여 나타난 초기 탑재 네트워크는 〈그림 25〉와 같다.

〈표 16〉블럭간 I, J관계

E11 → E31	A22 → A23	B38 → B40	D3 → F11
E11 → E32	B53 → B34	B38 → S18	F11 → F41
E11 → D1	A23 → A31	B38 → S38	B14 → B34
E31 → E42	A23 → A41	B22 → B42	B16 → B36
E32 → E42	B34 → B36	B40 → B42	E52 → LAUNCH
D1 → E42	B34 → S12	B42 → B24	E62 → LAUNCH
D1 → B11	B34 → S32	B42 → S20	S11 → LAUNCH
E42 → E52	B36 → S15	B42 → S40	S18 → LAUNCH
E42 → E62	S12 → S15	B24 → B64	S20 → LAUNCH
B11 → B14	B36 → S35	B64 → B65	S31 → LAUNCH
B11 → B51	S32 → S35	B64 → S23	S38 → LAUNCH
E42 → A21	S15 → D2	B64 → S43	S40 → LAUNCH
B14 → B16	S35 → D2	B65 → S24	F41 → LAUNCH
B51 → B52	D2 → B18	S23 → S24	A31 → LAUNCH
B51 → S11	B18 → B20	B65 → S44	A41 → LAUNCH
B51 → S31	B18 → B38	S43 → S44	
A21 → A22	B20 → B22	S24 → D3	
B52 → B53	B20 → B40	S44 → D3	

입블럭의 방지가 가능하고 구획 등 특수한 형태의 접속 관계 역시 규정할 수 있으며 블럭의 탑재 안정성 등을 고려하는 데 무리가 없다. 뿐만 아니라 제안된 Algorithm을 사용하여 초기 탑재 네트워크를 생성할 수도 있다. 블럭간 선후 관계를 이용한 탑재 순서 결정 방법의 활용 방안으로는 Dock 사용 기간이나 블럭간 Pitch가 변화하는 경우 초기 탑재 네트워크의 변경을 임의의 중간 노드를 중간 기점으로 하여 복수 대안으로 생성할 수 있다. 향후 선박의 설계 단계에서 선각 공정에 이르기까지 CAD에 의한 블럭 데이터의 인식이 자동화되면 본 연구의 탑재 순서 결정에 별도로 블럭 데이터의 입력이 필요하지 않게 될 것이다.

본 연구의 결과로 얻어진 탑재 순서는 실제 생산 계획에서 사용될 일정 계획의 입력 자료로 사용되므로 추후의 연구 과제로는 이러한 일정 계획과의 연계성



〈그림 25〉 초기 탑재 네트워크

**IV. 결론 및 향후 과제**

본 연구에서 제안된 내용은 블럭간 선후 관계를 이용한 탑재 순서 결정 방안을 수립하고 구현하였다. 또한 이 방안을 이용한 초기 탑재 네트워크 생성 방안을 제시하였고, 표준선에 적용한 실험을 통하여 제시된 방안의 타당성을 검토하였다.

본 연구에서 제시된 방법에 대한 활용 방안으로 블럭간 선후 관계를 이용한 탑재 순서 결정 방법의 경우, 블럭간 접속 관계를 기존 정보로 사용하기 때문에 물리적인 접속 관계를 정확히 표현할 수 있어서 삼

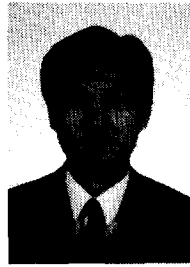
에 관한 고찰이 필요하다.

또한 본 연구에서 제안된 방안을 실제로 적용하게 될 중·대형 선박으로의 적용 가능성을 검증하는 작업이 요구된다.

본 연구를 수행함에 있어 충분한 자료 제공과 협조를 아끼지 아니하신 대우, 삼성, 현대중공업 관계자 여러분과 한국기계연구원 연구원 여러분께 감사드립니다.

**【참고문헌】**

- [1] 과기처 연구 보고서, “선박 설계 생산 전산 시스템(VI) - 초기 일정 · 공정 계획 일관 시스템 개발,” 1995.
- [2] 박명환, 이운식, 옥영석, 이태억, “국내 조선 산업의 개요와 국내외 산업공학관련 연구,” 산업공학, 제8권, 제2호, pp. 5-20, 1995. 7
- [3] 김극천외의 4명, 조선공학개론, 문운당, pp. 195-214, 1974.
- [4] 이재규 외의 9명, “대우조선의 일정관리 전문가시스템 개발: DAS 프로젝트,” 지능 정보시스템지, 제1권, 제1호, pp. 9-32, 1992.
- [5] 김훈주, 이재원, “유전 알고리즘을 이용한 탑재 공정과 일정 계획,” '94 추계 조선 학회.
- [6] Renzo Di. Luca, “An Integrated Procedure for Hull Design and Production,” Journal of Ship Production, Vol. 3, No. 3, pp. 175-185, 1987.
- [7] Richard J. Neumann and David J. McQuade, “Application of PC-Based Project Management in an Integrated Planning Process,” Journal of Ship Production, Vol. 8, No. 4, pp. 191-204, 1992.
- [8] Lee, J. K. and Choi, H. R., “Erection Scheduling at Shipbuilding Using Constraint Directed Graph Search: DAS-ERECT,” '93 Korea Japan Joint conference on Expert Systems, Seoul, 1993.
- [9] Lee, Jae Kyu, Hyung Rim Choi, Ok Ryul Yang and Ho Dong Kim, “Scheduling Shipbuilding using Constraint Directed Graph Search: DAS-ERECT,” Intelligent Systems in Accounting, Finance and Management, Vol. 3, No. 2, pp. 111-125, 1994.



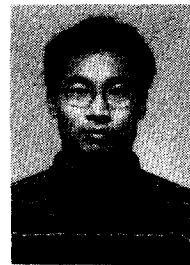
홍윤기

1974~1980 고려대학교 산업공학과 학사  
 1983~1985 남가주대학교(USC) 산업공학과 석사  
 1985~1989 남가주대학교(USC) 산업공학과 박사  
 1989~1991 캘리포니아 주립대(노스리지) 경영과학과 조교수  
 1991~현재 한성대학교 산업시스템공학부 부교수  
 관심분야 Systems Simulation, War-game Analysis



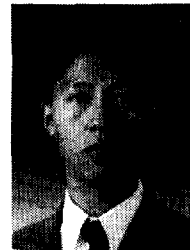
정은경

1991년 고려대학교 산업공학과 학사  
 1993년 고려대학교 산업공학과 석사  
 1993~ 현재 고려대학교 산업공학과 박사과정중  
 관심분야 시뮬레이션, 네트워크 최적화, 정보통신



전진

1992년 고려대학교 산업공학과 학사  
 1994년 고려대학교 산업공학과 석사  
 1994~ 현재 고려대학교 산업공학과 박사과정중  
 관심분야 CIM, 시뮬레이션



김세영

1996년 한성대학교 산업공학과 학사  
 1996~ 고려대학교 산업공학과 석사과정중  
 관심분야 인공지능, 품질관리, 시뮬레이션