

계약 만족 기법을 이용한 조선 산업에서의 탑재 일정 생성에 관한 연구

A Study on the Erection Scheduling in the Shipbuilding Using Constraint Satisfaction Technique

김기동* 장용성**
Kim, Ki-Dong Jang, Yong-Sung

Abstract

The dock is the most important resource in shipbuilding yard. Among the shipbuilding schedules, the ship erection schedule in a dock is preferentially built. As results of it, the other schedules(machining in plants, block assembly, pre-painting, pre-rigging, painting and etc) are made.

In this study, ship erection scheduling is formulated using ILOG Scheduler. This study is to develop a new problem solving method for ship erection to make an effective schedule based on Constraint Satisfaction Technique(CST).

키워드 : 탑재, 도크, 제약만족기법

Keywords : *Erection, Dock, Constraint Satisfaction Technique*

1. 서론

조선산업에서의 이익 극대화를 위해서는, 공기 단축을 통한 매출 증대, 주어진 생산 자원(인력, 설비 등)의 효율적인 이용을 통한 생산비 절감이 필수적인 사안이다. 다시 말하면, 효율적인 자원의 이용으로 선박의 건조 공기를 줄임으로써 조선 산업에서의 이익 극대화를 도모할 수 있고 이는 효과적인 생산 계획과 관리를 통해 얻을 수 있다.

생산 계획과 관리는 선행, 선내 및 가공 shop등 선박 건조의 전 과정에서 이루어지고 있는데, 생산 계획의 가장 중요한 의사결정 사항 중 하나는 중요한 작업들에 대한 일정계획의 수립이다.

조선 산업에서의 일정계획은 주어진 계획 기간이 길고, 일정 수립의 대상이 되는 작업들의 종류가 다양한 관계로 매우 어려운 문제로 인식되고 있다. 게다가, 작업이 이루어지는 장소에 따라서도 주어진 일

정계획 문제는 많은 차이를 보이고 있다. 탑재 일정의 경우에는 프로젝트 일정계획의 성격이 강하고, 가공 공장의 경우에는 job shop 일정계획, 조립 공장의 경우에는 flow shop 일정계획과 유사한 면을 보이고 있다. 그렇지만, 조선 산업의 특성에서 오는 여러 가지 제약들(공간 제약, 인력 제약, 이종 제품 생산 등)은 조선 일정계획을, 앞에서 언급한 여타의 일정계획과 뚜렷이 구별짓게 하고, 보다 풀기 어려운 문제로 만들고 있다.

조선 산업에서의 일정계획 수립과정을 살펴보면, 가장 중요한 자원인 도크이용에 관한 일정을 우선 수립한다. 즉 도크에서의 선각 조립과 관련된 탑재 일정계획을 우선 수립하고, 그 후 이의 결과를 바탕으로 가공 및 조립공장에 대한 일정, 선행 도장 및 의장과 관련된 일정들이 수립된다.

본 연구에서는 도크에서의 탑재 일정 문제를 제약만족기법(CST : Constraint Satisfaction Technique)과 객체 지향 기법을 지원하는 방법론을 이용해서 실제 현장의 업무규칙을 고려하여 풀이하고, 모형 수립의 편의성, 해에 대한 당위성과 조선 탑재 일정 문제에 대한 적합성을 살펴본다.

* 강원대학교 산업공학과 전임강사, 공학박사

** 강원대학교 대학원 산업공학과 석사과정

제 2장에서는 관련 연구 현황을, 3장에서는 객체지향 모형화와 CST를 지원하는 도구인 ILOG Scheduler를 이용한 일정계획 모형화에 대한 개요를, 4장에서는 실제 조선 탑재 일정 문제의 업무규칙을 반영한 탑재 일정계획 모형 수립과 실제 자료를 이용한 예제 풀이 결과를, 5장에서는 결론 및 추후 연구 방향을 제시한다.

2. 관련 연구 현황

2.1. 조선 탑재 일정 관련 연구 현황

일정계획 문제는 일반적으로 NP hard인 문제로 분류되어 최적해를 얻기는 매우 어려운 일인 것으로 인식되었다[10]. 따라서 80년대 이전에는 주로 발견적 기법에 기반한 방법론들이 현장에서 이용되었다. 80년대 말 이후에는 새로운 일정계획 해법의 등장, 그리고 IT(Information Technology) 및 컴퓨팅 환경의 급속한 발전, 다양한 시스템 모델링 및 개발 지원 도구의 활용을 통하여, 인공지능 및 최적화와 관련된 기법들이 현장에서도 이용될 수 있는 길이 열렸다.

조선 일정계획과 관련되어 보고된 연구 중 가장 많은 부분을 차지하는 분야는 탑재 일정계획의 수립에 관련된 내용이다. 일반적으로 조선 shop에서 가장 중요한 자원은 도크라고 볼 수 있다. 따라서 도크의 효율적인 이용을 위해 도크 외에서는 이루어질 수 없는 작업들에 대한 일정계획 수립은 조선 shop 전체의 생산성과 밀접한 관련이 있다.

김훈주[1]는 탑재 순서(sequence) 생성과 탑재 일정 생성 문제를 다루었다. 선정된 수행도는 탑재 부하 평준화이며, 고려한 제약은 블록간 위상(topology), 탑재 공법에 따른 탑재 순서, 탑재 작업의 기술적 제약 사항, 중간진수 범위, 크레인 일일 탑재 블록 수(탑재 횟수), 탑재 블록간 용접장 부하 등이다. 이용한 방법론은 유전 알고리즘이다.

홍윤기[6]는 탑재 순서 생성과 탑재 초기 네트워크 구성 문제를 다루었다. 실행 가능한 모든 탑재 순서 생성을 목표로 했다. 고려된 제약은 블록간 위상, 탑재 공법에 따른 탑재 순서 등이다. 이용한 방법론은 backtracking을 이용한 탐색 방법이다. 탑재 일정 생성 문제는 다루지 않았다.

Hyung Rim Choi[7]는 PE(Pre Erection)/도크 장의 탑재 중일정계획(日 단위의 일정 수립) 문제를 다루었다. 선정된 수행도는 전문가의 제약을 만족하는 실행 가능한 탑재 일정이다. 고려된 제약은 Goliath Crane(이후 G/C라 칭함) 부하, 탑재 기술적 전문가

지식 등이다. 제약 조건하의 그래프 탐색기법(constraint directed graph search)을 이용한 전문가 시스템을 이용해서 풀이했다.

그 외 보고된 자료에 의하면 탑재 관련 연구에서는 탑재 순서 생성 문제, 탑재 일정 생성 문제, 도크 내 배치 문제, PE장 배치 및 PE장의 작업 일정, 가공 및 조립공장의 일정 등의 문제와 복합적으로 고려되고 있음을 알 수 있다. 이용되는 방법론에서는 발견적 기법 및 탐색 기법이 주로 이용되며, 효율적인 탐색을 위해 인공 지능 기법들이 이용되고 있음을 알 수 있다[1,2,3,4,5,6,7].

2.2. 제약 만족 기법

제약만족문제(CSP; Constraint Satisfaction Problem)는 주어진 제약조건을 만족하는 가능해를 찾는 것을 목적으로 하는 문제이며, 제약만족기법은 CSP의 해를 구하는 데 사용되는 방법의 총칭이다. 제약만족문제는 변수(variable), 변수에 할당 가능한 값의 집합(domain), 그리고 변수간의 관계(제약, constraint)로 정의된다. 일정계획 문제는 변수와 제약을 이용하여 제약만족문제로 표현된다. 제약만족기법으로는 제약전파(constraint propagation), 변수정렬(variable ordering), 값정렬(value ordering), 탐색 방법 등을 들 수 있다. 제약전파는 부분해를 이용하여 다른 변수가 가질 수 있는 가능한 값의 집합을 줄여 주는 기법이다.

제약전파의 기본 알고리즘인 호일관화기법(arc consistency enforcement algorithm)은 작업 순서를 정의하거나, 작업의 시작시점(또는 종료시점)을 결정하였을 때에 해당 작업과 관계를 갖고 있는 다른 작업의 작업가능시간을 조정하는 과정에 사용된다. 이외에 한정된 자원 용량을 이용한 제약전파 방법(immediate selection)이 있다. 변수정렬은 일정계획의 대상이 되는 작업 또는 공정을 선택하는 방법이며, 값정렬은 작업 또는 공정의 작업순서, 시작 시점(또는 종료시점)을 결정하는 방법이다. 탐색 방법은 가능해를 발견하지 못한 경우에 이전의 부분해로 돌아오는 backtracking 방법을 의미한다. 즉, 제약만족기법을 이용한 해법은 경영과학에서 널리 사용되어 온 분지한계 방법과 유사하다.

3. ILOG Scheduler를 이용한 일정계획 모형화 개요

일반적으로 수식에 의한 모형 수립에는, 일정계획

문제의 상황을 변수를 이용하여 재해석하고, 분석하는 과정이 필요하다. ILOG Scheduler는 수식을 이용한 모형화 대신 객체를 이용해서 모형화하는 방법을 제공한다. ILOG Scheduler는 C++을 이용하여 개발되었으며, 이용법 또한 C++ 이용법과 동일하다.

ILOG Scheduler에서는 일정수립의 대상이 되는 작업(탑재일정계획에서는 블록의 탑재)을 activity라는 객체로 표현하고, 작업을 위해 이용되는 자원(탑재일정계획에서는 G/C, PE장의 정반, 심출/취부/용접인력 등)을 resource라는 객체로 표현한다. ILOG Scheduler를 이용해서 탑재일정계획을 모형화하기 위해서는 우선 activity와 resource를 정의하고, 이들 간의 관계를 activity와 resource의 멤버 함수를 이용해서 서술식으로 나열한다. ILOG Scheduler를 이용한 모형화 방법을 간략히 설명한다.

3.1. activity의 생성과 activity 간의 선후행 제약

ILOG Scheduler에서 제공하는 activity class의 class hierarchy는 다음 그림 1과 같다.

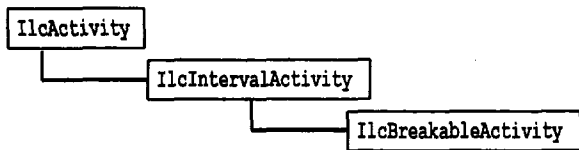


그림 1. IlcActivity의 class hierarchy

이 중 IlcIntervalActivity라는 class는 작업이 중간에 멈추는 경우(다른 작업으로 인해)가 없는 activity를 표현한다. 이 class의 생성자 중 하나는 다음과 같다.

```
IlcIntervalActivity::IlcIntervalActivity(IlcSchedule schedule,
    IlcInt duration);
```

생성자의 입력 인자 중 IlcSchedule은 ILOG Scheduler를 이용해서 형성된 문제를 총체적으로 관리하며, IlcManager를 만들고 난 후, 만들어야 한다. 그리고 ILOG Scheduler에서 제공하는 class는 IlcSchedule을 만들고 난 후 사용할 수 있다. 입력 인자 duration은 해당 activity의 시작에서 끝나는 시간을 의미한다. 참고로 이 시간은 변수를 이용해서 줄 수도 있다. 이 때 이용되는 생성자는 다음과 같다.

```
IlcIntervalActivity::IlcIntervalActivity(IlcSchedule schedule,
    IlcIntVar duration);
```

이를 이용해서 preJob(소요 공기는 3일)과 postJob(소요 공기 5일)이라는 activity를 생성하면

다음과 같다.

```
IlcManger m(IlcEdit);
IlcSchedule sch(m, 0, MAX_DATE); //일정계획 기간이 0부터
//MAX_DATE인 객체 sch 생성
IlcIntervalActivity preJob(sch, 3);
IlcIntervalActivity postJob(sch, 5);
```

이렇게 생성된 activity간에는 선후행 관계에 대한 제약을 다양하게 줄 수 있는데, 선후행 관계에 대한 제약은 IlcIntervalActivity의 상위 class인 IlcActivity의 멤버 함수를 이용해서 준다. 다음은 멤버 함수 startsAfterStart의 선언부이다. 함수 호출 결과 return되는 object는 IlcConstraint의 하위 class 형인 IlcPrecedenceConstraint이다. 다음의 멤버 함수는 act라는 activity 수행 후 delay 동안 경과 후 해당 activity를 수행할 수 있음을 나타낸다. 이 제약을 반영하려면, IlcManager의 멤버 함수 add()를 이용하여 제약 사항임을 알려줘야 한다.

```
IlcPrecedenceConstraint IlcActivity::startsAfterEnd(IlcActivity
    act, IlcInt delay = 0);
```

이를 이용해서 postJob 이 preJob 수행 후 3일 후부터 실행 가능하다는 제약은 다음과 같이 줄 수 있다.

```
m.add(postJob.startsAfterEnd(preJob, 3));
```

이렇게 activity의 선후행 관계를 줄 수 있는 멤버 함수의 종류는 다음과 같다.

```
StartsAfterStart, StartsAfterEnd, EndsAfterStart,
EndsAfterEnd, StartsAtStart, StartsAtEnd,
EndsAtStart, EndsAtEnd
```

위에서 At의 의미는 delay 경과 후, 정확히 그 때 activity 수행이 시작되어야 함을 의미하고, After의 의미는 delay 경과 이 후부터 activity 수행이 가능함을 의미한다. Activity 수행 시작 시간에 제약을 절대적 시간으로 줄 수 있는 멤버 함수들도 있다. 아래의 함수 startsBefore는 activity가 time 전에 시작되어야 함을 의미한다.

```
IlcTimeBoundConstraint IlcActivity::startsBefore(IlcInt time);
```

이렇게 activity의 시작 시각에 대한 제약을 줄 수 있는 멤버 함수의 종류는 다음과 같다.

```
StartsBefore, StartsAfter, StartsAt,
EndsBefore, EndsAfter, EndsAt
```

이를 이용해서 preJob 이 스케줄링 기준 시점(0)에서 7일 후부터 실행 가능하다는 제약은 다음과 같이 줄 수 있다.

```
m.add(preJob.startsAfter(7));
```

3.2. resource 생성과 activity와 resource 관계 설정

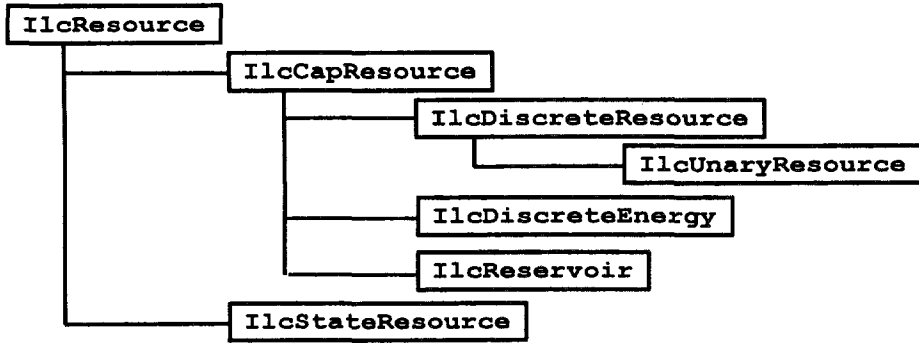


그림 2. IlcResource의 class hierarchy

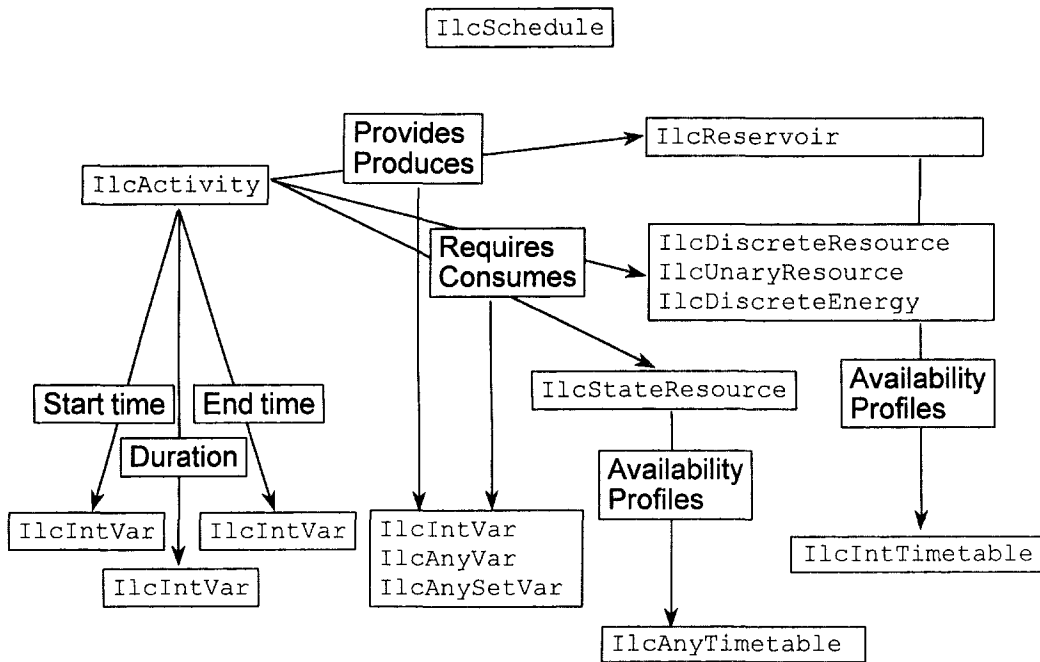


그림 3. activity와 이용되는 resource에 대한 관계도

ILOG Scheduler에서 제공하는 resource class의 class hierarchy는 그림 2와 같다.

이 중 IlcDiscreteResource는 단위 시간 당 자원의 사용 총량이 정해져 있는, 동종의 기계나 동일한 능력을 가진 인력 등을 모형화하기 적합하다. 다음은 생성자의 선언부이다.

```

IlcDiscreteResource::IlcDiscreteResource(IlcSchedule schedule,
    IlcInt capacity, IlcBool timetable = IlcTrue);
    
```

이 생성자의 마지막 입력 인자인 IlcBool은 IlcTrue 혹은 IlcFalse 값을 갖는 자료형인데 IlcTrue이면 이 자원에 대한 timetable을 만들고, 매일 매일의 사용량에 대한 제약을 자동으로 추가한다. 즉 마지막 인자 값을 생략하거나, IlcTrue를 입력하면, 해당 자원에 대해서는 주어진 용량을 초과해서 사용하지 못한다. 이를 이용해서 용량이 4인 man이라는 자

원을 만들면 다음과 같다.

```

IlcDiscreteResource manHour(sch, 4);
    
```

'Activity 수행을 위해 자원이 필요하다'라는 제약은 activity들이 가지고 있는 멤버 함수를 이용한다. 예를 들어, 위에서 설명한 preJob이라는 activity수행에 man이라는 자원이 3만큼 필요하다면, 다음과 같이 표현한다.

```

m.add(preJob.requires(man, 3));
    
```

requires라는 함수의 선언부는 다음과 같다.

```

IlcResourceConstraint
    
```

```

IlcIntervalActivity::requires(IlcCapResource resource, IlcInt capacity = 1);
    
```

이용되는 자원의 특성과 수행되는 activity의 형태에 따라 자원 요구에 대한 다양한 함수들을 제공하고 있다. 그림 3에는 activity와 이용되는 resource에

대한 관계도이다.

4. 제약만족기법을 이용한 탑재 일정계획 모형

제약만족기법과 객체지향 모형화를 지원하는 일반적 도구인 ILOG Scheduler를 이용하여 문제를 모형화한다. 문제의 대상이 되는 탑재 공정에 대해 4.1절에서 설명하고, 본 연구에서 대상으로 삼은 문제의 범위에 대해 4.2절에서 설명하고, 4.3절에서는 문제에서 고려해야 할 현장의 업무 규칙에 대해 설명하고, 4.4절에서는 ILOG Scheduler를 이용한 탑재 일정계획의 모형을 설명하고, 4.5절에서는 수립된 모형을 이용하여 예제 풀이한 결과를 제시한다.

4.1. 탑재 공정

탑재는 G/C나 크레인을 이용하여 PE장이나 PE장의 공간에 위치한 블록(또는 대형블록)을 도크에 세우는 작업인데, 탑재순서는 탑재공법(ring식, 층식, 피라미드식)에 따라 영향을 많이 받고, 기점블록탑재(K/L : Keel Laying), 중간진수(F/O : Floating Out)와 진수(L/C : Launching) 일정에 맞추어 미리 계획된 선각 및 의장, 도장작업이 수행된다[4].

선각작업은 일반적으로 심출, 취부, 용접 작업 순으로 이루어지는데, 심출은 탑재 후 대형블록의 고정작업을 말하며, 취부는 탑재된 블록의 정위치 배열 및 가접작업, 용접은 탑재된 블록의 접합면을 결합하는 최종 용접작업을 말한다. 이 중 심출 작업은 크게 두가지의 경우로 나눌 수가 있다. 그 중 하나가 one time setting이 가능한 경우이다. 이 경우에는 블록간의 접합면에 대한 다듬기 작업이 PE장에서 모두 이루어지기 때문에, G/C가 탑재 블록을 탑재할 때 심출 작업이 별도로 필요하지 않다. 다른 하나는 one time setting이 불가능한 경우인데, 이 경우에는 G/C가 탑재 블록을 들고 있는 동안 접합면에 대한 고르기 작업 즉 심출 작업을 별도로 수행한다.

탑재에 이용되는 중요 자원으로는 PE장의 정반 면적, 심출, 취부, 용접을 담당하는 인력, G/C로 볼 수 있다. 또한 탑재 블록들은 탑재전 PE장에서의 작업을 위해 소요되는 시간, G/C 이용 시간, 탑재에 필요한 인력시수(man-hour로 표현됨)를 블록별로 가지고 있다. 또한 선행블록 및 후행블록 정보와 min pitch time이라는 정보를 선·후행 관계가 존재하는 블록쌍에 대해서 가지고 있다. 선행 블록의 탑재 후 특정 시간(절대적 시간)이 지난 후에야 후행 블록의 탑재가 가능한데, 이 시간을 min pitch time이라 한다.

4.2. 본 연구에서 고려한 일정계획의 범위

본 연구에서는 도크내 탑재 작업을 대상으로 일정계획을 수립했다. 대상으로 삼은 작업의 범위는 PE장 입고에서 탑재 완료시까지 도크에서 행해지는 선각 작업이다. 대상으로 삼은 기간은 1 batch(도크내로 해수의 유입이 이루어지는 사이 기간이며, 이 기간동안 도크내에서 작업이 가능하다.) 동안이며, 탑재 네트워크에 등장하는 모든 블록과 주요 장비 설치를 일정수립의 대상으로 삼았다.

4.3. 탑재 일정 모형의 입력, 출력, 제약사항

선박의 도크내 탑재일정계획 문제의 입력, 출력, 제약 사항, 일정계획의 수행도 중 본 연구에서 다루는 문제의 범위와 관련 있는 사항을 다음에 정리한다. 제약 사항들은 실제 중공업의 업무 규칙에서 도출될 수도 있는데, 탑재와 관련된 업무 규칙도 같이 정리한다.

<탑재일정계획 문제의 입력자료>

- ① PE장의 면적
- ② 선각 작업의 인력
- ③ 대일정 선표의 batch당 net day
- ④ 호선별 탑재 정보(순서, 공기, 시수), 진수까지의 탑재 블록
- ⑤ 블록별 PE장에서의 점유면적과 점유기간 정보
- ⑥ 탑재블록별 G/C 시수
- ⑦ 선각과 관련된 주요 장비 설치

<일반적인 제약 사항>

- ① PE장의 정반 면적 부하
- ② 심출, 취부/용접의 인력부하
- ③ 블록별 탑재 순서와 min pitch time
- ④ G/C의 일일 작업 부하

<심출/취부/용접 작업에 관한 업무 규칙>

- ① 심출과 취부/용접은 연속 작업으로 이루어지지 않을 수 있다.
- ② 취부/용접은 연속 작업으로 이루어진다.
- ③ 특정 블록의 취부/용접은 초기 투입된 인력이 끝까지 담당한다.
- ④ One time setting이 가능한 블록은 심출 작업이 탑재 당일 이루어진다.
- ⑤ One time setting이 불가능한 블록은 심출 작업이 두 번 이루어진다.

- ⑥ One time setting이 불가능한 블록의 첫 번째 심출 작업은 탑재 당일 이루어진다.
- ⑦ One time setting이 불가능한 블록의 두 번째 심출 작업은 발판 설치가 끝난 날(보통 탑재 이틀 후) 이루어진다.
- ⑧ One time setting이 불가능한 블록의 심출 시수는 대부분 두 번째 심출 작업에 소요된다.
- ⑨ One time setting이 불가능한 블록의 두 번째 심출 작업은 보통은 하루에 이루어지지만, 경우에 따라서는 이틀 이상 소요될 수 있다.
- ⑩ One time setting이 불가능한 블록의 두 번째 심출 작업이 이틀 이상 소요될 경우, 두 번째 날부터는 취부/용접 작업과 병행할 수 있다.

<PE장 점유에 관한 업무 규칙>

- ① 어느 블록은 PE장에서 특정 기간 작업을 마치고 난 후, 도장 작업을 위해 PE장을 나왔다가 탑재되는 경우가 있다. 이때, 탑재 당일엔 PE장 점유 없이 바로 탑재된다.

<불확실한 업무 규칙에 대한 가정>

- ① 블록당 취부/용접의 최대 공기와 최소 공기가 정해질 수 있다.
- ② 선행블록 탑재 후 후행블록 탑재까지는 일정한 시간(min pitch time)이 필요한 경우가 있는데, 이 시간 중 선행블록의 취부/용접 공정의 진행 정도에 따라서 결정되는 시간은 취부/용접의 시수 중 일부로 표현될 수 있다.

<출력 사항>

- ① 각 블록별 PE장 입고 일자

- ② 블록별 도크내 탑재일
- ③ 블록별 도크내 심출 완료일
- ④ 블록별 도크내 취부/용접 착수일, 완료일
- ⑤ 도크에서 이용되는 G/C 부하
- ⑥ 도크에서의 심출/취부/용접의 일별 시수 부하,
- ⑦ 도크에서의 취부/용접의 일별 시수 부하

<고려 가능한 일정계획의 수행도>

- ① 모든 블록의 탑재 완료 시점(진수 시점) 최소화
- ② 심출/취부/용접 부하 평준화
- ③ 심출 부하 평준화
- ④ 취부/용접 부하 평준화
- ⑤ G/C 부하 평준화
- ⑥ PE정반 면적 부하 평준화

4.4. 탑재일정 모형 수립을 위한 업무 규칙 해석

ILOG Scheduler를 이용하여 일정계획을 모형화하는 과정은 activity와 resource를 정의하고 이들 관계에 대해 서술하는 것으로 볼 수 있다. 탑재일정 모형의 수립을 위해 업무 규칙을 중심으로 activity와 resource 관계를 살펴본다.

탑재 공정 상황을 모형화하기 위해, 각 블록의 탑재 activity를 다음 그림 4, 그림 5와 같이 sub activity로 분해했다. 그림 4는 one time setting이 가능한 블록의 sub activity 분해도이고, 그림 5는 one time setting이 불가능한 블록의 sub activity 분해도이다. 분해도에서 타원은 activity이고, 아크는 선행관계를 각각 나타낸다. Sub activity는 각기 공기, 사용 자원, 요구되는 자원의 양을 특성으로 갖는다. 이들 activity는 시작 시각과 종료 시각을 유지하며 일정계획을 수립한 결과로 이러한 시각들이 결정된

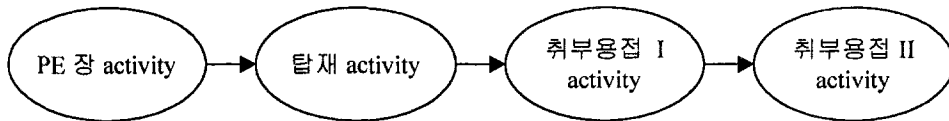


그림 4. one time setting이 가능한 블록의 sub activity 분해도



그림 5. one time setting이 불가능한 블록의 sub activity 분해도

다.

PE장 activity는 탑재일정계획 문제의 입력자료의 'PE장 점유기간'동안 PE장 정반이라는 자원을, 입력자료의 '점유면적'만큼 이용한다. PE장에서의 소요기간 업무 규칙 측면에서 보면, 블록들을 두 가지 부류로 나눌 수가 있다. 그 중 하나는 'PE장 점유기간'이 지난 후 PE장 밖으로 이동이 필요한 블록이고, 다른 하나는 'PE장 점유기간'이 지난 후 탑재 activity가 시작될 때까지 PE장에 계속 남아있는 블록이다. PE장 밖으로 이동이 필요한 블록은 입력자료의 'PE장 공기'와 'PE장 점유기간'을 비교해 보면 알 수 있는데, 공기가 점유기간보다 큰 블록은 이동이 필요하다. 이 경우 PE장 activity는 'PE장 점유기간' 동안에만 정반 자원을 이용한다. 그 외의 블록은 후속 activity인 탑재 activity가 발생할 때까지 정반 자원을 이용하게 된다. 또한, PE장에 입고되지 않는 단독탑재, 주요장비설치 등에 대해서는 PE장 activity의 점유 기간을 0으로 한다.

탑재 activity는 실제 탑재가 이루어지는 상황을 표현한다. 탑재 activity의 작업 공기는 하루 미만이지만 일정 수립의 단위를 일로 가정했으므로 최소단위인 하루로 설정한다. 탑재 activity는 G/C, 심출 시수라는 두 종류의 자원을 이용한다. G/C 자원을 입력자료의 '탑재블록별 G/C시수1(분으로 표현됨)'만큼 이용하고, 심출시수를 입력자료에 기록된 만큼 이용한다. 단, 그림 5와 같이 정위치 탑재 activity가 후속 activity로 존재하는 블록의 경우 대부분의 심출 작업이 정위치 탑재시 이루어지므로, 탑재 activity에서의 심출시수 자원 이용은 없는 것으로 간주한다.

그림 5와 같이 탑재 activity 후에 정위치 탑재

activity가 있는 경우에 정위치 탑재는 탑재 2일 후(그 이상일 수도 있음) 시작되는 것으로 간주한다. 정위치 탑재 activity는 G/C, 심출시수 자원의 두 종류 자원을 이용한다. G/C 자원을 입력자료의 '탑재블록별 G/C 시수2(분으로 표현됨)'만큼 이용하고, 심출시수를 입력자료의 심출시수 만큼 이용한다. 정위치 탑재의 공기는 하루이다.

취부용접 I activity와 취부용접 II activity는 후행 블록의 탑재를 고려해서 취부/용접 activity를 다시 구분했다. 취부용접 I과 취부용접 II activity는 취부/용접 시수라는 자원을 확보해서 작업공기동안 이용한다. 작업공기는 미리 정의된 최대 작업공기(표준공기)와 최소 작업공기(표준공기-줄일 수 있는 공기) 사이의 값을 갖는 변수로 처리한다. 취부용접 I과 취부용접 II의 작업시수 즉, 취부용접 I 작업공기와 취부용접 II 작업공기 동안 확보되는 취부/용접 시수가 입력자료의 취부/용접 시수 보다 클 수 있도록 한다.

Min pitch time은 후행 블록의 탑재전에 선행 블록 탑재에 필요한 장비, 의장과 도장의 공기, 선행블록의 안정성(후행블록을 올려도 될만큼의 취부/용접 진행)등을 고려해서 경험적으로 결정된다. 이러한 min pitch time과 관련된 항목 중 선행블록의 안정성은 취부/용접 시수 중 특정 부분(예를 들어 40% 공정진행)이 진행된다면 확보되는 사항인 만큼, min pitch time중 이와 관련된 부분은 제외할 수 있다고 가정했고, 이를 모형에 반영하기 위해 취부/용접을 두 개의 activity로 구분했다.

다음 그림 6과 그림 7에서는 이와 관련된 제약 사항을 그림으로 나타내었다. 그림 6은 one time setting이 이루어지는 경우를, 그림 7는 그렇지 않은

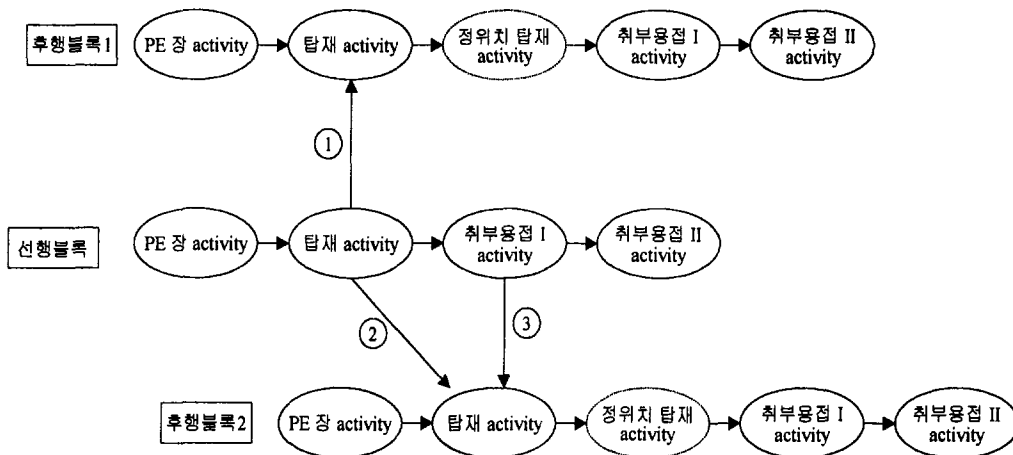


그림 6. 선행블록이 one time setting이 가능한 경우 후행 블록과의 관계

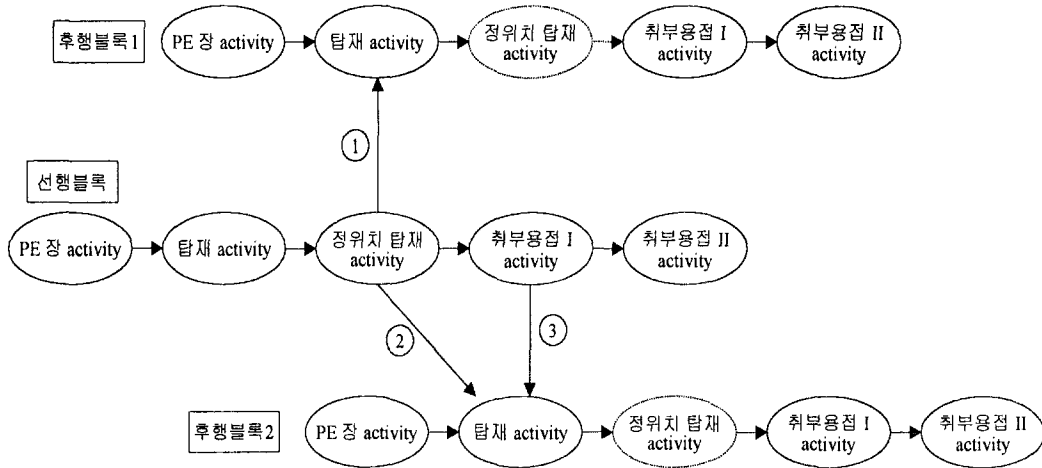


그림 7. 선행블록이 one time setting이 불가능한 경우 후행 블록과의 관계

경우를 각각 표현했다. 각 그림에서 후행블록1과 후행블록2의 정위치 탑재 activity가 점선인 것은 각 블록들이 one time setting이 가능한 경우도 있고, 그렇지 않은 경우도 있음을 의미한다.

후행블록1은 선행블록의 탑재 후 min pitch time(가접과 관련된 부분을 제외함)이 지나면 곧바로 탑재가 가능한데, 이 경우 ①의 선후행 제약이 작용한다. 즉 선행블록의 탑재 activity가 끝나면 후행블록1의 탑재가 가능한 것이다. 후행블록2는 선행블록의 탑재 후 min pitch time(취부용접 I 과 관련된 부분 제외)이 경과되고, 또한 선행블록의 취부용접 I이 끝나야만 탑재가 가능하다. Min pitch time과 관련된 제약은 ②의 제약이고 가접 관련된 제약은 ③의 제약이다.

4.5. ILOG Scheduler를 이용한 예제 수행 결과

D조선의 하나의 batch에 등장하는 블록 및 중요 장비 설치 정보를 이용하여 예제를 만들고 이를 풀이했다.

실험에 이용된 자료에는 총 4척의 배가 조립되는데 이 중 두척은 중간 진수에서 진수까지, 나머지 두척은 기점블록 탑재에서 중간 진수까지이다.

탑재 블록의 총 수는 208개이고, 선·후행 관계 제약은 400개이고, 선정된 수행도는 makespan의 최소화이다. 예제의 풀이에 이용된 컴퓨터는 펜티엄 PC이며, 최적해를 도출할 때까지의 수행시간은 1분이었다. 예제 풀이 결과를 바탕으로 그린 배 한척의 탑재 일정표를 그림 8에 도시했다. 각 행은 블록을 의미한다.

5. 결론 및 추후 연구 방향

조선 산업은 숙련된 기술자와 고부가가치를 수반하는 첨단 설비를 총동원하는 노동 및 기술 집약적 산업이다[6]. 이러한 현실에서 자원 활용을 극대화하기 위한 생산 계획 및 관리는 조선 산업의 국가 경쟁력 제고 측면에서도 매우 중요한 사안이라고 할 수 있다. 생산 계획 및 관리의 기능 중 일정계획은, 수행되어야 할 작업들에 대한 자원의 할당과 작업의 진행 순서 결정을 담당하는 중요한 기능이다.

본 연구에서는 조선 산업에서의 일정계획 중 가장 중요한 분야 중의 하나인 탑재일정을 제약 만족기법과 객체 지향을 지원하는 ILOG Scheduler를 이용하여 실제 탑재 일정과 관련된 업무 규칙 및 제약을 고려하여 모형화 하였고, 예제 문제를 형성하여 풀이했다. ILOG Scheduler를 이용한 현장의 업무 규칙 모형화는 비교적 용이하며, 실제 크기의 문제에 적용한 결과, 수행도(수행 속도와 해의 우수성) 측면에서도 뛰어난 결과를 확인할 수 있었다.

추후 연구 방향은 ILOG Scheduler를 이용한 탑재 일정 모형에서, 현재는 고려되지 않았던 업무 규칙에 대해 추가적인 모형화를 수행하고, 선행 도장공 기관 의장 등 조선 산업의 타 일정계획 부분에 대한 모델링과 프로토타입 시스템을 개발할 예정이다.

참고문헌

- [1] 김훈주, "유전 알고리즘을 이용한 조선 탑재 공정 계획 연구," 석사학위 논문, 인하대 자동화공학과, 1995.
- [2] 대동조선, "PE정반 Arrangement 시스템," 조선 생산 기술연구회, 1999년 춘계 연구발표회, 1999.

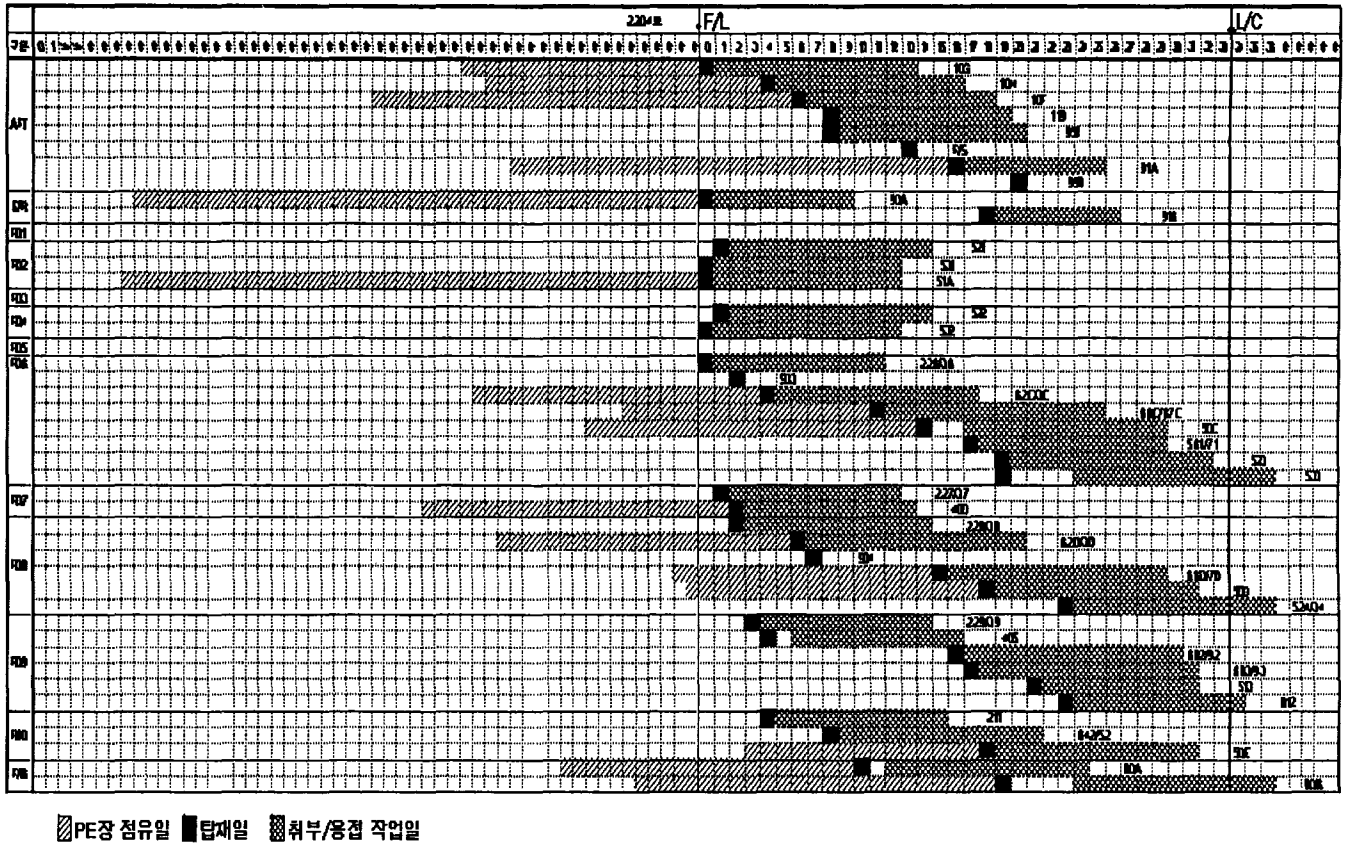


그림 8. 예제풀이 결과를 이용한 탑재 일정표 (4개 호선 중 1호선)

[3] 박주철, 황하룡, "조선 기준 일정계획을 위한 재계획 절차의 개발," 산업공학지, 제11권, 제3호, pp.129-14, 1998.

[4] 우상복, "대우조선의 제조 프로세스 및 생산관리 기술," Technical Paper, 고등기술연구원 생산기술연구실, 1999.

[5] 이경준외, "Spatial Scheduling and its Application to Shipbuilding," 제2회 환태평양 인공지능 국제학술 회의, Seoul, Korea, Vol. II, pp.1183-1189, 1992.

[6] 홍윤기외, "조선 공정계획에서 탑재 순서 생성," 산업공학지, 제 10권 제 1호, pp.189-207, 1997.

[7] Hyung Rim Choi, "Erection Scheduling at Shipbuilding Using Constraint Directed Graph Search : DAS-ERECT," 박사 학위 논문 draft, KAIST 경영정보공학과. 1994.

[8] ILOG, "ILOG Scheduler 4.3 Reference Manual," ILOG, 1998.

[9] ILOG, "ILOG Solver 4.3 Reference Manual," ILOG, 1998.

[10] K. R. Baker, "Introduction to sequencing and scheduling," John Wiley & Sons, 1974.