

제약만족기법을 이용한 조선 블록조립 공정의 일정계획에 관한 연구

Scheduling of Shipyard Block Assembly Process using Constraint Satisfaction Problem

김희문, 강장하, 박성수
한국과학기술원 산업공학과

Abstract

본 연구에서는 선박 건조 공정 중 블록조립공정을 다룬다. 블록과 정반(bay)의 정보가 주어질 때, 각 블록의 조립작업을 하게 될 정반과 조립작업을 시작하는 시작시점을 결정해야 한다. 정반에서 중요하게 고려할 두 가지 자원은 바로 면적과 시수이다. 본 연구에서 다루는 문제의 목적식은 정반에 할당되지 못하거나 작업시점을 결정하지 못해 작업을 할 수 없게 된 실패 블록의 수를 최소화시키는 것과 시수 부하의 부하평준화를 이루는 것이다.

이 문제를 풀기 위해 제약만족기법(CSP)에 기반을 두는 알고리즘을 제안한다. 그리고 이 알고리즘은 실제 데이터를 가지고 실험을 하게 되며 그 결과와 기존의 일정계획 결과와 비교 분석하게 될 것이다.

1. 서론

일반적으로 선박을 건조할 때에는 조선소의 생산능력과 건조하려는 선박의 특성에 따라서 블록이라는 단위로 분해를 해서 먼저 제작을 한 다음 이들을 조립공정과 탑재공정을 통해 한 척의 배로 완성하게 된다. 블록 조립 작업장에서의 작업은 도크(dock)에서의 탑재 일정에 맞추어서 결정이 되는데, 이 범위 내에서 조선소의 생산 능력을 고려하여 일정계획을 수립하는 것이 본 연구의 대상이 되는 블록 조립 공정의 일정계획이다.

블록 조립 작업장을 정반(bay)이라고 부른다. 정반은 크게 고정정반과 이동정반으로 구분된다. 고정정반에서는 주로 구부러진 모양의 곡블럭을 조립하게 되고, 이동정반에서는 주로 평블럭을 조립하게 되며, 작업이 끝나게 되면 이동정반이 공장밖으로 이동하게 되는 것이다. 본 연구에서는 고정정반에서의 일정계획을 다루고 있다.

조선공정의 일정계획 문제는 자원제약하의 프로젝트 스케줄링 문제(RCPSP)에 속한다. RCPSP는 각 작업의 시작시점을 조정하여서, 자원제약 조건을 만족시키고 작업간의 순

서 제약을 만족시키면서 총 작업시간을 최소화시키는 문제이다. 이 문제를 해결하기 위한 방법으로는 경영과학(OR)적 접근방법과 인공지능분야에서 사용되는 제약만족기법(CSP)을 통한 접근방법이 있다.

조선 공정의 일정계획에 관한 기존연구를 간략히 살펴보면 다음과 같다. 민상규[1]는 탑재 공정에 대한 유전알고리즘을 개발하였다. 백태현[2]은 탑재공정에 대한 자원평준화 휴리스틱을 개발하였다. 박경철[3]은 time window 제약과 면적 제약을 만족하면서 조선 블록조립공정의 일정계획 알고리즘을 개발하였다. 이들 연구는 대부분 OR적 접근방법을 이용한 휴리스틱 알고리즘을 개발하여 사용하고 있다. 본 연구에서는 CSP를 이용하여 일정계획을 수립하고자 한다.

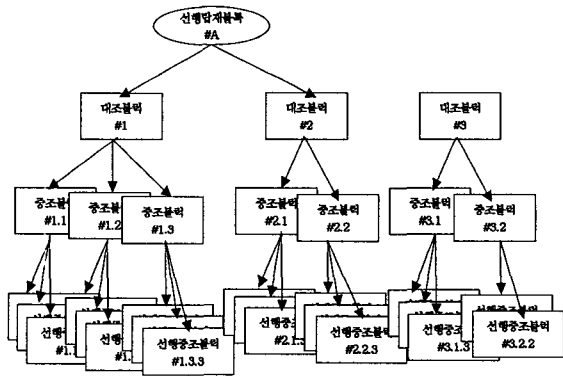
본 논문의 구성은 다음과 같다. 2장에서는 연구대상인 고정정반에서의 블록조립공정을 설명하고, 3장에서는 제약만족기법을 이용하여 모델링을 하게 된다. 4장에서는 해를 구하기 위한 알고리즘을 소개하게 되고, 5장에서는 실제 데이터를 이용한 실험 결과를 보여줄 것이다. 6장에서는 결론과 앞으로의 연구방향을 제시하게 된다.

2. 고정정반 블록조립 공정

블록조립공정은 선박 건조의 기본 자재인 강재를 가공한 다음, 이를 조립하여 도크에서 탑재할 블록을 만드는 과정이다. 조립공정의 일정계획을 세울 때 사용되는 자원으로는 정반의 면적, 작업인력, 기타 장비 등으로 구분할 수 있는데, 본 연구에서는 정반의 면적과 작업 인력을 자원으로 사용하게 된다.

정반은 매우 넓은 공간을 가지고 있으나 조립하게 되는 각 블록들의 모양과 차지하는 면적이 제각각이기 때문에 면적과 모양을 동시에 고려해야 하는 어려움이 있다. 따라서 기계의 이용률보다는 면적 활용율이 더 중요한 자원이 될 수 있다. 그리고 작업 인력은 작업자 수로 관리하는 것이 아니라 시수(man power)단위로 관리한다. 정반별 작업 인력은 이동이 용이하지 않고 노동 집약화 되어 있기 때문에 공장의 생산비용을 좌우하는 중요한 요소가 된다. 따라서 본 연구에서는 정반의 면적과 작업 인력을 고려할 때 면적활용율과 시수활용율을 함께 사용하게 될 것이다.

블록조립 공정에서 조립하게 되는 블록은 크게 대조블럭, 중조블럭, 선행중조블럭으로 구분된다. 각 블록들의 포함관계는 [그림 1]에 나타나 있다. 대조블럭들이 몇 개 모여서 옥외 작업장에서 미리 함께 조립한 다음 탑재공정으로 옮겨지는 경우가 있는데, 이렇게 만들어진 블록을 PE(Pre-Erection)블럭이라고 하며, 이 과정을 선행탑재라고 부른다.



[그림 1] 블록간의 관계 구조

[그림 1]에 나타난 블록들 중에는 고도의 기술을 요구하는 작업이 매우 어려운 블록들이 존재하는데 ER/DB(Engine Room Double Bottom)블럭이 이에 해당한다. 이 블록들 간에는 작업 시점에 대한 제약이 따른다.

여러 제약 조건을 만족하면서 정반에 블록을 할당시킨다 하더라도 정반상에 형상 배치시킬 때 작업을 할 수 없는 블록이 발생할 수 있다. 본 연구에서는 형상배치는 고려하지 않는다. 다만 현장의 경험을 토대로 해서 정반

의 면적활용율을 이용하여 할당된 블록은 모두 작업이 이루어질 수 있을 만큼만 블록을 할당시킨다.

3. 제약만족기법을 이용한 모델링

제약만족문제(CSP)는 주어진 제약조건을 만족하는 가능해를 찾는 것을 목적으로 하는 문제이며, 제약만족기법은 CSP의 해를 구하는데 사용되는 방법이다. 본 절에서는 CSP에서 해를 찾아가는 과정을 소개하고, 이를 이용한 블록조립 공정에 대한 모델링을 소개하고자 한다.

CSP는 변수(variable), 도메인(domain), 제약식(constraint)으로 구성되어 있다. 변수들의 집합은 $X = \{x_1, x_2, \dots, x_n\}$ 이며, 각 변수 x_i 가 취할 수 있는 값들의 집합을 도메인이라고 부른다.

3.1. 변수

블록조립 공정의 일정계획에서의 변수들은 블록 i 의 조립 작업을 하게될 정반 b_i 와 이 블록의 조립작업을 시작하는 시작 시점 t_i 가 있다.

3.2. 도메인

변수 b_i 의 도메인은 조선소 내에서 작업할 수 있는 정반이 된다. 대조블럭, 중조블럭, 선행중조블럭의 경우 블록의 특성상 작업할 수 있는 정반이 보통 달라지게 된다. 그리고 같은 종류의 블록이라고 하더라도 선호하는 정반이 서로 다를 수 있고 특수한 블록은 정해진 정반에서만 작업을 해야 하기 때문에 블록들간의 도메인은 서로 달라질 수 있다. 각 블록들의 정반도메인 내에는 실패정반을 가지는 것으로 한다. 실패정반이란 일정계획 수립에 사용된 정반에 할당되지 못하는 실패 블록들을 할당시키는 가상의 정반이다.

변수 t_i 의 도메인은 각 블록이 가지고 있는 SSD(Schedule Start Day, 옥외기준일정)정보에 의해 결정된다. 그런데 중조블럭과 선행중조블럭의 경우 SSD가 존재하지 않기 때문에 이들 블록들은 블록들간의 버퍼제약에 의해 도메인이 결정된다. 모든 대조블럭들은 동일한 SSD-ESD가 값을 가지고 있다. 이 값에 의해 ESD(Earliest Start Day)가 결정된다. 시작시점도메인내의 값의 단위는 시(hour)가 아니라 일(day)이다. 만일 어떤 블록의 SSD가 25이고 SSD-ESD가 5라고 한다면 이 블록의 시작시점도메인은 {20,21,22,23,24,25}가 되는 것이다.

3.3 제약조건

본 절에서는 가장 중요한 몇 개의 제약을

한국과학기술원(KAIST) 2002년 5월 3일~4일
살펴보기로 한다. 먼저 제약조건과 알고리즘에
서 사용할 notation을 살펴보면 다음과 같다.

- L : 대조블럭들의 집합
- M : 중조블럭들의 집합
- P : 선행중조블럭들의 집합
- B : 정반들의 집합
- N : 총블럭의 수
- SSD_i : 대조블럭 i 의 SSD
- ESD_i : 대조블럭 i 의 ESD
- α : $SSD_i - ESD_i, \forall i \in L$ 에 대해 일정하다
- d_i : 블럭 i 의 조립작업시간
- te_i : 블럭 i 의 작업종료시간
- tb_{ij} : 블럭 i 와 블럭 j 사이의 버퍼(buffer)
- β : PE배치 최대간격
- δ : ER/DB 착수 일자 간격
- pl_ν : 한 PE블럭집합 ν 를 구성하는 대조블럭
들의 작업종료시점의 하한치
- pu_ν : 한 PE블럭집합 ν 를 구성하는 대조블럭
들의 작업종료시점의 하한치
- $M(i)$: 대조블럭 i 를 구성하는 중조블럭들의
집합
- $P(j)$: 중조블럭 j 를 구성하는 선행중조블럭들
의 집합
- s_b : 정반 b 의 면적
- sb_i : 블럭 i 에 의해 요구되는 면적
- R_S : 면적활용율
- m_b : 정반 b 에서 하루동안 사용가능한 시수
- mb_i : 하루동안 블럭 i 의 작업에 필요한 시수
- R_M : 시수활용율
- $\max - R_M$: 최대시수활용율
- $x_{i,b,t}$: 블럭 i 가 정반 b 에 시점 t 에 할당되면 1,
그렇지 않으면 0
- N_f : 실패블럭의 수
- U_f : 실패블럭의 수의 상한치
- $TIME$: 일정계획 기간내의 시간들의 집합

모든 대조블럭들은 각 블럭들의 ESD와
SSD 사이에서 작업이 시작되어야 한다.

(제약조건 1)

$$ESD_i \leq t_i \leq SSD_i, \text{ where } ESD_i = SSD_i - \alpha, i \in L$$

같이 선행탑재되는 대조블럭들간에는 조
립 착수 시점에 제약이 따르게 된다. 하나의
PE블럭을 형성하는 대조블럭들은 작업 시점
이 서로 다르므로 반드시 동시에 작업이 종료
될 수 있는 것은 아니다. 따라서 PE 배치 최
대 간격이라고 하는 대조블럭들의 작업 종료
시점의 간격을 두게 되고 반드시 이 시간 간
격내에는 작업이 종료되게 만든다. 하나의 PE
블럭 집합 ν 내의 블럭중에서 가장빨리 작업을
종료하는 시점을 pl_ν , 가장 늦게 작업이 종료
하는 시점을 pu_ν 라고 할때 다음의 관계가 성

립한다.

(제약조건 2)

$$pl_\nu \leq te_i \leq pu_\nu, pl_\nu + \beta \geq pu_\nu, \forall i \in \nu \text{ and } i \in L$$

ER/DB블럭은 작업장에서 시수부하가 많
이 걸리고 블럭의 특수성으로 인해 작업에 여
러 가지 어려움이 따르게 된다. 따라서 동시
한 정반에 서로 다른 두 개 이상의 ER/DB블
럭의 조립 작업을 진행할 수 없다. ER/DB블
럭이 같은 정반에 할당되기 위해서는 시작시
점 t_i 에 어느 정도의 차이가 있어야 한다. 이
간격을 ER/DB 착수일자 최소간격이라고 부른
다. 예를 들어, 두 개의 ER/DB 블럭 E_1 과 E_2
가 같은 정반에 할당되었다고 하자. 각 블럭의 작
업 시작 시점은 t_{E_1} 과 t_{E_2} 가 된다. 이들 사이
는 $t_{E_1} + \delta \leq t_{E_2}$ 또는 $t_{E_2} + \delta \leq t_{E_1}$
이 성립해야 한다. 따라서 다음의 관계가 성립한다.

(제약조건 3)

$$\sim (b_{i_1} = b_{i_2}) \vee (t_{i_1} + \delta \leq t_{i_2}) \vee (t_{i_2} + \delta \leq t_{i_1})$$

for all possible pair of ER/DB blocks i_1, i_2

대조블럭의 작업시작시점과 이것을 형성
하는 중조블럭들의 작업종료시점 사이에는 버
퍼(buffer)가 존재한다. 중조블럭 $j \in M(i)$ 의 작
업종료시점 te_j 는 대조블럭 i 의 작업시작시점과
중조블럭 j 와의 버퍼에 의해 결정되는 것이다.
하나의 대조블럭을 만드는 중조블럭들 중
에서 하나는 메인(main)중조블럭이 되며 이 블
럭의 버퍼(buffer)는 0이다. 나머지 중조블럭들의 버
퍼는 양의 값을 가진다. 중조블럭과 이를 형성
하는 선행중조블럭간에도 마찬가지로 버퍼 제
약이 존재한다.

(제약조건 4)

$$te_j \leq t_i + tb_{ij}, \forall i \in L, \forall j \in M(i)$$

$$te_l \leq t_k + tb_{kl}, \forall k \in M, \forall l \in P(k)$$

각 정반에 할당된 블럭들에 의해 사용되
어지는 면적과 시수 자원의 양은 일정계획기
간에 걸쳐서 사용 가능량을 초과하지 말아야
한다. 각 정반에서는 잔업과 할당된 블럭들의
형상 배치 제약 등으로 인해 사용 가능한 시
수와 면적의 양을 조금씩 조정할 수 있으므로
시수활용율과 면적활용율을 사용할 것이다.

(제약조건 5)

$$\sum x_{i,b,t} \times sb_i \leq R_S \times s_b, \forall b \in B, \forall t \in TIME$$

$$\sum x_{i,b,t} \times mb_i \leq R_M \times m_b, \forall b \in B, \forall t \in TIME$$

마지막으로 본 연구의 목적식을 설명하고
자 한다. 조선공정에서의 목적은 많이 존재하
겠지만 본 연구에서는 두 개의 목적을 고려한
다. 하나는 실패 블럭의 수를 최소로 만드는
것이고 다른 하나는 시수부하평준화이다. CSP
에서는 제약식을 모두 만족시키는 해를 찾아

한국과학기술원(KAIST) 2002년 5월 3일~4일

주기 때문에 문제의 목적도 제약식처럼 모델링에 추가해서 원하는 해를 찾아나가게 된다.

시수부하평준화를 이루지 못하면 정반마다 비슷한 수의 블록이 할당되더라도 특정정반에 할당된 블록들이 시수 부하를 많이 요구할 수 있게 되어 정반마다의 작업량에 편차가 생길수 있다. 그러므로 정반별 시수평준화를 이루어야 한다. 그런데 각 정반의 시수양이 다르므로 시수 활용율에 대한 평준화를 이루게 할 것이다. 이것은 전체 스케줄 기간동안에서 각 정반에서 최대시수활용율을 가능한 낮추어 감으로써 정반마다의 최대시수 활용율을 비슷하게 만들 수 있다. 시수부하 평준화에는 기간별 평준화도 이루어야 한다. 이것은 일별 평균 부하 변동이 가능한 작게 만드는 것이다.

4. 알고리즘

본 절에서는 해를 구하는 알고리즘을 설명하게 된다. 4.1절에서는 알고리즘의 전체적인 개요를 보이고 4.2절에서는 알고리즘을 통하여 탐색트리를 형성하는 과정을 보일 것이다.

4.1. 알고리즘의 개요

시수활용율을 가능한 최대치로 설정하고 실패블럭의 수는 총 블럭의 수로 초기화한다. 초기가능해가 구해지고 나면 해의 개선단계에서 최대시수활용율을 현재 해의 시수활용율에서 일정한 값만큼 감소시킨 값으로 설정한다. 그리고 실패블럭의 수의 최대치는 현재 해의 실패블럭의 수로 설정한다. 이 과정은 더 이상 해의 개선이 이루어지지 않을 때까지 계속한다. 알고리즘의 전체 개요는 [그림 2]에 나타

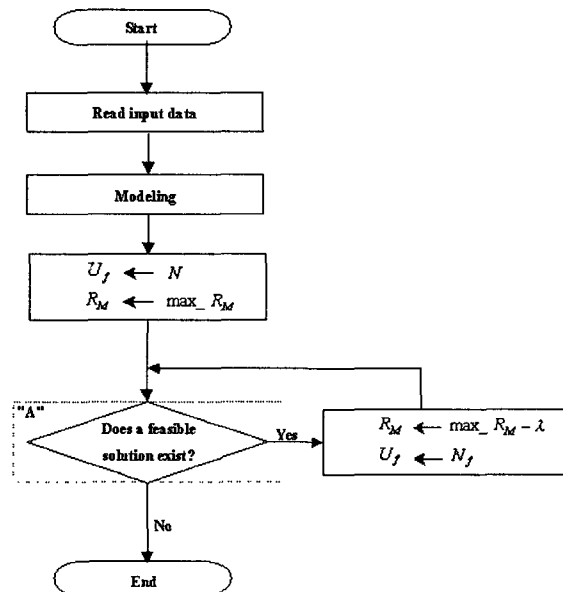


그림 2. 알고리즘

나 있다.

4.2. 탐색트리(search tree)의 구성

본 절에서는 그림 2에서의 "A" 부분을 설명하고자 한다. CSP에서는 가능해를 구하기 위해 모든 변수들에 가능한 값들을 모두 고려해 보는데 이 과정은 탐색트리로 표현이 된다.

탐색트리를 만들어갈 때 한 노드에서 변수의 도메인 내에 가능한 값이 둘 이상 존재할 경우 이 노드를 두 개의 자식 노드로 분지(branching)한다. 이러한 과정은 모든 변수의 값이 결정될 때까지 계속한다. 제약식 등으로 인하여 더 이상 변수의 도메인 내에 있는 값이 가능해에 포함될 수 없게 될 때 그 값은 제외시킨다. 한 변수의 값 또는 도메인이 수정이 되면 이 변수와 관련을 맺는 모든 변수들의 도메인에 변화가 생기지는 잘 파악해야 한다. 그리고 다른 변수에서 절대 가질 수 없는 값이 도메인 내에 존재하는 것으로 판단되면 그 값을 제외시킨다. 이러한 과정을 propagation이라고 부른다.

CSP algorithm: sub-function

[단계 0] 탐색트리에서 한 노드를 선택한다.

[단계 1] 블록 선택

할당되지 않은 블록들이 존재하면 한 블럭을 선택한후 단계 2로 간다. 그렇지 않으면 해가 발견되었으므로 종료한다.

[단계 2] 정반 branching

선택된 블록을 할당한 정반이 존재하지 않으면 infeasible하다. 단계 4로 간다.

블록을 할당할 정반이 한 개 존재하면 단계 3으로 간다.

두 개 이상의 정반이 존재하면 두 개의 노드로 branch 한다. 하나의 노드는 블록을 정반 b 에 할당하는 것이고 다른 노드는 b 정반을 제외한 정반에 할당하는 것이다.

[단계 3] 작업 시작 시점 branching

선택된 블록의 가능한 시작시점이 존재하지 않으면 infeasible하므로 단계 4로 간다.

그렇지 않으면 두 개의 노드로

branching한다. 하나는 특정 시점 t 에서 작업을 시작하는 것이고 다른 한 노드는 시작시점 도메인에서 t 를 제외한 다른 시점에 작업을 시작하게 하는 것이다.

[단계 4] sub-function의 종료

이 수행 절차는 그림 3에 나타나 있다.

단계 1에서 블록을 선택할 때 정반 또는 작업시작시점이 결정되지 않은 블록들 중에서 LST(latest start time)를 가지는 블록을 선택

한국과학기술원(KAIST) 2002년 5월 3일~4일

한다. LST는 SSD와 동일한 것은 아니다. 왜냐하면 탐색 트리를 형성하는 과정에서 각 블록의 시작시점도메인은 propagation에 의해 변하게 되기 때문이다. 단계 2에서 정반을 선택할 때에는 정반 도메인 내의 정반 중에서 랜덤하게 선택하면 된다.

단계 1에서 블록이 선택되고 단계 2에서 작업할 정반이 선택되었을 때 단계 3에서 작업시점을 결정하게 된다. 작업시점은 정반의 시간에 따른 시수 부하를 고려해서 결정된다. 앞에서 설명했었지만, 본 연구에서 다루는 일 정계획의 목적 중에는 일별 시수 부하의 변동을 작게 만드는 것이 있다. 그러나 모든 가능한 스케줄 중에서 가장 작은 시수 부하의 변동을 가지는 스케줄을 찾아낸다는 것은 거의 불가능하다. 왜냐하면 탐색트리의 크기가 지수적으로 증가할 수 있기 때문이다. 따라서 좀 더 현실적으로 시수 부하의 변동을 줄일 수 있는 방법으로 작업시점을 결정할 때 정반의 시간에 따른 시수 부하를 고려하는 방법을 사용한다.

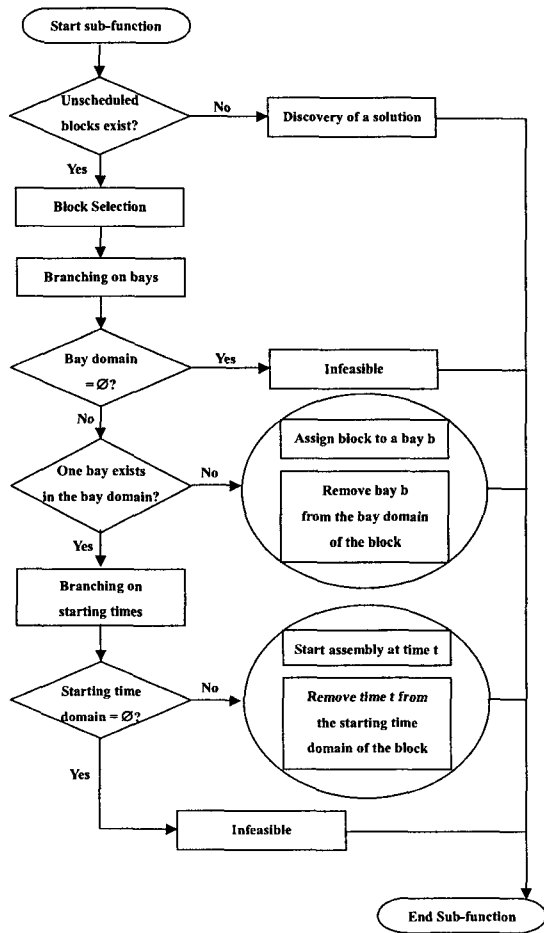


그림 3. sub-function의 순서도

시점 t 에서 정반의 시수부하를 L_t 라고 하

자. 선택된 블록이 i 일 때, $[t, t + d_i]$ 에서 정반의 최대시수활용율을 C_i^{max} 라 두자. 이때 다음을 정의한다.

$$T_i = ((L_{t-1} - L_t) + (L_{t+d_i} - L_{t+d_i-1})) \times \frac{R_M}{C_i^{max}}$$

for all $t \in [ESD_i, SSD_i]$

작업시작시점은 T_i 를 최대화시키는 t 값으로 결정된다. 다음의 예를 보면 이 방법에 의해 시수부하의 변동을 줄일 수 있게 됨을 알 수 있다. 한 정반의 시간에 따른 시수 사용량이 다음과 같다고 하자.

날짜	9	10	11	12	13	14	15	16	17	18	19	20
시수	8	8	6	6	10	8	7	5	4	6	7	9

블록 i 의 작업시간 $d_i=3$ 이라 두고, 시작시점도메인은 $[12,15]$ 라고 두자. 시수활용율 R_M 이 110%라고 하고, 정반의 표준 시수사용가능량을 12라고 두자. t 값의 변화에 따른 T_i 의 값은 다음과 같이 계산된다.

t	12	13	14	15	16	17	18
L_t	6	10	8	7	5	4	6
K	-1	-6	1	3			
C_i^{max}	83%	83%	67%	58%			
T_i	-1.33	-7.95	1.64	5.69			

$$K = (L_{t-1} - L_t) + (L_{t+d_i} - L_{t+d_i-1})$$

$t \in [12, 15]$ 에서 T_i 의 최대값은 5.69이다. 그러므로 작업시작시점은 15일로 결정되는 것이다.

이 방법으로 작업 시작 시점을 결정하게 되었을 때, 일반적으로 부하 평준화 문제에서 나타나는 최대 부하를 최소화시킴으로써 부하 평준화를 이루는 것 외에도 특정 시점에 갑자기 부하량이 작아지는 현상을 막을 수 있다는 장점이 있다.

5. 실험 결과

4절에서 소개된 알고리즘은 두 개의 데이터 set에 의해 실험이 되었다. 각 데이터 set의 구성은 다음과 같다. 이들 데이터는 조선소에서 사용된 실제 데이터이다.

	대조블럭 수	중조블럭 수	선행중조블럭 수	PE블럭 수	Time horizon
set 1	459	308	19	144	174day
set 2	506	255	17	138	174day

사용 가능한 정반은 7개가 있다. 대조블럭은 7번 정반을 제외한 모든 정반에서 작업이 가능하고 중조블럭은 6번과 7번 정반, 그리고 선행중조블럭은 7번 정반에서만 작업이 가능한 것으로 하였다.

실험에서 사용한 parameter들의 값들은 다음과 같다.

- 면적활용율 $\sigma = 80\%$
- 초기 최대시수활용율 $R_M=120\%$
- PE배치 최대간격 $\beta=10$ 일
- SSD-ESD $\alpha=12$ 일
- ER/DB 착수일자간격 $\delta=4$ 일
- 시수활용율 감소치 $\lambda=3\%$

실험은 Pentium 800Mhz에서 실행되었으며, ILOG Solver 5.0과 ILOG Scheduler 5.0을 사용하였다. 가능해를 탐색하기 위해 시간을 20분으로 제한하였다. 이 이상의 시간이 경과한 후에는 더 이상 해가 개선되는 경우가 없었다. 각 데이터 set에 대해 해가 개선되어 가는 과정을 살펴보면 각각 다음과 같다.

표 1. 데이터 set 1에 대한 결과

	max_ R_M	실패블럭수	실행시간(sec)
탐색 1	119	52	9.72
탐색 2	114	44	9.68
탐색 3	109	43	9.71
탐색 4	해의 개선 실패		

표 2. 데이터 set 2에 대한 결과

	max_ R_M	실패블럭수	실행시간(sec)
탐색 1	115	57	9.55
탐색 2	111	47	11.20
탐색 3	107	48	9.49
탐색 4	해의 개선 실패		

표 1과 표 2에서 확인할 수 있듯이 개선된 해가 발견된다면 아주 빠른 시간 내에 발견된다. 하지만 개선된 해의 발견에 실패한 경우에는 오랜 시간동안 해가 존재한다는 것을 증명할 수 없었다. 이것은 본 문제에 대한 탐색 공간의 크기가 매우 크기 때문에 발생한 것으로 판단된다. 제안된 알고리즘에 의한 해는 만족할 만한 것으로 판단되었다.

본 연구에서 나온 결과를 조선소에서 기존에 가지고 있던 일정계획과 비교해 보았다. 그 결과가 표 3에 나타나 있다.

본 연구에서는 제안된 알고리즘을 각 정반에서 일간 시수 부하의 변동량을 감소시키는 효과를 알아보는데 사용하였다. 전체 일정계획 기간동안에 t 일과 $t-1$ 일의 시수 부하의 차이의 절대값을 계산하였고 이에 대한 분산과 표준편차를 통해 비교하였다.

전체 정반이라는 개념을 도입한 것은 각각의 정반별로 시수 부하 평준화를 적용했을

때 모든 정반에서 만족할 만한 결과를 얻을 수 있는 것이 보장되지 못하며 또한 해를 구하기도 어려워진다. 따라서 7개의 정반을 모두 합하여 하나의 정반으로 보고 시수 부하 평준화를 이룬 후 각각의 정반에서 수정을 해나가는 방법을 택한 것이다.

표 3. 시수부하 변동량의 비교

	평균			표준편차		
	기존 스케줄	set 1	set 2	기존 스케줄	set 1	set 2
정반1	15	8.62	11.56	12.23	10.65	12.19
정반2	15.11	8.13	10/19	16.55	12.78	12.51
정반3	18.28	9.23	13.65	13.08	10.15	17.29
정반4	22.89	7.32	6.19	42.82	11.84	9.6
정반5	27.11	9.08	7.95	50.72	10.75	8.85
정반6	16	6.01	5.93	29.93	6.16	8.37
정반7	17.11	4.59	5.68	32.01	5.93	7.82
전체	114.61	19.78	19.09	143.89	20.01	15.95

표 3에 나타난 결과는 기존 시스템에서의 일정계획과 데이터 set 1, 데이터 set 2의 알고리즘 적용 결과에 대한 전체 스케줄 기간동안의 시수 부하의 변동에 대해 단위 시간당 평균값으로 나타낸 것이다. 전체 정반의 시수 부하 변동 비율과 각 정반의 시수 부하 변동 비율을 비교할 때 전체 정반의 경우 약 85% 정도의 개선 효과를 볼 수 있었으며, 일간 변동치의 표준편차를 비교해 본 결과 상당히 많이 줄어들음을 확인할 수 있었다. 이것은 본 연구에서 사용한 알고리즘이 연구의 목적인 시수 부하 평준화에 많은 기여를 하고 있음을 나타내고 있는 것이다.

6. 결론

본 연구에서는 고정정반 블록조립공정의 일정계획 문제를 제약만족기법을 사용하여 구하였다. 블록조립공정의 일정계획문제는 크게 공간 일정계획문제와 부하평준화 문제로 나누어지는데, 본 연구에서는 부하평준화를 다루었다.

본 연구에서 사용한 알고리즘으로는 기존의 시스템으로 찾아낸 일정계획보다 개선된 결과를 얻을 수 있었으며, 부하 평준화를 이루는데 부하의 최대치를 최소화하는 것뿐만 아니라 최소치를 가능한 다른 시점의 부하와 비슷한 수준으로 유지할 수 있는 방법을 사용했다는 점에 의의가 있다.

본 연구에서는 고정정반 블록조립공정만을 다루었는데, 조선 산업의 특성상 블록 조립 공정에서의 일정계획의 최적해가 의장, 도장,

대한산업공학회/한국경영과학회 2002 춘계공동학술대회

한국과학기술원(KAIST) 2002년 5월 3일~4일

탑재 공정에서는 최적해가 되지 못하는 경우가 많이 발생한다. 따라서 한 척의 선박을 건조하기 위한 일정계획을 수립하기 위해 특정 부분의 공정을 따로 계획을 수립하는 것이 아니라 전 공정에 걸쳐 최적해가 될 수 있는 연구가 필요하다.

참고문헌

- [1] 민상규, 이민우, 문치용, 정귀훈, 박주철, 조선 탑재 일정의 부하 평준화를 위한 유전알고리즘, IE Interface, 제13권, 2호, pp225-233, 2000
- [2] 백태현, 정귀훈, 박주철, 부하평준화 휴리스틱을 이용한 선박 탑재 일정계획 편성 연구, IE Interface, 제12권, 3호, pp354-361, 1999
- [3] 박경철, 이경식, 박성수 외, "고정정반 일정계획시스템의 개발", IE Interface, 제8권, 2호, pp. 95-103, 1995
- [4] 이상협, 정귀훈, 하승진, 이병열, 오성훈, 조선 평블럭 조립공장 일정계획 시스템, 한국경영과학회/대한산업공학회 춘계공동학술대회, 2001
- [5] 이화기, 정재원, Constraint Programming을 이용한 자원제약 동적 다중프로젝트 일정계획, IE Interface, Vol. 12, No 3, pp362-373, 1999
- [6] 백영수, 류광렬, 박영만, 김갑환, "제약만족 및 휴리스틱 교정기법을 이용한 선석 및 크레인 일정계획", 한국지능정보시스템학회 99 추계학술대회 논문집, pp. 151-157, 1999
- [7] 조규갑, 이동하, "곡블럭 조립작업에 대한 공간 일정계획에 관한 연구", 한국경영과학회/대한산업공학회 춘계공동학술대회, 2001
- [8] 정귀훈, 백태현, 민상규, 김형식, 박주철, 조규갑, 박창규, "조선 도장공정 일정계획 시스템 개발", 한국경영과학회/대한산업공학회 춘계공동학술대회, 2001