creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사 학위논문

중기계획 부하를 고려한 제약 프로그래밍 기반 장기 계획 최적화

- 조립 공정 부하를 고려한 선표 계획 최적화 모델 -

2023년 08월

서울대학교 대학원

조선해양공학과

조 기 영

중기계획 부하를 고려한 제약 프로그래밍 기반 장기 계획 최적화

- 조립 공정 부하를 고려한 선표 계획 최적화 모델 -

지도 교수 우 종 훈

이 논문을 공학석사 학위논문으로 제출함
2023년 08월

서울대학교 대학원
조선해양공학과
조 기 영

조기영의 공학석사 학위논문을 인준함
2023년 08월

위 원 장 _____ 김 태 완 _____ (인)

부위원장 _____ 우 종 훈 _____ (인)

위 원 _____ 노 명 일 _____ (인)

초 록

선표 계획 과정은 조선소의 생산 계획 중 장기계획에 해당하며 조선소의 연간 운영에 직접적으로 영향을 미친다. 선표 계획은 3년에서 6년 또는 그 이상의 기간에 대한 건조 계획이기 때문에 계획 대상 호선이 많고, 대부분의 조선소에서 계획자의 휴리스틱한 판단에 의해 이루어지기 때문에 하나의 선표를 계획하는데 적지 않은 기간이 소요된다. 또한 선행공정의 부하를 고려하기 위해 다수의 선표를 계획한 뒤 선표 케이스별 부하를 계산하는 방법을 사용하는데, 이 방법은 계획 시간을 증가시키는 요인이 된다. 결과적으로, 특정한 알고리즘 없이 시행착오를 바탕으로 선표를 계획하고 있으며 이는 해 탐색 범위의 한계로 인해 도출된 계획의 최적 여부를 보장할 수 없다는 한계가 존재한다.

본 논문에서는 선표 계획 최적화를 위한 제약조건 및 목적함수를 정의하고 이를 만족하는 최적의 선표를 도출하는 제약 프로그래밍 기반의 모델을 개발하였다. 제약 프로그래밍은 논리 연산을 기반으로 하므로 복잡한 제약조건을 간단히 모델링 할 수 있고 그래프 이론을 기반으로 하여 스케줄링 문제를 풀기에 적합한 방법이다. 선표 계획 최적화 문제는 매우 복잡한 제약조건으로 구성되어 있고 대상 기간이 매우 길어 규모가 큰 문제이기 때문에 제약 프로그래밍 방법론을 사용하였다.

본 논문에서 제안하는 모델에는 일 별 진수 횟수, 선대 별 진수 가능 요일, 선대 공간, 조립 공정의 일 별 부하, 그리고 key event 시점 관련 항목까지 5개의 주요 제약조건이 포함된다. 공정의 일 별 부하 제약조건을 추가함으로써 선표 계획과 선행공정의 부하분석을 별개의 작업이 아닌 단일 모델로 통합하였으며, 제약 프로그래밍을 통해 최적의

해를 탐색하여 신속한 선표 계획을 가능하게 하였다.

목적함수로는 선표 계획에 가장 중요한 선대-선종 효율 최대화와 계약 납기 준수, 총 2가지를 모델에 반영하였다. 모델의 타당성 검토를 위해 동일한 데이터에 대하여 실적 결과와의 비교 및 제약조건 검증을 진행하였고, 다중 목적함수에 대한 가중치 분석을 통해 pareto optimal을 도출하여 선표 계획에 대한 모델의 범용성을 제시하였다.

제약조건 및 목적함수에 대한 타당성을 확보한 뒤, 다양한 선종의 구성 및 배치 대상 호선 수에 대한 사례 연구를 진행하였다. 사례 연구에서는 탐색이 진행되며 출력되는 해들에 대한 목적함수 값 분석과 호선 공기 증가율 및 선대 연속 배치 비율과 같은 지표에 대한 분석을 수행하였다.

본 논문에서는, 제약 프로그래밍 기반 선표 계획 최적화 모델을 제시하였다. 계획자의 암묵지 및 경험 기반의 선표 계획의 최적해 보장 및 계획 시간에 대한 한계를 극복하는 모델이며, 선행 연구에서 반영되지 못했던 배치 대상 기간에 대한 동적 최적화와 부하 분석을 통합한 모델을 개발하였다. 이를 통해, 선표 계획 최적화 문제에 대한 이론적 기반을 재정립하고, 선표 계획 최적화의 실용화를 달성하였다.

주요어 : 선표 계획(Berth scheduling), 선박 생산 계획(Ship production planning), 제약 만족 문제(Constraint satisfaction problem), 제약 프로그래밍(Constraint programming), 최적화(Optimization), 장기 계획(Long-term plan), 통합 생산 계획(Integrated production planning)

학 번 : 2021-23872

목 차

Abbreviation 1

Nomenclature 2

제 1 장 서 론 4

제 2 장 Unrelated parallel machine scheduling problem 14

제 3 장 방법론 20
제 3.1 절 제약 프로그래밍 20
제 3.2 절 제약 만족 문제(CSP) 22
제 3.3 절 CP 기반 4-queens 문제 최적화 23

제 4 장 문제 정의 28

제 5 장 모델링 32
제 5.1 절 OPL 정의 32
제 5.2 절 CP 기반 선표 수립 최적화 모델 36

제 6 장 CP 모델 검증 45
제 6.1 절 실적 결과와의 비교를 통한 성능 검증 46
제 6.2 절 조립 공정 부하를 고려하는 선표 수립 최적화 모델 검증 49
제 6.3 절 다중 목적함수에 대한 가중치 분석 51

제 7 장 사례 연구 55
제 7.1 절 수립 대상 호선 척 수 별 결과분석 55
제 7.2 절 실험요약 및 기타 지표 분석 61

제 8 장 결론 66

참고 문헌 68

표 목차

Table 2-1 Example of processing time for UPM with 4 jobs and 2 machines 15

Table 2-2 Comparison results of 4 algorithms for UPMSP(Yunusoglu and Topaloglu Yildiz (2022)) 18

Table 4-1 Constraint satisfaction problem of berth planning 31

Table 5-1 OPL for CP model 33

Table 6-1 Ship data for validation of constraint satisfaction 47

Table 6-2 Berth information for validation of constraint satisfaction 47

Table 6-3 Berth-shiptype priority for validation 47

Table 6-4 Result comparison between proposed model and heuristic 48

Table 6-5 Ship data for validation of workload constraint . 49

Table 6-6 Berth priority by shiptype for validation of multi-objective functions 50

Table 6-7 Ship data for validation of weights on multi-objective functions 52

Table 6-8 Result of the objective functions value by weight case 53

Table 7-1 Ship data for KPI analysis 56

Table 7-2 Berth priority by shiptype for KPI analysis 56

Table 7-3 Optimization results for KPI analysis 58

Table 7-4 Solutions in the range of objective function value

trade off 59

Table 7-5 Indices according to size of problem 65

그림 목차

Figure 1-1 Scheduling process of shipbuilding..... 4

Figure 1-2 Example of berth schedule..... 9

Figure 1-3 Berth scheduling process..... 10

Figure 1-4 Objective of proposed model 13

Figure 2-1 Example of PMSP with 4 jobs and 2 machines. 14

Figure 2-2 Gantt chart of result by SPT..... 15

Figure 3-1 Example of states in 4-queens problem;(a): impossible state, (b): possible state 23

Figure 3-2 4-queens problem search tree with simple back tracking(Smith (1995))..... 24

Figure 3-3 4-queens problem search tree with forward checking(Smith (1995))..... 25

Figure 3-4 4-queens problem search tree with maintaining arc consistency 25

Figure 3-5 Comparison of propagation techniques(Barták (1999))..... 26

Figure 4-1 Example of ship batch on a berth 29

Figure 5-1 Definition of interval variable..... 33

Figure 5-2 Mechanism of alternative constraint 34

Figure 5-3 Example of step function($t \rightarrow h$) 35

Figure 5-4 Example of pulse function($t_1 \rightarrow h$, $t_2 \rightarrow 0$) 35

Figure 5-5 Structure for ship variable 37

Figure 5-6 Division of berth A in CP model..... 38

Figure 5-7 Structure of ship variable(a)at the graving berth,(b)at the floating berth 40

Figure 5-8 Process of workstage priority calculation 41

Figure 5-9 (a)Impossible batch of parallel-tandem ship (b)Possible batch of parallel-tandem ship 44

Figure 6-1 Comparison of result for daily workload;(a)CP model without workload constraint, (b) CP model with workload constraint 51

Figure 6-2 Pareto frontier by non-dominant solutions of weight case 54

Figure 7-1 Value of objective functions(small size problem) 58

Figure 7-2 Value of objective functions by solution(big size problem) 60

Figure 7-3 Example of increased leatime occurrence 61

Figure 7-4 Example of berth plan for continuous batch:(a)worse case, (b)better case..... 63

Abbreviation

AIS	Artificial Immune System
CP	Constraint Programming
CSP	Constraint Satisfaction Problem
D/L	Delivery
ETO	Engineering To Order
FJSSP	Flexible Job Shop Scheduling Problem
GA	Genetic Algorithm
IP	Integer Programming
K/L	Keel Laying
L/C	Launching
MIP	Mixed Integer Programming
OPL	Optimization Programming Language
PMSP	Parallel Machine Scheduling Problem
SA	Simulated Annealing
SPT	Shortest Processing Time
S/C	Steel Cutting
UPMSP	Unrelated Parallel Machine Scheduling Problem

Nomenclature

j	Index for ships
i	Index for berths
t	Index for time periods in the scheduling horizon
v	Index for resources
J	Set of all ships
M	Set of all berths
T	Set of all time periods
J_f	Set of fixed ships
Jc	Set of contracted ships
Je	Set of expected ships
M_j	Set of eligible berths for each ship j, $j \in J$, $M_j \subseteq M$
p_{ij}	Processing time of ship(job) j on berth(machine) i
Q	Set of all quays
q	Index for quay
p_{ij}^{Q}	Quay processing time of ship j from berth i
e_{ij}	Efficiency of ship j on berth i
d_j	Contract delivery date of ship j
ap_i	The available number of parallel−tandem ship at berth i
r_{jv}	Resource need of ship j for renewable resource v
r_v	The available amount of resource type v
w	The available amount of daily assembly workload for all berth

p_j	Parent ship of ship j
s_j	Series number of ship j
ps_i	The possibility of semi−tandem of berth i : $st_i \in \{0, 1\}$
pp_i	The possibility of parallel−tandem of berth i : $pt_i \in \{0, 1\}$
X_i	Sets of optional interval variable on berth i
x_{ij}	Optional interval variable indicating whether ship j is assigned to semi−tandem unavailable berth $j, ps_i = 0$
xa_{ij}	Optional interval variable indicating whether ship j is assigned to semi−tandem available berth i before floating, $\forall ps_i = 1$
xb_{ij}	Optional interval variable indicating whether ship j is assigned to semi−tandem available berth i after floating, $\forall ps_i = 1$
y_{ijk}	Optional interval variable indicating preceding process of ship j completed to k% of total workload of preceding process and whether ship j is assigned to berth i
Z	Sets of all interval variable for ship
z_j	Interval variable for ship j that must be present in the solution
Obj_1	Objective function for work stage priority
Obj_2	Objective function for due date tightness
λ_1	Weight for Obj_1
λ_2	Weight for Obj_2
R^I	Increasing rate of berth lead time, I: Increased leadtime
R_i	Continuous batch rate of berth i

제 1 장 서 론

선박 건조 산업은 ETO(Engineering To Order)산업의 대표적인 사례로, 생산 규모가 매우 크기 때문에 효율적인 운영을 위해서는 철저한 생산 계획 수립이 필수적이다. 특히 조선소의 선박 생산 계획은 대상 기간 및 목적에 따라 장기, 중기, 단기계획으로 구분될 수 있으며, 이는 조선 산업의 복잡성과 규모를 반영하는 중요한 관리 과정이다(Figure 1-1).

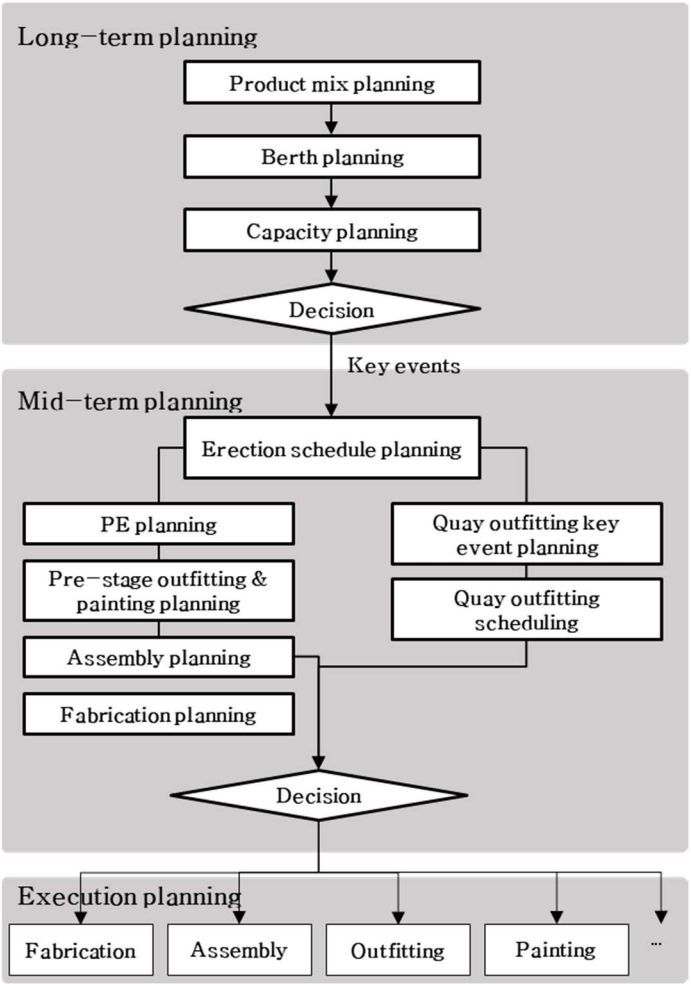


Figure 1-1 Scheduling process of shipbuilding

장기 계획은 대략 3~6년의 범위를 다루며, 조선소의 작업 공간인 선대에서 이루어지는 작업들을 계획하는 선표 계획과 설비의 가용성을 검토하는 설비 계획으로 구분된다. 선표 계획의 중요성은 선대의 효율성이 조선소의 전체 선박 건조 효율성에 큰 영향을 미치기 때문에, 이는 매출에 중대한 영향을 끼치는 단계이다.

중기 계획은 대략 6개월의 시간을 다루며, 선대 공정의 선행 공정 및 후행 공정의 계획을 포함한다. 이 단계에서는 블록 탑재, 선행 탑재, 선행 의장 및 도장, 그리고 조립 및 전처리와 같은 여러 선행 공정을 계획하고, 후행 공정인 안벽 계획을 진행한다.

단기계획은 약 1~2개월에 대한 계획으로 상세한 공정에 대한 실제 실행을 위한 일정 계획이다. 각 단계는 고유의 생산 관리 목표를 가지고 있어 제약조건과 목표 함수는 단계마다 다르다. 대부분의 조선소는 이러한 다양한 요소들을 만족시키는 계획을 수립하기 위해 피드백 기반의 논의를 활용한다.

선박 건조의 중요한 작업 중 하나인 생산 계획의 수립은 수십 년 동안 연구자들과 산업 전문가들이 고민해 왔던 문제이다. 복잡한 이 과정을 최적화하려는 다양한 연구가 수행되어 왔으며, 1990년대부터는 이러한 계획들을 체계적으로 시스템화 하려는 노력이 확대되었다.

김광욱 (1991)은 계산량이 방대하고 반복되는 작업이 많아지는 선박 설계작업의 전산화 필요성을 지적하며, 해당시점의 조선소에서의 전산화 실태를 조사하였다. 윤덕영 and 김근철 (1993)은 선박의 초기 설계와 공정 설계 및 공정 설계와 일정관리 간의 통합을 중심으로 시스템 개발에 대한 개념을 정립하였다. 박주철 et al. (1995)은 이상의 조건속에서 최적의 선표구성을 위한 계산과 분석을 토대로 선표작성을 위한 최적화 알고리즘 및 전산시스템을 개발하였다. 더 나아가 유병철 and 이민우 (1996)는 수작업으로 이루어지던 기본 선표 작성을 정수

계획법을 이용한 지식기반 시스템으로 변환하는데 성공하였다. 이는 단순히 수작업을 시스템화 시키는 것을 넘어 제약 조건을 고려하는 방법론을 도입하여 생산성을 향상시키는 최적화 시스템을 구현하는데 중요한 역할을 하였다. 이러한 연구 결과들은 조선소의 생산 계획 수립에 중요한 역할을 하는 기술적 성과로 간주될 수 있다.

21세기 초반에는 조선소에서의 생산 효율을 극대화하려는 노력이 본격화되었다. 특히, 최적의 생산 계획을 수립하는 방법에 대한 다양한 연구가 활발히 진행되었다. 김기동 et al. (2001)은 본 연구의 알고리즘 기반이 되는 제약 만족 기법을 활용하여 선대 자원 제약을 고려한 탑재 일정 계획 문제에 대한 해결책을 제시하였다. 한편, 배희철 et al. (2007)은 조선소의 소조립 공정에서 makepan을 최소화하고자 skid conveyor system을 도입하였으며, 이를 위해 수학적 모델과 유전 알고리즘을 기반으로 한 연구를 수행하였다.

그러나 알고리즘의 이해가 어려우며, 정확성을 보장하기 어렵고, 너무 광범위한 전체 계획 시스템의 규모 등의 이유로 이전의 최적화 알고리즘을 실제 현장에서 적용하는 데 어려움이 존재했다. 이러한 이유로 휴리스틱 방법에 대한 연구도 병행되어 진행되었다. 이런 연구들은 알고리즘의 이해를 돕고, 실제 현장에서의 적용 가능성을 높이며, 최적화 문제에 대한 새로운 접근 방식을 제시하였다. 여성주 et al. (2008)는 조선 산업에 특화된 의장 작업을 위해 실무 전문가의 경험을 바탕으로 다양한 제약 조건을 직접 고려하는 휴리스틱 알고리즘을 개발하였다. 이와 마찬가지로, 안남수 et al. (2007)은 계획 기간의 길이와 복잡한 제약들이 결합된 안벽 배치 문제를 수리 모델을 이용하여 직접적인 제약 조건을 고려하는 해결책을 제공하는 알고리즘을 개발하였다.

이에 더불어, 단순히 makespan을 최소화하는 연구를 넘어, 주어진

자원을 작업 기간 동안 효율적으로 분산하는 부하 평준화에 초점을 맞춘 연구도 진행되었다. 민상규 et al. (2000)는 각 선박에 할당된 선대 사용 기간과 탑재 순서라는 제약 조건을 만족시키는 동시에, 선대에 배정된 인적 자원의 부하를 균등하게 분산하는 탑재 공정의 탑재 일정을 유전 알고리즘을 이용하여 모델링하였다. 류지성 et al. (2004)은 탑재 네트워크 내에서 부하를 균등하게 분산하기 위한 효율적인 모델을 개발하였다. 이 모델은 제약 만족 기법을 활용하여 최대 부하 감소 알고리즘을 제시하고, 부하 평준화와 관련된 우수한 결과를 도출하였다.

상위 단계의 계획에 초점을 마준 기존의 연구와는 대조적으로, 2010년대 이후로는 하위 단계의 계획에 대한 연구가 활발하게 이루어졌다. 김태호 (2020)는 강재 입출고의 순서가 무작위로 이루어짐으로 인해 발생하는 작업 낭비를 감소시키기 위해 강화학습을 활용하여 강재 적치를 최적화하는 모델을 개발하였다. 또한, 이봉현 and 서윤호 (2020)는 공정 계획과 현장 계획의 일치성 부재를 지적하며, 트랜스포터의 운반 시간과 운반 능력을 제약조건으로 반영하여 조립 일정 계획을 수립하는 휴리스틱 알고리즘을 제안하였다.

이와 같이, 생산계획의 중요성이 점차 부각되면서 최적화된 계획 수립을 위한 연구가 활발히 진행되었으며, 다양한 공정이 포함된 선박 건조 과정에 적용될 수 있는 통합 알고리즘의 필요성이 강조되었다는 것을 확인할 수 있었다. 이러한 점을 감안하여, 본 논문에서는 장기 계획 수립 과정에서 중기 계획의 부하를 고려한 통합 모델을 제안하고자 한다.

선표 계획은 장기계획의 일환으로, 약 3년 동안 건조할 선박의 탑재 기간과 선대를 결정하는 과정을 말하며, 이는 곧 빈 선표에 호선을 배치하는 과정으로 이해할 수 있다. Figure 1-2는 선표의 예시를 보여주며, 선대(세로축)와 기간(가로축)에 따라 각 호선의 탑재 시점이

어떻게 결정되는지 보여준다. 또한, 각 호선에 대한 선대 공정 기간이 결정됨에 따라, 선박 공정의 key event들의 시점이 결정된다. 선박 공정의 key event는 아래와 같다:

- S/C(Steel Cutting) : 첫 번째 철판을 절단하는 과정
- K/L(Keel Laying) : 선박을 구성하는 PE블록 중 첫번째 블록을 선대에 설치하는 과정
- L/C(Launching) : 선대에서의 탑재를 완료하여 처음으로 선체를 물에 띄우는 과정
- D/L(Delivery) : 선주(사)에게 건조가 완료된 선박을 인도하는 과정

호선을 선표에 배치하는 방법은 크게 두 가지로 분류될 수 있다. 첫 번째 방식은 각 호선을 순차적으로 앞에서부터 배치하는 'forward'방식이다. 이 방식은 배치해야 하는 호선들을 정해진 규칙에 따라 대상 기간의 시작 시점부터 차례대로 배치하며, 동시에 미세한 조정 과정을 통해 최선의 배치를 탐색한다. 이 경우, '최선의 배치'는 주어진 배치 순서에 따라 가능한 가장 좋은 배치를 의미한다. 이 방식의 장점은 순서가 정해진 상황에서 최선의 배치를 찾을 가능성이 노다는 것이다. 하지만, 이 방식의 탐색 범위가 넓어 많은 시간이 소요될 수 있고, '최적의 배치'를 항상 보장하기 어렵다는 한계가 있다. 여기서 '최적의 배치'는 배치 순서와 무관하게 전체 상황에서 가장 이상적인 배치를 의미한다.

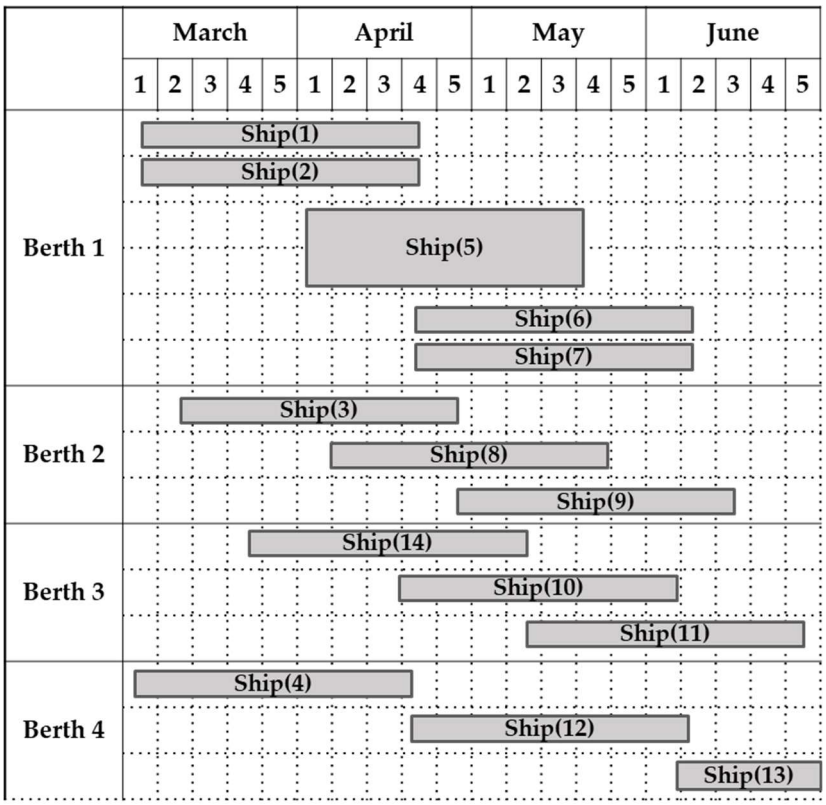


Figure 1-2 Example of berth schedule

두 번째 방법은 선대별로 시점을 일정한 간격으로 분할하거나 표준을 바탕으로 분할하여 빈 배치를 먼저 생성한 뒤, 이를 적절한 호선과 연결하는 방식이다. 이 방식은 최적해를 찾을 가능성이 상대적으로 낮아지지만, 주어진 배치에서 최선의 결과를 빠르게 출력할 수 있다는 장점이 있다.

두 방식 모두 수작업으로 진행되며, 최적에 가까운 결과를 도출하기 위해 추가적인 과정이 포함될 수 있다. 이들 방식을 포함한 전체 선표 구성 과정은 Figure 1-3와 같다. 검토 이벤트에는 신규 호선 수주, 연간 계획 수립 등이 포함되며, 생산 효율을 최대화할 수 있는 최적 선표 구성이 요구된다. 이에 따라, 배치 대상 호선들에 대한 초기 선표 구성 단계가 진행된다. 초기 선표 구성을 위해 계획자는 배치 대상 호선의 탑재 효율이 최대가 되는 선대를 선택하며, 이때 공간적인

요소와 납기일 등의 제약조건들을 고려한다. 초기 선표가 구성된 후에는 하위 단계의 다양한 부하 평가 과정이 이어진다. 이 과정은 각 단계에서 수립되는 계획의 제약조건과 목적함수가 다르기 때문에 필요하며, 선표 구성 시점에 고려하지 못했던 요소들을 고려할 수 있다. 부하 평가 항목으로는 적치장 부하, 조립 부하, 탑재 부하 등이 있으며, 이들 항목이 최소 기준을 만족하지 못할 경우에는 초기 선표에서의 호선별 일정을 조정하는 과정이 반복된다. 이 반복 과정을 통해 부하 기준을 만족하는 선표가 구성되면, 해당 과정을 종료하고 최종 선표를 결정하게 된다.

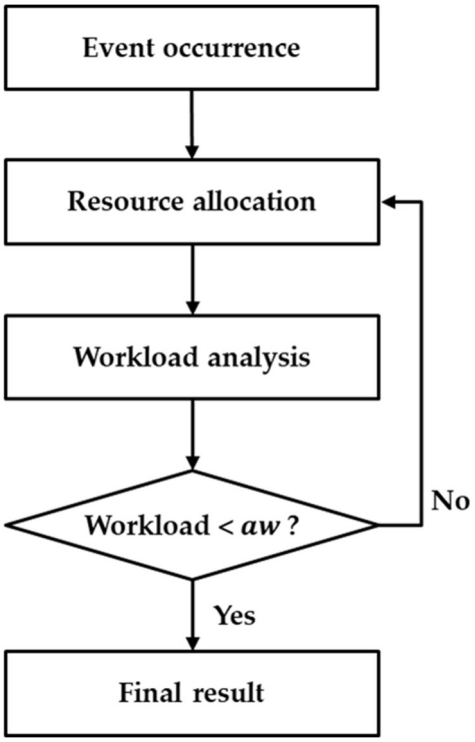


Figure 1-3 Berth scheduling process

선표의 중요성에 비해 기존 선표 수립 방식에는 몇 가지 개선이 필요하다. 우선, 현행 선표 수립 과정은 주로 계획자의 경험과 암묵지에 의존하여 수작업으로 이루어지는 경우가 많다. 이는 일관성 있는 선표

수립을 어렵게 만드는 요소이다. 생산 효율을 고려하는 규칙이 존재하지만, 선표에 호선을 배치하는 순서에 따라 결과가 달라지므로 계획자의 주관적인 판단이 불가피하게 개입하게 된다.

두 번째로, 제약조건과 목적함수의 복잡성 때문에 최적해 탐색 범위가 굉장히 넓어지며, 이로 인해 선표 수립 시간이 증가하는 문제가 있다. 제약 조건을 준수하는 다양한 경우의 수 중에서 목적함수를 최대화 또는 최소화하는 해를 휴리스틱 방법으로 찾는 과정에서는 시간적인 제한이 있고, 이로 인해 최종 수립된 선표가 최적해인지를 보장하기 어렵다.

마지막으로, 기존의 선표 수립 과정은 호선 배치와 부하 분석 과정이 분리되어 있다는 점이다(Figure 1-3). 이 두 과정이 별개로 진행되므로, 담당자도 각각 다르고, 반복적인 피드백 과정에서 소요 시간이 증가하게 된다. 따라서 이런 문제점들을 개선하고자, 본 논문에서는 계획자의 암묵지와 경험에 의존하지 않는 효율적인 선표 수립 방안을 제시하고자 한다.

이러한 개선사항들이 발견되었고 세계적인 선박의 수주 물량이 증가함에 따라 선표 수립 관련 연구의 필요성도 증가하였다. 허영주 et al. (1998)은 선표 수립 과정을 세 계층으로 나누어 순차적으로 최적화하는 시스템을 제시하였다. 각 계층별 최적화 방법으로 GA(Genetic Algorithm), convex polygon, 그리고 tabu search 등 3가지 알고리즘을 사용하여 효과적인 결과를 제시하였다. 그러나 호선 배치를 위한 선대 별 시점을 일정한 간격으로 나누는 방법을 채택하였다는 점에서 완벽한 최적화와는 거리가 있으며 선행 공정의 부하를 고려하는 과정과는 별개로 호선 배치만을 위한 연구라는 점에서 한계를 갖는 연구였다. Ju and Woo (2021)는 선표 수립과정에 중기 조립 계획의 부하를 고려하는 연구를 수행하였다. 호선 배치와 부하

평가를 2phase로 분리하지만 하나의 모델로 통합했다는 점에서 의미를 갖는 연구이지만, 초기 호선 배치가 완료된 이후에 진행되는 과정이라는 점에서 최적화에 한계를 갖고 있다. Kwak et al. (2022)에서는 제약 프로그래밍 기반의 선표 수립 최적화 모델을 개발하였다. 실제 산업에서의 결과 비교에 따르면 최적화된 값에 가까운 목적함수 결과를 출력하는 모델을 개발하였지만, 표준 batch table을 기반으로 호선을 빈 배치에 호선을 매칭하는 방식을 사용하였기 때문에 엄밀한 정의의 최적화와는 다른 결과를 출력한다고 할 수 있다. 선표 수립 최적화와 관련된 연구들을 조사해본 결과, 복잡한 제약조건과 방대한 탐색 범위라는 한계점이 존재하기 때문에 많은 연구가 존재하지 않았고, 일부 제약조건에 휴리스틱 방식을 적용하는 연구들이 존재함을 확인할 수 있었다.

본 연구에서는 기존의 선표 수립 과정에서 개선이 필요했던 사항들을 보완한 최적화 모델을 구현한다(Figure 1-4). 기존에 분리 및 반복 진행(Figure 1-4-④)되었던 호선 배치(Figure 1-4-②) 및 부하 분석(Figure 1-4-③) 과정이 모델 구현을 통해 하나의 과정(Figure 1-4-ⓒ)으로 통합 진행되며 이는 수립 시간 및 최적화 측면에서 개선된 결과를 출력한다.

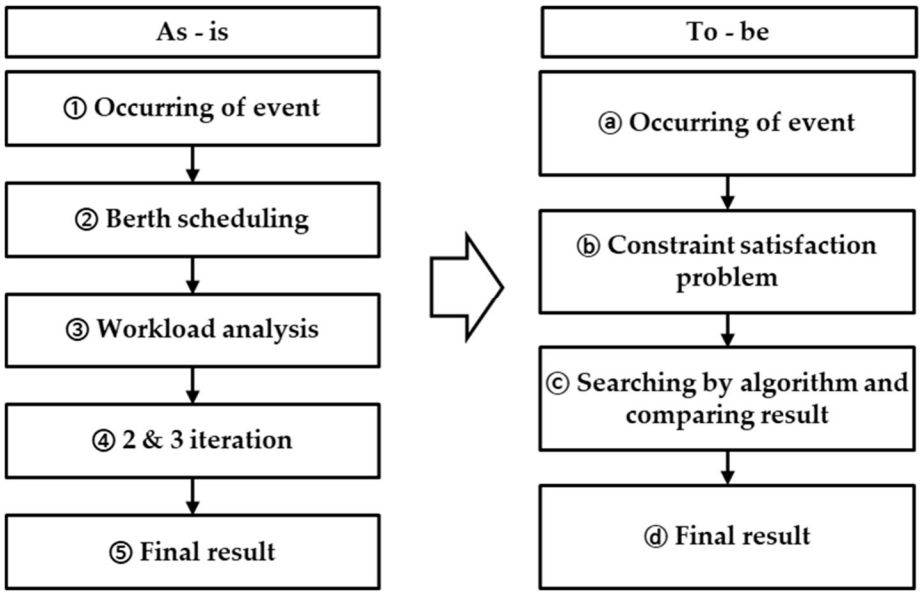


Figure 1-4 Objective of proposed model

본 논문은 8장으로 구성된다. 제 2 장에서, 선표 수립 최적화 문제의 이론적 근간이 되는 UPMSP(Unrelated Parallel Machine Scheduling Problem)에 대한 정의와 기존의 선행 연구 분석을 통해 제약 프로그래밍 적용에 대한 논의를 진행한다. 제 3 장에서는 제약 프로그래밍의 정의 및 원리를 소개하며 제 4 장에서는 선표 수립 최적화 문제를 제약 만족 문제로 변환하는 과정을 설명한다. 제 5 장에서는 정의된 문제에 대해 제약 프로그래밍 기반 모델 구현과정을 설명하며, 이를 위해 사용되는 OPL(Optimization Programming Language)에 대한 설명을 제공한다. 제 6 장에서, 구현된 모델의 성능 검증을 진행하며, 제 7 장에서는 다양한 데이터를 기반으로 출력된 결과에 대한 분석을 진행하여 본 연구의 주요 성과를 설명한다. 마지막으로, 제 8 장 결론에서는 본 연구의 의의와 한계점을 분석하여 향후 연구에 대한 가능성 제시한다.

제 2 장 Unrelated parallel machine scheduling problem

UPMSP(Unrelated Parallel Machine Scheduling Problem)는 PMSP(Parallel Machine Scheduling Problem)에서 파생된 문제이다. PMSP는 Figure 2-1와 같이 도식화할 수 있으며, 주어진 job들에 대하여 동일한 작업을 수행하는 여러 대의 기계가 존재할 때, job들에 대한 작업 순서 및 기계를 결정하는 일정 수립 문제이다. 일반적인 목적함수는 모든 job을 완료하는 공정 시간 최소화이지만 문제에 따라 추가되거나 변경될 수도 있다.

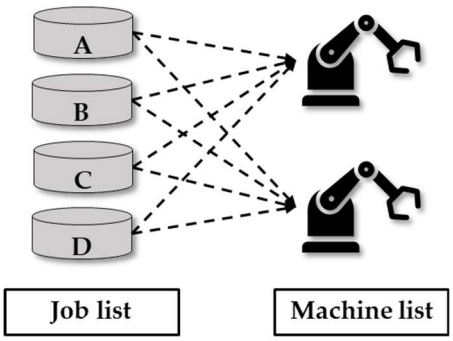


Figure 2-1 Example of PMSP with 4 jobs and 2 machines

PMSP는 machine별 job의 작업시간이 어떻게 이루어지는지에 따라 세 가지로 분류한다. machine에 상관없이 특정 job에 대해 모두 동일한 작업시간을 갖는다면 identical machine, machine별로 동일한 비율의 속도 차이를 갖는다면 uniform machine, 그리고 기계별로 job의 작업시간이 모두 다르다면 unrelated machine으로 구분한다. Piersma and van Dijk (1996)는 p_{ij}가 machine i에서 job j의 작업시간을 의미할 때 각 문제를 다음과 같이 구분하였다:

- Identical machines: $p_{ij} = p_{1j}$ for all i and j;
- Uniform machines: $p_{ij} = \frac{p_j}{v_i}$ for all i and j , where v_i is the speed of machine i, and p_j is the processing time at unit speed;
- Unrelated machines: p_{ij} arbitrary for all i and j.

UPM(Unrelated Parallel Machines)의 정의에 따르면 Figure 2-1문제에 대한 작업시간은 Table 2-1과 같이 주어질 수 있다. UPMSP의 최적해를 도출하기 위한 알고리즘은 문제의 복잡도에 따라 다양하며 Table 2-1에 대해 휴리스틱 중 하나인 SPT(Shortest Processing Time)를 적용하면 Figure 2-2과 같은 결과를 얻을 수 있다. 해당 문제의 최적해는 4이고 SPT를 사용하여 도출한 결과도 4이기 때문에 SPT를 통해 최적화가 완료된 것을 확인할 수 있다.

Table 2-1 Example of processing time for UPM with 4 jobs and 2 machines

Machine	A	B	C	D
Machine 1	3	4	1	5
Machine 2	5	2	3	1

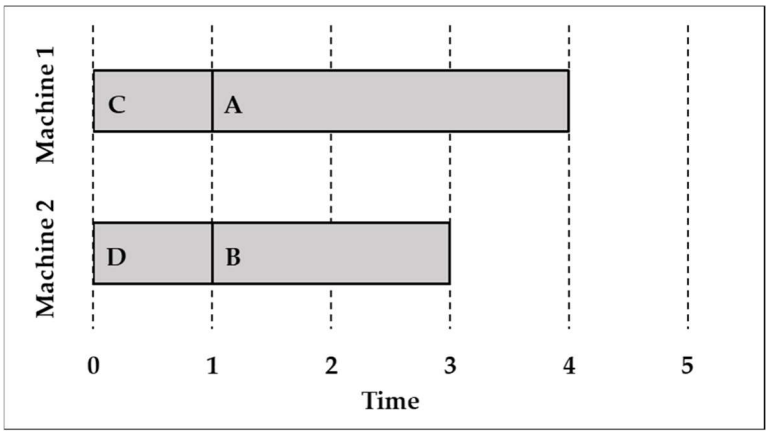


Figure 2-2 Gantt chart of result by SPT

일반적으로는, 실제 산업에서의 문제를 해결하기 위해 UPMSP의 기본 문제에서 제약조건들을 추가적으로 정의한다. 재사용가능한 자원의 유무, 작업 간의 준비 시간, 작업의 선후행 관계, 작업별 가능 기계 그리고 작업 시작 가능 일자 등의 추가 제약조건들이 문제의 특성에 따라 선택적으로 부여된다. 또한, 목적함수로 makespan 최소화 또는 지연 최소화 등이 정의될 수 있으며 두 개 이상의 목적함수가 정의된 문제는 많은 지역 최적점을 갖기 때문에 최적해 탐색 영역이 복잡하다. 따라서 복잡도가 증가할수록 SPT와 같은 특정 휴리스틱으로 최적해를 탐색하는 것에 한계가 존재하며, 이를 극복하기 위해 GA(Genetic Algorithm) 및 SA(Simulated Annealing)과 같은 메타 휴리스틱이나 제약 프로그래밍 및 혼합 정수 최적화와 같은 tree search 기반의 방법을 사용한다.

UPMSP는 조선소 뿐만 아니라 flexible job shop의 형태로 운영되는 많은 산업에서 중요한 문제이며, 다양한 연구가 이루어져 왔다. Chen (2005)은 휴리스틱 알고리즘을 사용하여 재사용 가능 자원 및 준비시간을 갖는 문제에 대해 makespan최소화를 위한 연구를 진행하였다. 이 외에도 간단한 제약조건에 대해 우수한 성능을 보이는 휴리스틱 기반 연구들이 존재했다(Chen and Wu (2006); Villa et al. (2018)). Edis and Oguz (2012)는 유사한 제약조건을 갖는 UPMSP에 대해 IP모델을 해결책으로 제시하였다. 좀 더 복잡한 제약조건을 갖는 문제로 재사용 가능 자원, 준비 시간, 선행 공정 관계, 작업 시작 가능 일자로 구성된 문제를 해결하기 위해 Wang et al. (2013)은 MIP 모델을 제시하였다. 시간이 흐름에 따라 문제의 사이즈 및 복잡도에 따라 전통 최적화 방법이 아닌 메타 휴리스틱 알고리즘을 적용하는 연구들도 많이

등장하였다(Costa et al. (2013); Cappadonna et al. (2013); Afzalirad and Rezaeian (2016)).

Yunusoglu and Topaloglu Yildiz (2022)는 기본 UPMSP에 복잡한 제약조건(재사용 가능한 자원, 작업 간의 준비 시간, 작업의 선행 관계, 작업 가능 기계, 작업 시작 가능 일자)이 반영된 문제를 풀기위해 4가지 알고리즘(IP, CP, GA, AIS)으로 구현하고 결과를 비교하였다. 문제의 크기에 따라 나누어 분석을 진행하였으며, Table 2-2는작은 크기의 문제에 대한 결과 값을 나타낸 것이다.

Table 2−2 Comparison results of 4 algorithms for UPMSP(Yunusoglu and Topaloglu Yildiz (2022))

	Problem size	Problem complexity		IP		CP		GA		AIS	
No	|N| × |M| × |R|	NV	NC	Best	Time (s)	Best	Time (s)	Best	Time (s)	Best	Time (s)
1	3 × 2 × 1	14	145	39	1.64	39	0.02	39	6.39	39	5.65
2	3 × 2 × 2	14	146	56	1.98	56	0.02	56	6.36	56	5.94
3	4 × 2 × 1	17	179	58	3.59	58	0.01	58	8.44	58	7.39
4	4 × 2 × 2	17	180	50	28.88	50	0.03	50	7.62	50	7.02
5	4 × 3 × 1	23	261	50	359.24	50	0.01	50	8.81	50	8.67
6	4 × 3 × 2	23	265	60	149.56	60	0.03	60	8.80	60	8.64
7	5 × 2 × 1	20	214	67	442.53	67	0.05	67	9.53	57	9.51
8	5 × 2 × 2	20	214	61	233.94	61	0.05	61	9.55	61	9.59
9	5 × 3 × 1	27	311	58	3,600	57	0.06	57	10.42	57	10.52
10	5 × 3 × 2	27	313	66	3,600	61	0.04	61	40.77	61	10.74
11	6 × 2 × 1	23	247	66	286.39	66	0.04	66	11.65	66	11.26
12	6 × 2 × 2	23	248	64	1,001.17	64	0.08	64	10.81	64	11.52
13	6 × 3 × 1	31	360	72	3,600	69	0.07	69	12.53	69	12.86
14	6 × 3 × 2	31	365	78	3,600	72	0.05	72	12.69	72	13.12

Abbreviations : |N|, |M|, |R|: # jobs, # machines, # resource types; NV: number of variables; NC: number of constraints; GA: genetic algorithm; AIS: artificial immune system; optimum results are shown in bold

비교결과, 작은 크기의 문제들임에도 불구하고 IP는 일부 문제에 대해 최적해를 탐색하지 못했고, CP, GA, AIS는 모두 최적해를 탐색했지만 CP의 탐색속도 성능이 가장 우수한 것을 확인할 수 있었다.

본 논문의 대상 연구인 선표 수립 최적화 문제는 UPMSP로 정의할 수 있다. 재사용가능한 자원(선대 공간), 선후행 관계(시리즈 호선), 작업별 가능 기계(선종별 건조 가능 선대), 작업 시작 가능 일자(계약일)와 같이 일반적인 제약조건을 정의할 수 있으며 실제 작업 과정을 고려하기 위해 복잡한 몇 가지 제약조건들이 정의된다. 또한 목적함수에는 지연 최소화(납기 준수)와 양질의 선표를 수립하기 위한 추가적인 요소들이 반영된다. 따라서 선표 수립 최적화 문제는 UPMSP에 대응되며, Table 2-2와 같이 검증된 CP의 성능을 바탕으로 해결하고자 한다.

제 3 장 방법론

제 3.1 절 제약 프로그래밍

조합 최적화 문제를 해결하는 방법 중 하나인 CP는 변수들 사이에 주어진 제약조건을 만족하는 해를 찾는 것을 목표로 한다. 이는 CSP로 표현될 수 있는데, CSP는 변수, 도메인, 제약조건으로 구성된다. 변수들은 문제의 해를 표현하고, 도메인은 각 변수가 가질 수 있는 값의 범위를 나타내며, 제약조건은 변수 간의 관계를 정의하고 가능한 해의 범위를 제한한다.

CSP를 해결하기 위한 CP의 전략들 중에는 back tracking, forward checking, 그리고 maintaining arc consistency 등이 있다. 이러한 전략들은 모두 tree search를 기반으로 하며 차이점은 탐색 과정에서의 branch선택 방법에 있다.

CP는 논리 프로그래밍을 기반으로 하므로 복잡한 제약 조건을 쉽게 모델링할 수 있다. 또한 그래프 이론을 기반으로 하여 스케줄링 문제를 풀기에 적합한 방법이다. 따라서 복잡한 제약 조건과 큰 규모의 스케줄링 문제를 다루는 데 CP는 유용할 수 있다. Novas (2019)는 CP기반의 모델을 사용하여 FJSSP(Flexible Job Shop Scheduling Problem)을 푸는 방법을 제시했다. Ham and Cakici (2016)은 FJSSP에 batch processing이 접목된 문제를 풀기 위해 MIP와 CP를 모두 제시하였다. Goel et al. (2015)도 CP기반의 모델을 구현하여 LNG 운반선의 일정을 관리하는 문제를 해결하였다. Wang et al. (2015)은 수술실 일정 문제를 해결하기 위해 MIP모델과 CP모델을 모두 구현하여 결과를 비교하며 각 모델의 장단점을 설명하였다. CP모델은 문제의

크기에 비해 더욱 복잡한 제약조건을 반영하는데 우수한 성능을 보였다.

CP는 단일 적용 이외에도 복잡한 문제를 해결하기 위해 다른 알고리즘과 결합하여 최적해를 탐색하는 데 사용된다. Qin et al. (2020)는 MIP와 CP를 결합하여 항만에서 단일 선박의 컨테이너 일정관리 문제를 해결하는 모델을 제시하였다. 또한, Backer et al. (2000)은 CP와 메타 휴리스틱 방법을 통합하여 차량 경로 탐색 문제에 적용하였다.

제 3.2 절 제약 만족 문제(CSP)

CSP는 일반적으로 (X, D, C)로 정의할 수 있으며 각 구성요소는 다음과 같다.

- 변수: $X = \{x_1, \ldots, x_n\}$
- 도메인: $D = \{d_1, \ldots, d_n\}$
- 제약조건: $C = \{c_1, \ldots, c_n\}$

변수는 CSP에서 결정되어야 하는 요소이며, 도메인은 변수의 가능 해 범위를 나타낸다. 제약조건은 각 변수의 가능 범위를 제한하거나 변수 간의 관계를 규명하는 역할을 하며 정의 방법에 따라 모델의 탐색 성능에 큰 영향을 주는 요소이다.

대표적인 CSP의 예시로, 4-quenns problem에 대해 변수, 제약조건, 도메인은 다음과 같다:

- 변수: $X = \{x_1, x_2, x_3, x_4\}$
- 도메인: $D = \{1, 2, 3, 4\}$
- 제약조건: $C = \{c_1, c_2\}$
  - c_1: $|x_i - x_j| \neq |i - j|, \quad \forall i, j \in \{1,2,3,4\}$
  - c_2: $x_i \neq x_j, \quad \forall i, j \in \{1,2,3,4\}$

각 queen은 행 별로 1, 2, 3, 4열 중 하나를 선택하며 모두 다른 행 및 열에 위치해야 한다. 동시에 임의의 두 queen은 대각선 방향에 존재할 수 없을 때, 모든 queen의 위치를 결정하는 문제이다.

제 3.3 절 CP 기반 4-queens 문제 최적화

4-queens 문제는 대표적인 CSP로, 4개의 행에 queen이 각각 한 개씩 배치가 되어야 하는 문제이다. 다만, 어떤 두 queen도 가로, 세로, 대각선 방향의 일직선 상에 놓일 수 없기 때문에 Figure 3-1(a)는 대각선 방향에 2개의 queen이 놓여있기 때문에 불가능한 state이고, Figure 3-1(b)는 가능한 state이다. 따라서 4개의 queen이 서로 가능한 state가 되도록 배치하는 문제이며 이는 back tracking, forward checking, maintaining arc consistency의 알고리즘으로 각각 해결할 수 있다.

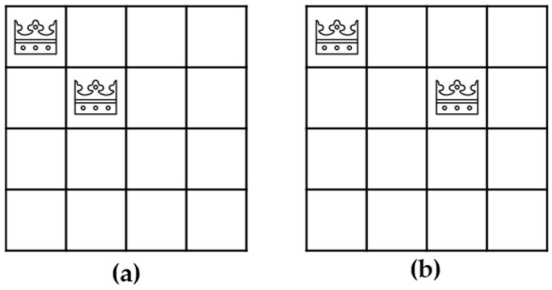


Figure 3-1 Example of states in 4-queens problem;(a): impossible state, (b): possible state

문제를 풀기 위한 기본 과정은 동일하며 각 행 별로 queen의 위치를 결정해 나가는 방법으로 진행된다. 우선, back tracking 방법은 첫 행의 임의의 위치에 queen을 배치한다. 본 논문에서는 가능한한 왼쪽에서 시작하여 배치를 진행한다. 그 다음 행으로 이동하면, 각 열에 퀸을 배치해 보고, 가능한 위치와 불가능한 위치를 구분하여, 불가능한 위치는 제거한다.

두 번째 행까지 가능한 배치가 확인되면, 세 번째 행에 대해 동일한 과정을 반복한다. 그러나 만약 네 번째 행에서 모든 배치가

불가능하다면, 가능한 가장 가까운 이전 단계로 돌아가 다른 경우를 탐색한다. 이렇게 모든 가능성을 조사하면서 최종적으로 해답을 찾는 것이 back tracking 알고리즘의 기본 구조이다. Smith (1995)는 이 과정을 Figure 3-2의 search tree를 통해 시작적으로 작 표현하였다.

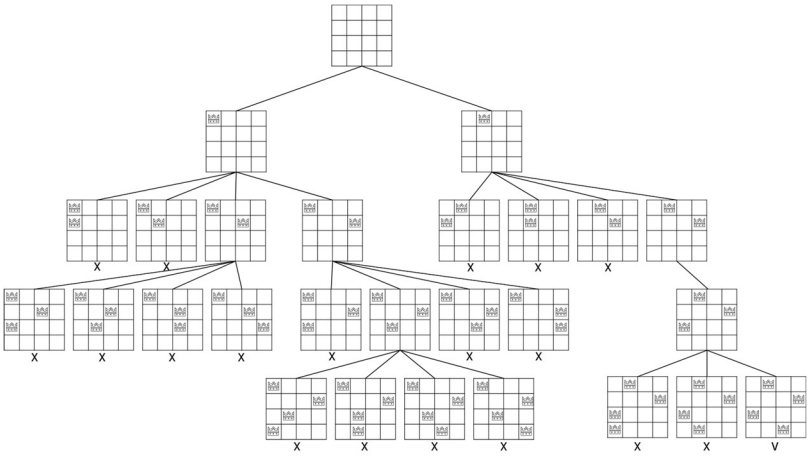


Figure 3-2 4-queens problem search tree with simple back tracking(Smith (1995))

Back tracking이 변수를 결정하는 순간에만 해의 가능성을 검토하는 반면, forward checking은 변수가 결정될 때 미래의 변수에 대해 불가능한 해를 함께 제거하는 방식을 채택한다. 이를 통해 탐색 공간을 더욱 효과적으로 줄여 나간다.

Figure 3-3에서 볼 수 있듯이, 각 행에 퀸의 위치가 결정될 때마다 그에 따라 다른 행의 퀸 위치에서 불가능한 경우를 제거해 나가는 과정을 진행한다. 이러한 방식은 전체 탐색 공간을 신속하게 줄여 나가는데 기여한다.

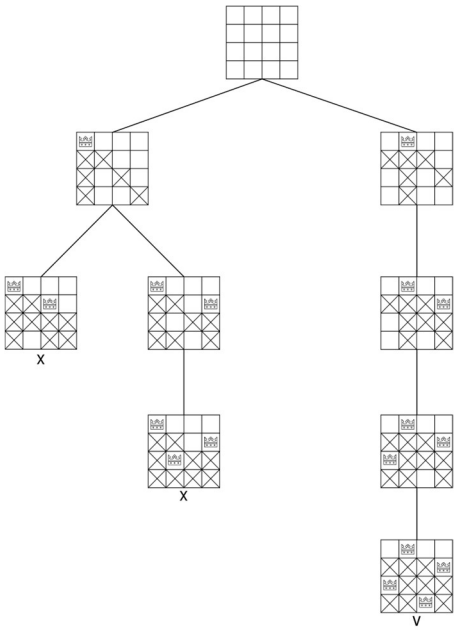


Figure 3-3 4-queens problem search tree with forward checking(Smith (1995))

마지막으로, maintaining arc consistency는 full look ahead라고도 부르며 forward checking에 비해 더욱 신속하게 해 탐색 공간을 줄여 나가는 알고리즘이다. 단순히 결정 대상 변수의 해에 따른 미결정 변수의 불가능한 해 집합만을 제거하는 것이 아니라, 미결정 변수 간 상충되는 제약조건에 대해 불가능한 해 집합까지도 제거하는 방법이다.

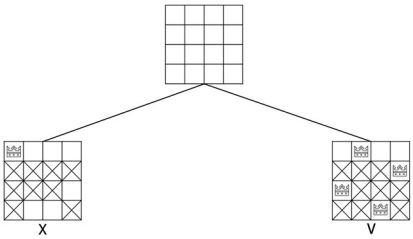


Figure 3-4 4-queens problem search tree with maintaining arc consistency

각 알고리즘은 Figure 3-5와 같이 비교할 수 있으며 가능한 모든 해를 탐색하는 것이 목표인지, 최적해를 탐색하는 것이 목표인지에 따라 알고리즘 선택이 결정될 수 있다.

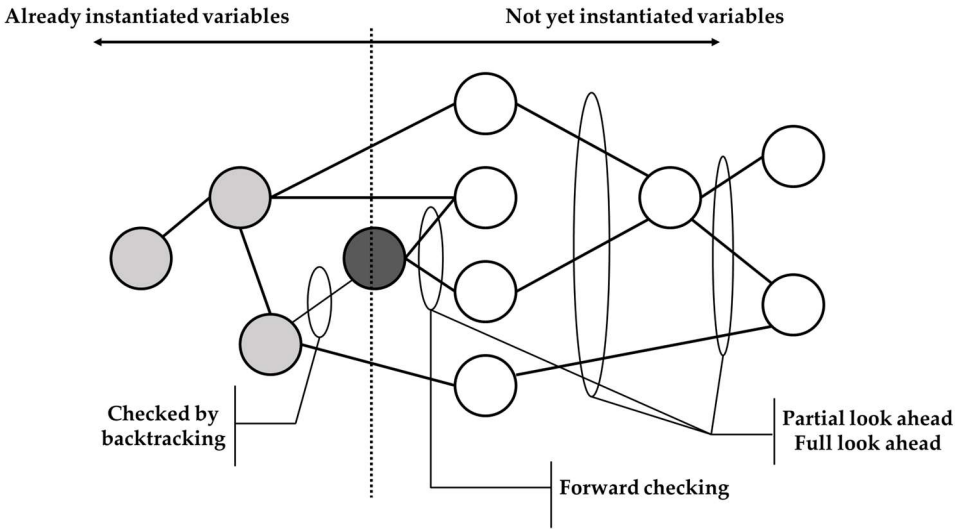


Figure 3-5 Comparison of propagation techniques(Barták (1999))

제약 프로그래밍은 3개의 알고리즘을 적절히 사용하여 constraint propagation을 진행하며 탐색 효율을 높이기 위해 변수의 선택에 대한 알고리즘으로 휴리스틱을 적용한다. 앞에 예시 문제였던 4-queens problem의 search tree를 구성할 때, 왼쪽부터 배치를 해본다는 휴리스틱을 적용하여 풀었지만 일반적인 해결방법으로 다양한 휴리스틱이 존재한다. 대표적인 2가지 휴리스틱은 variable ordering과 value ordering이며 variable ordering은 결정 변수를 선택하는 기준을, value ordering은 변수의 값을 선택하는 기준을 선택하는 알고리즘이다.

제약 전파를 진행하는 과정에서 4-queens problem 예시에서는 변수 선택을 우측 하향으로 진행하였지만 variable ordering 방법에는 dynamic ordering, static ordering으로 분류되는 여러가지 알고리즘이 존재한다. Dynamic ordering은 결정 시점마다 알고리즘이 포함하는 여러 정보를 바탕으로 탐색 대상 변수를 결정하는 방법이다. 다양한 정보를 기반으로 탐색 효율은 높일 수 있지만, 알고리즘의 복잡도에 따라 속도 저하를 일으킬 수 있다는 특징을 갖는다. Static ordering은 정해진 규칙에 따라 일관성 있는 탐색 대상 변수 선택을 진행하는

방법으로 탐색 속도는 빠르지만 효율이 떨어질 수 있다는 단점을 갖는다. 때문에 적절히 혼합된 알고리즘을 사용하여 탐색 알고리즘을 구현하며 목적에 알맞은 알고리즘을 사용하는 것이 중요하다.

Value ordering은 결정 대상 변수의 가능 해 집합 중 어떤 값에 대해 먼저 탐색을 진행할 지 결정하는 과정이며, 일반적으로 가능한 모든 해를 탐색하는 모델의 관점에서는 중요성이 떨어지기 때문에 random 선택으로 진행한다. 그러나 최적해를 탐색하는 모델의 관점에서는 value의 선택에 따라 제거할 수 있는 탐색 범위가 증가하기 때문에 매우 중요하며 smallest value ordering이나 biggest value ordering과 같이 search failure를 발생시킬 가능성이 높은 value 부터 탐색을 진행하는 알고리즘을 사용한다.

CP는 이와 같은 constraint propagation 및 ordering heuristics를 적절히 사용하여 최적해 또는 가능해를 탐색하는 과정이다. 최근에는, variable 또는 value 선택 과정에서 일반 heuristic이 아닌 machine learning이나 meta heuristic과 같은 알고리즘을 사용하여 모델의 성능을 높이는 연구가 활발히 진행되고 있다. 본 연구에서는 모델 자체의 성능을 결정짓기 보다는 주어진 선표 수립 최적화 문제의 제약 프로그래밍 모델 구현에 집중하며 solver로는 IBM사의 ILOG CPLEX CP Optimizer 패키지를 사용한다.

제 4 장 문제 정의

본 연구에서 해결하고자 하는 선표 계획은 호선에 대한 key event 및 선대를 결정하는 작업이며, 제약조건을 만족하며 목적함수를 최대화 또는 최소화하는 과정이다. 선표 수립 과정에서 각 배치 대상 호선은 3가지 종류로 이루어진다. 고정호선, 계약호선 그리고 미정선으로 구성되어 있다. 고정호선은 선표 수립 시점에 이미 선대 및 탑재 일정이 정해져 있는 호선, 계약호선은 계약이 수립되어 실제 선대 및 탑재 기간을 정해야 하는 호선, 그리고 미정선은 현재 계약되어 있지는 않지만 계약될 가능성이 높거나 조선소 수익을 극대화할 수 있는 선종들로 구성되는 호선이다. 호선의 종류에 따라 제약조건 및 목적함수에 미세한 차이가 있다. 제약조건은 크게 4가지로 이루어진다. 우선, tandem공법과 관련된 제약조건이 있다. Tandem공법은 하나의 선대에서 2 척 이상의 선박을 동시에 건조하는 공법을 의미하며, 선대의 공간 활용을 극대화하여 회전율을 높이기 위해 도입되었다. 방향에 따라 parallel-tandem과 semi-tandem으로 나눌 수 있다. Figure 4-1은 임의의 선대에 대한 배치 예시를 도식화 한 것으로, Figure 4-1①과 Figure 4-1②와 같은 배치 방법은 semi-tandem으로, 길이 방향으로 2척을 동시에 건조하는 공법이다. 일반적으로 한 척의 전체 건조 호선(Figure 4-1②)과 나머지 한 척의 일부 건조 호선(Figure 4-1①)으로 이루어지며, Figure 4-1②가 L/C을 진행하는 경우, Figure 4-1①은 floating을 진행하게 된다. Figure 4-1②와 Figure 4-1③과 같은 배치 방법은 parallel-tandem으로, 폭 방향으로 2척을 동시에 건조하는 공법이다.

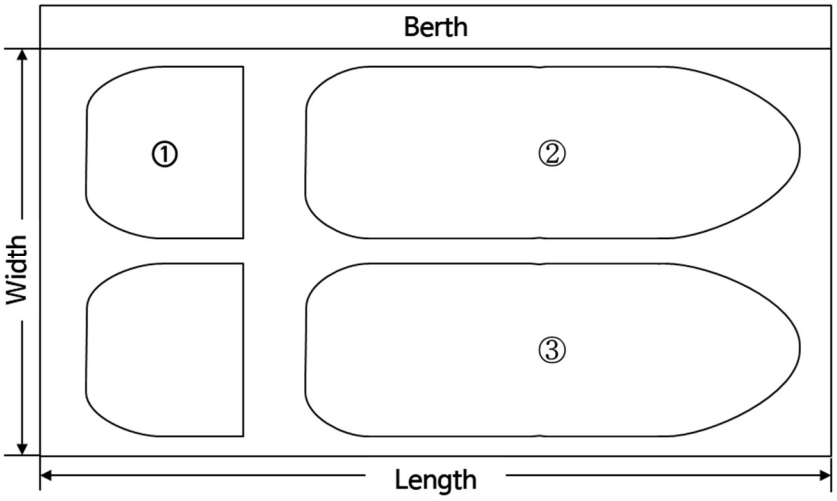


Figure 4-1 Example of ship batch on a berth

동일 선대 내에서 semi-tandem이나 parallel-tandem공법이 이루어질 경우, 동일 시기에 건조되는 호선 간의 key event일자가 교차되지 않아야 한다. 두 번째, 선대별로 진수 가능 요일이 지정 되어있고 이는 선대의 종류에 따라 구분된다. 다만, 고정 호선의 경우, 예외적으로 진수 불가능 요일에 지정되어 있을 수 있으며, tandem 공법을 위해 진수 일자를 조정하는 과정에서 계약 호선 및 미정선은 해당 고정 호선의 진수 요일을 따를 수 있다. 세 번째 제약조건으로, 시리즈 호선일 경우 계약 납기일 순서 또는 시리즈 번호 순서에 따라서 탑재 계획이 수립되어야 한다. 마지막으로, 모든 호선에 대해 선대 결정 과정에서 각 선대의 공간을 초과하는 선표를 수립할 수 없다. 제약조건은 본 연구에서 제시하는 모델이 해를 탐색하는 과정에서 반드시 만족해야 하며, 제약조건을 단 하나라도 어기면 해가 될 수 없다. 제약조건을 모두 만족하는 선표 해 중에서도 평가 기준에 따라 우선순위가 결정되며 목적함수가 그 역할을 담당한다. 본 연구의 목적함수로는 2가지가 존재한다. 첫 번째, 선대 별로 선종에 대한 작업 효율에 차이가 존재하며 해당 효율을 최대화하는 선표를 선택해야 한다. 두 번째, 계약

호선의 경우, 계약 납기일이 존재하기 때문에 납기일과 작업 종료일의 차이가 최소가 되어야 한다. 이와 같이 호선 배치를 위한 제약 만족 문제를 정의할 수 있으며 본 문제의 최적해는 최종 선표로서 출력한다. 추가적으로, 본 논문에서는 호선 배치와 동시에 선행 공정 중 조립 공정의 부하를 고려하는 문제를 정의하기 위해 추가적인 변수 및 제약조건이 정의되었다. 우선, 호선 별 선행공정이 진행되는 기간에 대한 변수를 정의하였으며, 해당 변수의 구간에 따라 일별 부하를 누적할 수 있다. 누적된 일별 부하의 상한선을 지정하기 위해 제약조건을 추가하였으며, 이를 통해 선표 수립 및 선행공정의 부하를 동시에 고려할 수 있는 제약 만족 문제를 정의할 수 있다. 이를 요약하면 Table 4-1과 같이 정리할 수 있다.

Table 4-1 Constraint satisfaction problem of berth planning

Category	Content	Description
Variable	Fixed ship	Ship fixed when berth scheduling
	Contracted ship	Ship contracted and actual target for berth scheduling
	Expected ship	Ship expected to contract
Constraint	Key event	The dates of floating or L/C coincidence when parallel-tandem or semi-tandem at the same berth
	L/C day	L/C possibility on the exact day(s) of week
	L/C capacity	Number of possible L/Cs per a day
	Series ship	Delivery order by rules about series ships
	Spatial	Length and width capacity of berths
	Workload	Capacity for daily workload of assembly
Objective function	Work stage priority	Value of cost effectiveness between shiptype and each berth
	Due date tightness	Difference between contracted delivery date and scheduled delivery date

제 5 장 모델링

제 5.1 절 OPL 정의

본 연구의 모델 구현을 위해 IBM의 ILOG CP optimizer 패키지를 사용하였다. 해당 패키지에서는 OPL(Optimization Programming Language)기반의 함수들을 제공하며 복잡한 scheduling을 위한 제약조건을 용이하게 구현할 수 있다(Laborie et al. (2018)). 본 연구에서 모델 구현을 위해 사용되는 제약조건 및 함수는 Table 5-1와 같다.

본 연구의 결정 변수는 Table 5-1①과 같은 global variable을 사용하며 이는 scheduling을 목적으로 하는 모델의 대표적인 변수이며, 계획대상의 착수 시점과 마감 시점의 가능 범위와 기간 등을 기반으로 정의한다(Laborie and Rogerie (2008)). Figure 5-1는 Table 5-1①을 도식화한 것이며 α는 release date, β는 due date으로서, 정의되는 변수의 가능 범위를 나타낸다. 또한, s와 e는 각각 모델에서 결정되어야 할 대상으로, 변수의 착수일과 마감일을 나타낸다. 마지막으로 l은 계획대상의 작업 기간으로 본 모델에서는 작업 기간의 최솟값 및 최댓값을 반영하여 범위로 정의하였다. Table 5-1①은 탐색 결과에서 해당 변수의 존재성에 대해 정의하는 속성을 포함하고 있으며 이를 통해 다양한 성질의 변수를 생성할 수 있다.

Table 5-1 OPL for CP model

Function	Description	Index
interval_var	An interval variable with start time and end time of unknown thing for scheduling problem	①
endBeforeStart	Constraint on preceding relationships	②
endAtStart	Constraint on serial relationships	③
alternative	Constraint on alternative thing among variable list	④
startOf	Unknown start time of an interval variable	⑤
endOf	Unknown end time of an interval variable	⑥
presenceOf	Existence of an interval variable	⑦
sizeOf	Size of an interval variable	⑧
step_at	Cumulative function of constant value from the given point	⑨
pulse	Cumulative function of constant value between the start and the end of interval	⑩

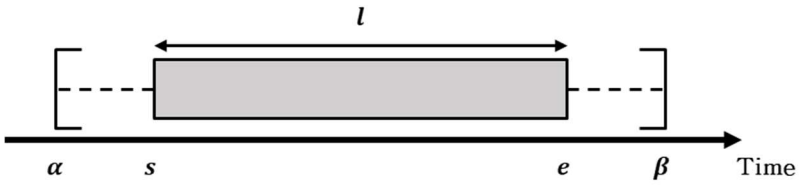


Figure 5-1 Definition of interval variable

본 모델의 global constraints 중, Table 5-1②와 Table 5-1③은 변수 간의 순서에 대한 제약조건으로 Table 5-1②는 두 변수의 선행 관계 조건을, Table 5-1③은 더 나아가 선행 변수의 마감 시점과 후행 변수의 시작 시점의 일치성을 포함한다(Dechter et al. (1991)). Global constraints 중, Table 5-1④는 변수 간의 관계를 규명하는 제약조건이다. 하나의 대표 변수 하위에 여러 optional 변수를 생성한

뒤, 일부를 대표 변수에 할당하는 제약조건으로 본 연구에서 제안하는 모델에는 여러 하위 변수들 중 1개를 대표 변수에 할당하도록 구현하였다. Figure 5-2는 Table 5-1④의 작동원리를 도식화한 것으로, Table 5-1①로 정의된 X에 Table 5-1①로 정의된 X의 가능 경우의 수 X_1, X_2, X_3 중 한 가지가 X에 최종 할당되는 것을 나타낸 것이다. 이때, 할당되는 기준은 목적함수에 대한 최적화 또는 제약조건이 되며, 모든 경우에 대해 동일한 목적함수 값이나 제약조건이 만족 될 경우, 가장 마지막에 부여된 경우의 수가 대표 변수로 할당된다.

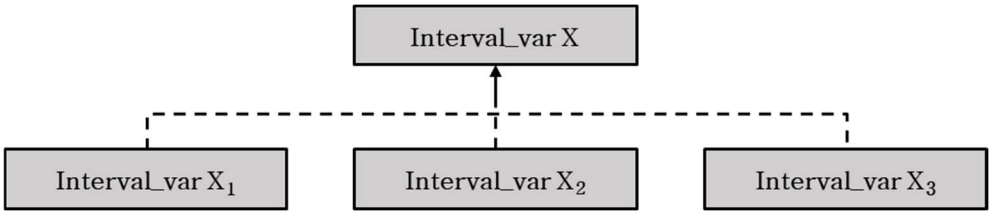


Figure 5-2 Mechanism of alternative constraint

Table 5-1⑤와 Table 5-1⑥은 Figure 5-1에서 s와 e를 의미하며, 모델 내부의 변수로서 작동하는 함수이다. 탐색 이전에 선언하기 때문에 정해진 값으로 선언할 수 없으며 목적함수나 제약조건의 계산 식을 구현하기 위해 사용된다. Table 5-1⑦ 또한 Table 5-1①의 속성 중 하나를 변수화 하기 위한 함수로, optional 속성이 부여된 Table 5-1①에 대해 탐색과정에서의 참 또는 거짓에 대한 값을 정수 1과 0으로 변환한다.

Table 5-1⑨과 Table 5-1⑩는 누적 값에 대한 제약조건 및 목적함수를 구현하기 위해 사용되는 함수들이며 각각 Figure 5-3, Figure 5-4과 같은 형태의 함수로 표현된다. 일반적으로, Table 5-1⑨에 대해 전구간의 값을 0으로 초기화 한 뒤, 모델 내부에 정의된 변수들의 점유 구간 동안의 자원 사용량을 Table 5-1⑩을 사용하여

누적하는 형태로 사용된다. 특히, Table 5-1⑨의 상한 값을 정수 값으로 정의하는 방법으로, 자원 사용 제약 조건을 구현할 수 있으며 본 연구에서는 사용하지 않는 정수 변수를 기반으로 목적 함수를 구현할 수도 있다.

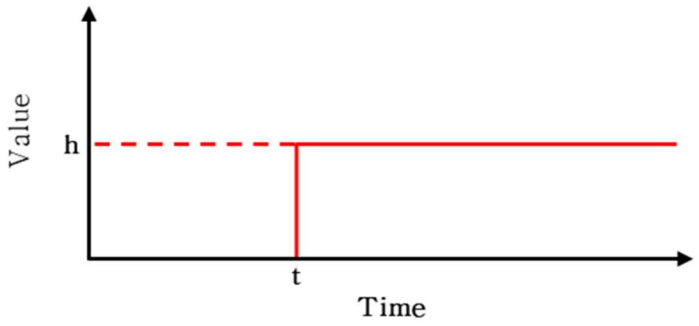


Figure 5-3 Example of step function(t → h)

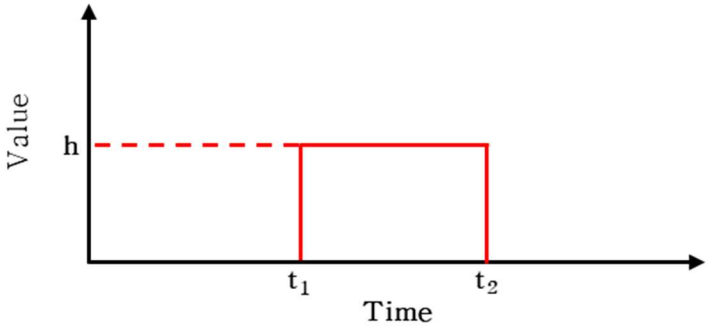


Figure 5-4 Example of pulse function(t_1 → h, t_2 → 0)

OPL기반의 CP 모델링은 MIP(Mixed Integer Programming)와 같이 수학식을 기반으로 하는 모델링과 달리 직관적이라는 점에서 제약조건 구현에 높은 자유도를 갖는다. 선표 수립 최적화 문제가 복잡한 제약조건을 포함하고 있고 각 호선에 대해 착수시점 및 마감시점에 대한 변수를 모두 구현해야 하기 때문에, 본 연구에서는 OPL을 사용하여 CP모델을 구현한다.

제 5.2 절 CP 기반 선표 수립 최적화 모델

선표 수립 최적화를 위한 CP모델은 다음과 같다.

$$\boldsymbol{Maximize}\ \lambda_1 Obj_1 - \lambda_2 Obj_2 \tag{1}$$

s.t.

$$Obj_1 = \frac{\sum(presenceOf(x_{ij}) \times eff_{ij})}{\sum\left(presenceOf(x_{ij}) \times max(eff_{ij})\right)}, \quad \forall\, i \in M_j,\ \forall\, j \in \tag{2}$$

$$Obj_2 = \frac{\sum\left(presenceOf(x_{ij}) \times \left(endOf(z_i) + pq_{ij} - d_j\right)\right)}{|J_c| \times 90}, \quad \forall\, i \in M_j,\ \forall\, j \in J \tag{3}$$

$$presenceOf(xa_{ij}) = presenceOf(xb_{ij}) \tag{4}$$

$$endAtStart(xa_{ij}, xb_{ij}), \quad \forall\, i \in M_j,\ \forall\, j \in J \tag{5}$$

$$alternative(z_j, \{(x_{ij}|i \in M_j), (xa_{ij}|i \in M_j)\}), \quad \forall\, j \in J \tag{6}$$

$$cumulative(z_j|j \in J, r_{jv}, r_v), \quad \forall\, v \in V \tag{7}$$

$$endOf(x_{ij}) \neq T_i, \quad \forall\, i \in M_j,\ \forall\, j \in J \tag{8}$$

$$endBeforeStart(z_j, z_k), \quad p_j = p_k,\ d_j < d_k \tag{9}$$

$$endBeforeStart(z_j, z_k), \quad p_j = p_k,\ d_j = d_k,\ sn_j < sn_k \tag{10}$$

$$\begin{aligned}&\left(startOf(xa_{ij}) - startOf(xb_{ij})\right)\\ &\times\left(startOf(xa_{ij}) - endOf(xb_{ij})\right) \geq 0,\\ &\left(endOf(xa_{ij}) - startOf(xb_{ij})\right)\\ &\times\left(endOf(xa_{ij}) - endOf(xb_{ij})\right) \geq 0, \quad \forall j \in J,\ \forall i \in M_j\end{aligned} \tag{11}$$

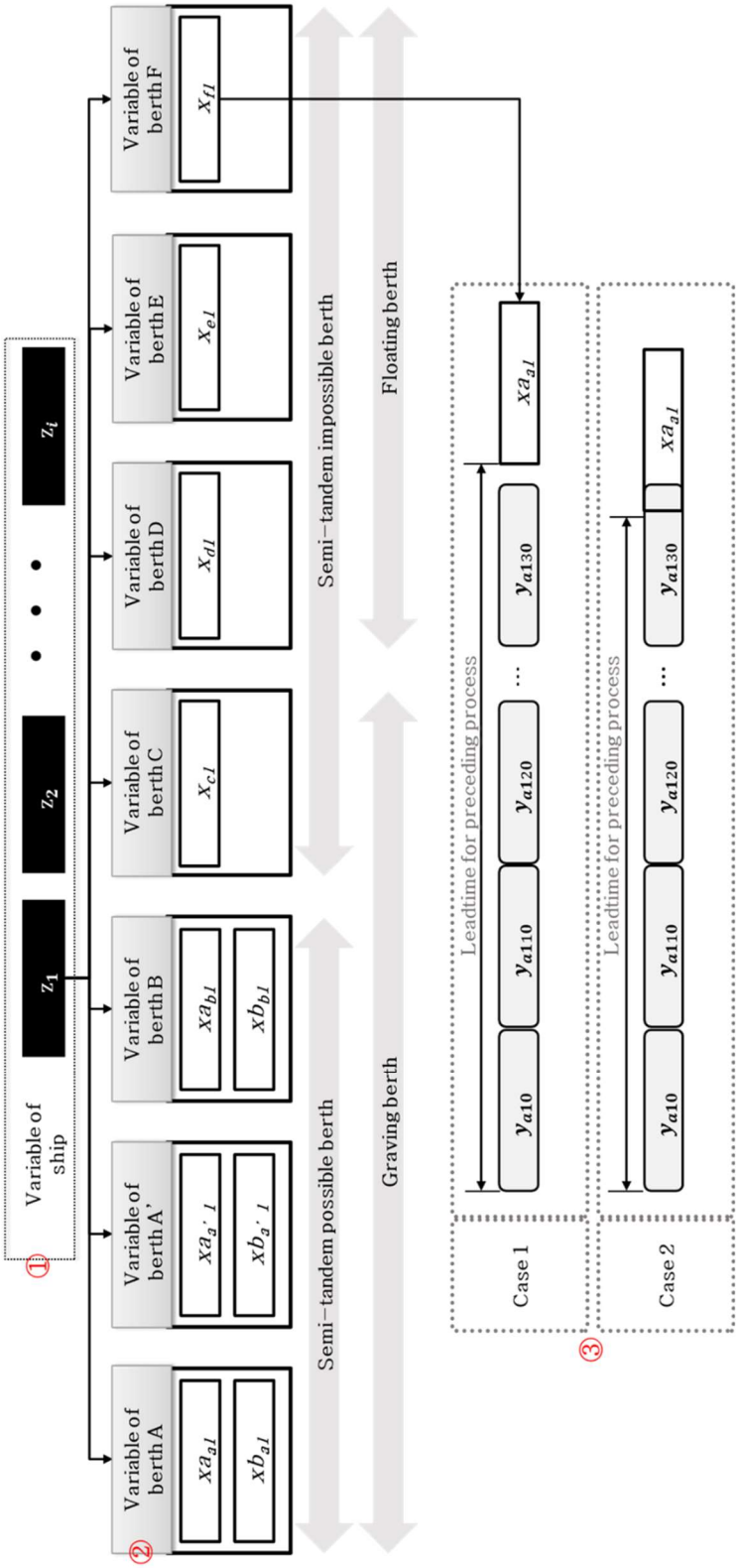


Figure 5-5 Structure for ship variable

본 모델의 변수구조는 Figure 5-5와 같다. 우선, 모든 계획 대상 호선들에 대해 대표 변수를 생성하며(Figure 5-5①) 하위에 optional 속성이 포함된 선대 별 변수들을 생성한다(Figure 5-5②). A 선대의 경우, 대부분의 건조 선박에 비해 넓은 폭과 긴 길이를 보유하기 때문에 유일하게 parallel-tandem이 가능한 선대이다. 그러나 parallel-tandem에 대한 공간 제약을 고려하기 위해, 동일 시점에 대한 길이를 비교하는 것이 복잡하기 때문에 폭 방향으로 A 선대를 반으로 나누었고 그에 따라 가상의 선대(A')를 생성하였다(Figure 5-6). 때문에 A' 선대에 대한 호선 별 변수도 생성하였지만, 사실상 동일 선대이기 때문에 제약조건들 에서는 동시에 고려된다. Figure 5-5③은 각 호선-선대 별 변수에 제약으로 구속되는 조립공정의 부하변수들을 의미한다.

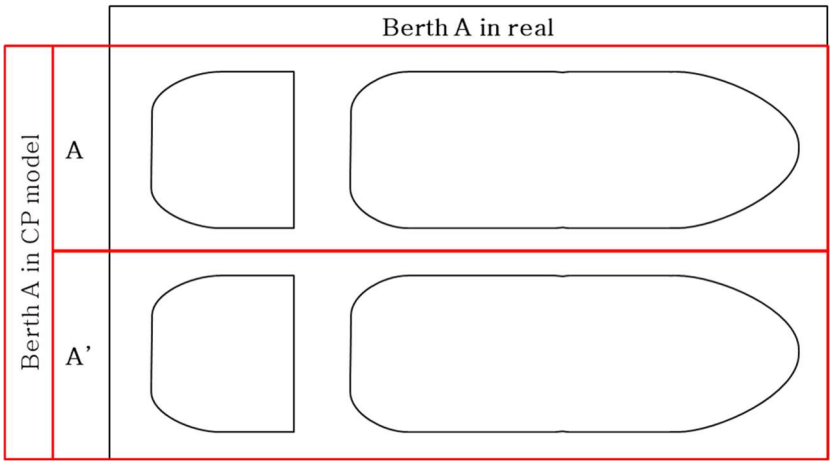


Figure 5-6 Division of berth A in CP model

부하변수들은 조립공정의 진도율을 표준 s-curve기반의 데이터를 기반으로, 진도율 구간에 따라 구현되었으며 표준 값으로 정의된 선행공정의 시작일과 종료일의 차이로 변수의 시점을 제약조건으로 정의한다. 때문에 부하 변수의 결과값은 호선 변수의 시점이 결정됨에

따라 동적으로 결정되는 것이며 이는 부하 분석 과정과 같은 역할을 하게 된다.

변수 구조의 가장 중요한 측면으로 변수 선언과 동시에 적용되는 제약사항이 있다. CP모델에서 선언하는 변수의 형태는 interval variable로 Figure 5-1과 같은 형태이다. 이는 α, β 와 같이 변수의 시점에 대한 가능 범위가 지정되어야 하며, l 이 정의되어야 함을 의미한다. 본 모델의 변수 길이는 동적변수로 주어지며 이후에 설명하게 될 key event제약조건을 구현하기 위해 필수적인 요소이다. l의 범위로, 표준 최소 공기와 변수의 선대 별 선종의 공기 중 최댓값을 갖게 된다. 또한 α의 경우, 계약 시점과 선행공정의 공기에 따라 결정되지만, β의 경우, 납기 및 재고 관리에 대한 특정 일자 범위가 주어진다. 이 때 작업 종료일이라는 일자가 고려되는데, 작업 종료일은 선대 작업 이후의 안벽 작업과 같은 후행 공정의 공기가 고려된 시점이며 계약 납기일과 최종적으로 비교되는 값이다. 본 연구에서 작업 종료일의 시점은 계약 납기일의 90일 이전과 30일 이후라는 범위에서 결정되어야 하며 변수 선언과 동시에 적용되는 가장 중요한 제약조건이다.

본 논문에서 제안하는 CP 모델의 핵심 제약조건 구현 과정에서 사용된 호선 별 optional변수는 선대의 종류에 따라 다르게 정의되었다. Optional 변수는 선대의 semi-tandem 가능 여부에 따라서 형태가 결정되며, semi-tandem공법이 가능한 선대는 탑재 기간 중에 floating 작업이 발생할 수 있기 때문에 Figure 5-7(a)와 같이 floating 이전과 floating 이후의 변수로 나누어 선언해야 한다. 다만, 수립 결과, floating 이벤트가 필요하지 않은 배치가 출력된다면 후처리 과정에서 floating 시점을 제거할 수 있다. Semi-tandem이 불가능한 선대의 호선 변수들은 Figure 5-7(b)와 같이 선언한다.

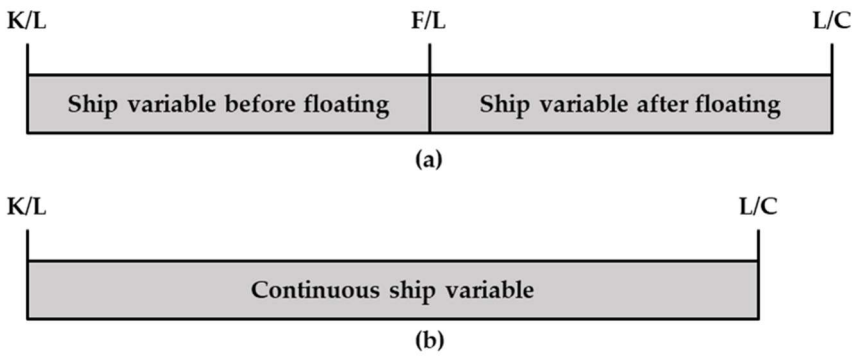


Figure 5-7 Structure of ship variable(a)at the graving berth,(b)at the floating berth

본 모델에서의 목적함수(1)는 제약조건(2)와 제약조건(3)으로 구성된다. 제약조건(2)는 배치 대상 호선의 선대 선호도를 나타내는 식이다. 선대 선호도는 작업 효율과 관련 있는 수치이며 호선 및 선대 별 변수의 존재성 여부에 해당 선호도를 곱하여 산출한다. Figure 5-8은 Obj_1의 계산 과정을 간략한 예시를 통해 도식화 한 것이다. 호선 별로 지정된 선종이 있기 때문에 탐색 과정 중에 각 선대 별로 배치된 선종의 수와 선대-선종 별 선호도를 각각 곱하여 총 합을 구하여 산출하는 방법을 사용한다.

Berth efficiency	AD	BD
ST1	10	5
ST2	3	10

Berth allocation	AD	BD
ST1	5	2
ST2	1	10

Berth priority(Obj_1)	AD	BD
ST1	10×5	5×2
ST2	3×1	10×10

Figure 5-8 Process of workstage priority calculation

제약조건(3)은 호선 별 계약 납기 일에 따른 작업 종료일 점수를 의미한다. 작업 종료일은 진수일에 후행 공정의 작업 공기를 더한 일자를 의미한다. 후행 공정 공기는 선종 및 탑재 선대에 따라 다른 값을 가지며 본 연구에서는 호선 별로 선대가 결정되지 않은 문제에 대하여 미정 값에 대해 목적함수를 정의해야 하기 때문에, 모든 호선-선대 별 변수에 대해 계산되어야 한다. 작업 종료일은 계약 납기일에 최대한 맞춰야 하며 너무 이르게 이루어져도, 계약 납기일을 초과하여 이루어져도 좋은 선표라고 보기 어렵다. 이에 따라 작업 종료일과 계약 납기일의 차이를 모든 호선에 대해 누적하여 Obj_2로 정의한다.

목적함수(1)를 구성하는 식을 다시 정리하면, 제약조건(2)로 정의된 값은 최대가 되어야 하고 제약조건(3)으로 정의된 값은 최소가 되어야 한다. 따라서 최종 목적함수는 제약조건(3)에 음의 부호를 부여하여 구성하였다. 그러나 최종 모델의 가용성을 위해 두 목적함수의 가중치 실험은 필수적이며 이는 검증과정에서 진행한다. 또한 두

목적함수의 값은 정규화를 통해 0과 1사이의 값으로 scale을 조정하였다. 정규화는 모델의 제약조건이 모두 완화되었을 경우의 값으로 실제 결과값을 나누어 진행하였다. 제약조건(2)의 분모는 모든 호선 변수의 선종-선대 별 효율 중 가장 큰 값을 갖는 선대에 배치되었을 경우를 가정하고 산출한 값이며, 제약조건(3)의 분모는 모든 계약 호선 변수의 계약 납기일과 작업 종료일의 차이가 최대인 90일 경우를 가정하고 산출한 값이다. 따라서 제약조건(2)의 값은 1을 상한으로 최대화가 진행되며 제약조건(3)의 값은 0을 하한으로 최소화가 진행된다.

제약조건(4), (5) 그리고 (6)은 선표 수립 최적화의 CSP로부터 직접적으로 발생된 제약조건이 아닌 모델의 변수를 정의하는 과정에서 부가적으로 파생되는 조건들이다. 제약조건(4)는 semi-tandem이 가능한 선대에서의 호선변수일 경우, floating 이전 및 이후 변수의 존재성은 동일해야 한다는 것이며 같은 맥락에서 제약조건(5)는 두 변수의 연속성을 의미한다.

제약조건(7)는 선표 수립을 위해서 고려해야 하는 모든 자원에 대한 조건이다. 자원의 종류에는 선대 공간, 일별 L/C 가능 척 수, 그리고 조립 공정의 일별 누적 부하 등이 있다. 선대 공간은 배치 호선들의 길이 또는 폭의 합이 선대의 길이나 폭을 초과할 수 없음을 의미하며, semi-tandem 및 parallel-tandem 가능 여부에 따라 판단 방법에 미세한 차이점이 존재한다. 일별 L/C 가능 척 수는 일정하게 정해져 있는데, 이는 L/C를 위해 사용되는 자원 할당량이 모두 반영된 결과값이다. 조립 공정의 일별 누적 부하는 호선 배치와 부하 평가를 동시에 할 수 있게 하는 조건으로 연간 capacity를 기준으로 일별 capacity를 산출하여 상한 값을 설정하여 초과할 수 없도록 반영된다. 제약조건(7)은 호선의 일정에 따라 각 자원 할당량을 누적하여 값을

제한하는 누적함수로 반영되었다.

제약조건(8)은 선대 별 진수 가능 요일에 대한 제약조건이다. 본 연구에서, 선대의 유형은 graving type과 floating type으로 구분되는데, 이는 선대의 위치가 육지인지 해상인지에 의해 결정되며 부가적으로 L/C방법과 같은 공법적인 영향을 미치는 요인이 된다. 본 연구에서, L/C 가능 요일을 결정짓는 파라미터로 사용된다:

- Graving type: 토요일,
- Floating type: 화요일, 수요일, 목요일

또한, 호선 별로 선대가 지정되기 전에 제약조건을 정의하는 방식으로, 호선의 optional 변수를 사용한다. 진수 불가능 요일에 대한 변수를 list로 정리하여 호선 변수의 종료일이 진수 가능 요일 중 선택하도록 제약조건을 정의하였다. 다만, 고정호선의 경우, 선대의 상황에 따라 임의 변경하여 선대 별 L/C 가능 요일이 아닌 경우에 L/C을 진행하는 경우가 있으며 해당 고정호선과 parallel-tandem을 해야 하는 배치의 경우, 예외적으로 L/C가능한 일자로 추가하여 계약호선이나 미정선의 가능 일자로서 지정하였다.

제약조건(9)과 (10)는 시리즈 호선에 대한 제약조건이다. 시리즈 호선은 계약일이 근사하며 선주가 동일한 선박을 의미한다. 제약조건(9)는 같은 시리즈 호선 그룹에 속해 있을 경우, 계약 납기일이 빠른 호선의 작업 종료일이 앞서야 한다는 것이며, 제약조건(10)은 같은 시리즈 호선 그룹에 속하며 계약 납기일도 동일할 경우, 사전에 부여된 시리즈 순번에 따라 작업 종료일이 지정되어야 한다는 조건이다.

제약조건(11)은 본 논문의 핵심 제약조건이며, 동일 선대에 배치되는 호선 간의 key event 일자가 어긋나게 겹치지 않아야 한다는 제약조건이다. Figure 5-9(a)는 동일 선대에 parallel-tandem 건조 중인 호선의 배치 과정에서 key event가 어긋난 상태로 겹쳐 있는 것을

나타낸 것이다. 그러나 실제 건조 과정에서 동일 시기에 건조되는 호선들 중 한 척이 L/C을 진행할 때 다른 호선이 L/C 또는 floating이 되지 않는 경우는 존재하지 않기 때문에 호선 간의 key event는 Figure 5-9(b)와 같이 수립되어야 한다.

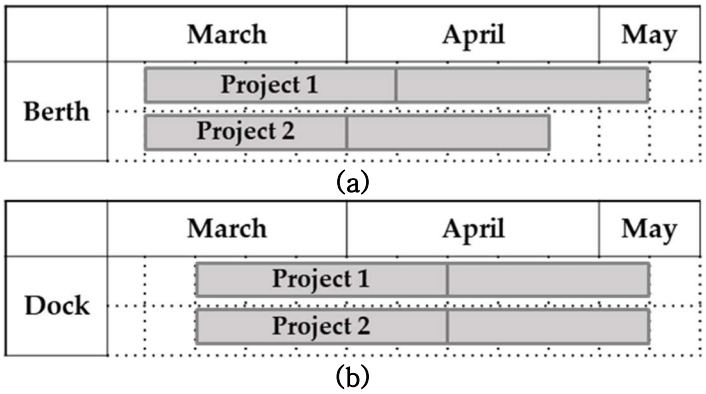


Figure 5-9 (a)Impossible batch of parallel-tandem ship (b)Possible batch of parallel-tandem ship

제 6 장 CP 모델 검증

CP기반 모델 구현의 최대 이점은 복잡한 제약조건을 구현하여 현실적인 탐색 시간을 기반으로 결과를 출력한다는 것이다. 따라서 모델 출력 결과의 제약조건 반영 및 성능에 대한 검증은 필수적인 과정이다. 이번 장에서는 검증용 데이터를 기반으로 모델을 구성하는 제약조건들의 반영 결과를 확인하고 다중 목적함수인 만큼 개별 목적함수의 가중치에 따라 어떠한 값을 갖는지에 대한 실험을 진행하여 구현된 모델의 타당성을 확보한다.

본 연구의 실험환경은 이후 모든 과정에서 동일하게 사용되며, Intel® Core™ i7-11700 CPU 2.50 Ghz 32GB RAM과 Python v3.8.10, 그리고 CP모델링을 위해 IBM ILOG CP Optimizer v22.1을 사용하였다.

제 6.1 절 실적 결과와의 비교를 통한 성능 검증

모델의 선표 최적화 성능을 확인하기 위하여 동일한 검증용 데이터를 바탕으로 기존의 rule기반 수작업으로 수립한 결과와 모델의 출력 결과를 비교하였다.

검증을 위한 데이터는 배치 대상 호선 데이터(Table 6-1)와 선대 정보(Table 6-2)로 구성된다. 선대 정보 중, berth type은 graving과 float로 나누어지며 이는 선대의 위치 및 건조 방식에 따라 분류된다. 본 연구에서는 L/C가능 요일을 결정짓는 요인으로 사용되는 정보이다. 또한 목적함수 중 Obj_1의 값을 계산하기 위한 선대-선종 별 선호도는 Table 6-3을 따르며 값이 높을수록 높은 선호도를 의미하고, 값이 0인 선대-선종 간의 변수는 생성되지 않는다. 호선 데이터는 52척의 고정호선과 143척의 미정선으로 구성되어 있고 계약호선은 제외되었다. 실적 수립 결과와 비교를 통해 모델 검증 과정을 진행하는데 계약호선의 경우, 실적 진행시에 배치를 위해 선대 탑재 기간에 표준 값 미만의 일자를 적용하는 경우가 존재하는데 이는 모델과 변수 자체의 제약조건 위반이기 때문에 모델과의 결과 비교를 위해서는 제외되어야 하기 때문이다. 선대 정보에는 선대의 명칭과 type, 그리고 각각의 tandem 가능 여부가 포함된다. 첨부한 데이터 외에 선박 및 선대의 길이 및 폭과 같은 공간적인 정보, 선대 별 L/C 가능 요일, 일별 L/C 가능 척수에 대한 정보가 입력 값으로 사용된다.

Table 6-1 Ship data for validation of constraint satisfaction

Parameter	Value
Number of fixed ships	52
Number of contracted ships	0
Number of expected ships	143
Number of shiptype	6

Table 6-2 Berth information for validation of constraint satisfaction

Berth ID	Berth type	Semi-tandem possibility	Parallel-tandem possibility
AD	Graving	O	O
BD	Graving	O	X
CD	Graving	X	X
DD	Float	X	X
ED	Float	X	X
FD	Float	X	X

Table 6-3 Berth-shiptype priority for validation

Ship type	Berth ID					
	AD	BD	CD	DD	ED	FD
ST1	7	10	10	3	7	3
ST2	5	3	3	3	5	3
ST3	0	0	0	10	7	10
ST4	0	0	0	10	5	10
ST5	10	5	0	5	3	5
ST6	3	10	0	3	8	3

Table 6-4는 본 연구에서 제안하는 CP모델과 실제 조선소에서 rule기반으로 동일 데이터에 대하여 선표를 수립한 결과를 나타낸

것이다. ‘Optimization’으로 표기된 column이 CP모델의 결과이며 ‘Heuristic’으로 표기된 column이 실적 결과이다. 선표 수립을 위해 실적 결과가 하루 이상 소요된 것에 비해 CP모델은 967초가 소요되어 동일 조건에서의 선표 수립이 월등하게 빠른 것을 확인할 수 있다.

제약조건 비교 항목에서 모델 및 실적 결과 모두 부하 항목이 제외되어있는 것을 확인할 수 있는데, 이는 실제 선표를 수립할 때 호선 배치와 부하 분석 과정이 분리 되어있어, 모델 검증 수준에서의 부하 분석은 불가능하여 모델의 제약조건에서도 제외하고 비교를 진행하였다. 이를 제외하고도 다른 제약조건의 반영 결과를 통해 모델이 모두 반영한 반면, 실적 결과에서는 일별 L/C 척 수 및 공간 제약조건이 반영되지 못한 것을 확인할 수 있었다.

Table 6-4 Result comparison between proposed model and heuristic

Content		Value	
		Optimization	Heuristic
Scheduling time		967 (s)	Several days
Constraint	Number of L/C per day constraint	O	X
	L/C constraint	O	O
	Spatial constraint	O	X
	Key event constraint	O	O
	Workload constraint	X	X
Objective function value		1	0.95

제 6.2 절 조립 공정 부하를 고려하는 선표 수립 최적화 모델 검증

실적 수립 결과와의 비교를 통해 모델의 성능을 확인할 수 있었지만 부하 제약조건은 제외하고 비교하였기 때문에 모델의 검증을 완료하기 위해서는 조립 공정 부하 제약조건의 동작 여부를 검증해야 한다. 본 절에서는 동일 데이터 및 모델을 바탕으로 부하 제약조건의 반영 여부만을 다르게 하여 결과를 비교한다. Ship data는 Table 6-5와 같으며 선대의 정보는 Table 6-2과 같다. 선종-선대 우선순위 정보는 Table 6-6을 따른다. 부하 capacity 공수 기준 55,000/year을 적용하여 주말 만을 제외한 working day 기준 약 260일로 나누어 211/day로 계산하였다.

Table 6-5 Ship data for validation of workload constraint

Parameter	Value
Number of fixed ships	12
Number of contracted ships	59
Number of expected ships	143
Number of shiptype	12

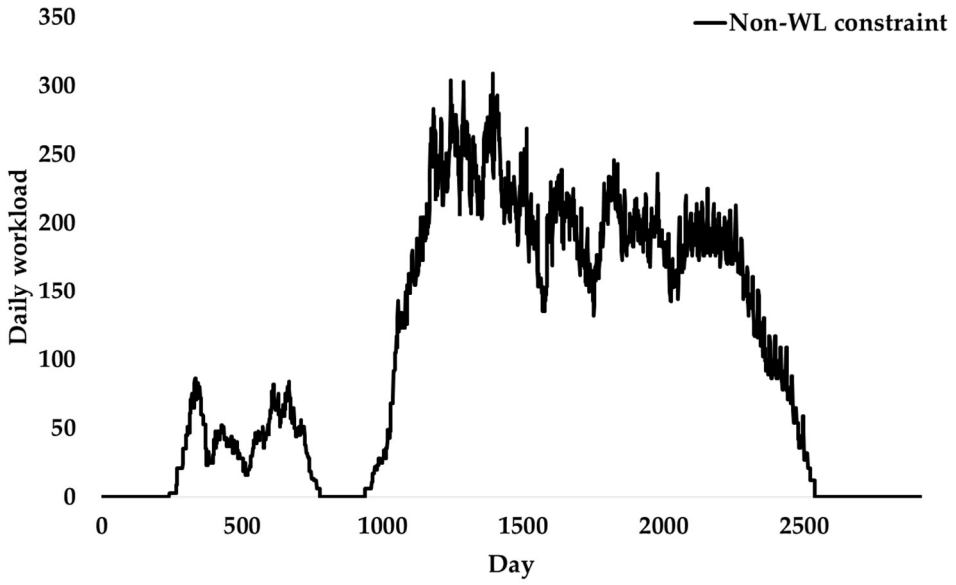
Table 6-6 Berth priority by shiptype for validation of multi-objective functions

Ship type	Berth ID					
	AD	BD	CD	DD	ED	FD
ST1	1	3	3	5	3	3
ST2	0	0	0	7	10	10
ST3	0	0	0	5	10	10
ST4	5	0	0	0	0	10
ST5	5	10	10	7	3	3
ST6	5	10	10	7	3	3
ST7	3	10	0	5	3	3
ST8	10	5	10	3	3	3
ST9	10	5	0	3	5	5
ST10	3	10	0	8	3	3
ST11	5	3	0	0	10	7
ST12	5	3	0	0	10	7

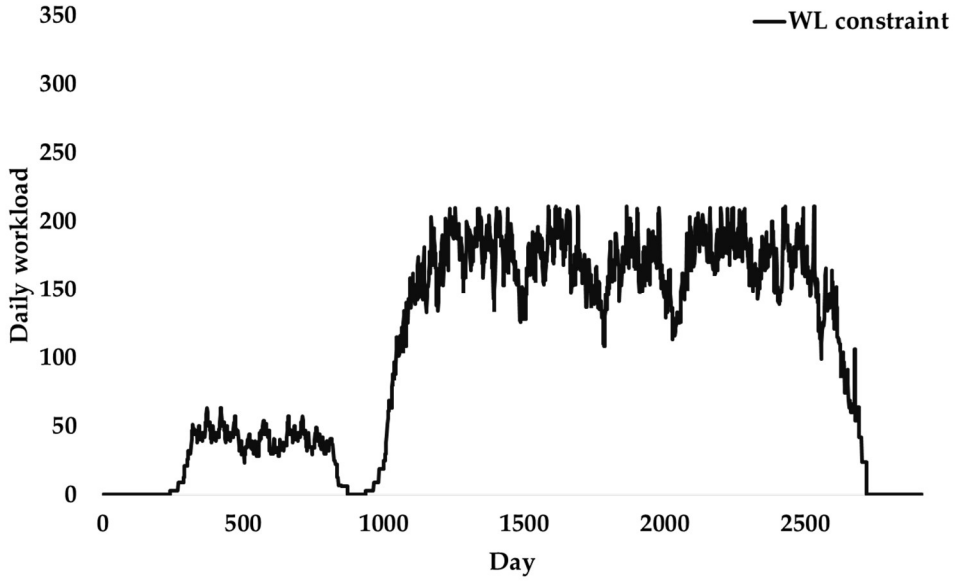
모델의 비교는 동일 모델, 동일 데이터에 대하여 부하 capacity 제약조건의 적용 유무에 따른 결과의 일별 부하 누적 값에 대하여 진행한다. 일별 부하 제약조건이 없다면 부하와 관계없이 다른 제약조건만을 만족하며 해를 탐색해 나갈 것이며 제약조건이 존재한다면, 넘치는 부하에 대해 호선의 일정을 조정하여 해를 탐색할 것이다. 비교 결과는 Figure 6-1와 같으며 (a)는 부하제약조건이 미적용 되었고, (b)는 적용되었다. 일별 부하 누적 값에 대해 (b)의 최댓값은 211로 정확히 부하 capacity를 초과하지 않도록 수립이 되었고 (a)의 최댓값은 315로 수립된 것을 확인할 수 있다.

본 절의 비교를 통해 모델 자체의 제약조건 반영에 대한 검증을

완료하였으며 다음 절 에서는 목적함수의 가중치에 따른 결과를 비교할 것이다.



(a)



(b)

Figure 6-1 Comparison of result for daily workload;(a)CP model without workload constraint, (b) CP model with workload constraint

제 6.3 절 다중 목적함수에 대한 가중치 분석

본 연구에서 제안하는 모델은 2개의 목적함수로 구성되며 각각 선대-선종 선호도(Obj_1)와 납기 준수(Obj_2)이다. 각 목적함수는 서로 상충되는 구조이며 이에 따른 가중치 실험을 통한 조정과정이 필요하다. 이 때, 고려해야 할 점은 선표의 수립 목적이며 선표는 목적에 따라 크게 영업 선표와 생산 선표로 나눌 수 있다. 영업 선표에서도 신규 계약 대상 호선의 납기 준수 파악 및 연간 목표 설정을 위한 수립 등으로 세분화될 수 있으며 생산 선표에서도 단기 적인 조정을 위한 수립과 생산량의 비정상적인 변화에 대한 대응을 위한 수립 등으로 세분화될 수 있다. 따라서 선대-선종 선호도와 납기 준수에 대한 목적함수 가중치는 상황에 따라 달라질 수 있으며 본 실험에서는 파레토 법칙을 기반으로 여러 가중치 조합에 대한 목적함수 값 변화를 파악하고자 한다. 가중치 조합은 총 9쌍으로 구성되며, 각 가중치 case에 따라 목적함수 값의 변화를 출력하고 선표 수립 목적에 따라 가중치 조합을 매칭할 수 있다. 가중치 실험에 사용되는 데이터(Table 6-7)는 고정호선 5척, 계약호선 59척, 미정선 38척으로 구성되어 있으며 선대 정보는 Table 6-1과 동일하다.

Table 6-7 Ship data for validation of weights on multi-objective functions

Parameter	Value
Number of fixed ships	5
Number of contracted ships	59
Number of expected ships	38
Number of shiptype	12

모든 가중치 case에 대해 1,800초로 동일한 시간동안 탐색을 진행하였으며, 결과는 Table 6-8과 같다.

Table 6-8 Result of the objective functions value by weight case

Case	λ_1	λ_2	Obj_1	Obj_2
1	5	95	0.867	0.396
2	10	90	0.857	0.371
3	15	85	0.876	0.420
4	20	80	0.860	0.429
5	25	75	0.872	0.399
6	30	70	0.856	0.378
7	35	65	0.856	0.373
8	40	60	0.862	0.373
9	45	55	0.859	0.430
10	50	50	0.862	0.402
11	55	45	0.866	0.385
12	60	40	0.884	0.437
13	65	35	0.875	0.413
14	70	30	0.873	0.427
15	75	25	0.910	0.531
16	80	20	0.914	0.560
17	85	15	0.976	0.784
18	90	10	0.976	0.736
19	95	5	0.981	0.789

본 논문에서 제안하는 모델에서, Obj_1은 최대화, Obj_2는 최소화로 최적화를 진행한다. 이를 바탕으로 각 가중치 case에 대한 pareto 분석 결과, case 1, case 2, case 3, case 5, case 8, case 11, case 12, case 13, case 15, case 16, case 18, case 19로 총 12개로 구성되는 pareto

optimal sets를 도출할 수 있다. Pareto optimal sets들을 바탕으로 Figure 6-2과 같이 pareto frontier line을 형성할 수 있으며 frontier를 기준으로 dominant solutions영역에 위치한 해들은 목적에 따른 가중치 선택에서 제외될 수 있다. 이러한 결과를 바탕으로, 모델 적용 목적에 따라 목적함수의 가중치는 pareto optimal sets 중 선택적으로 사용할 수 있다. 본 연구에서 해결하고자 하는 선표 수립 최적화 문제는 특정 선표에 국한되지 않는 일반적인 선표 수립 모델을 구현하는 것에 초점을 맞추었으며, 목적함수의 최적화보다는 복잡한 제약조건을 모두 구현하는 것에 우선순위를 두었다. 따라서 이후에 진행하는 실험에서, 각 목적함수 가중치의 비율은 1대1로 설정한다.

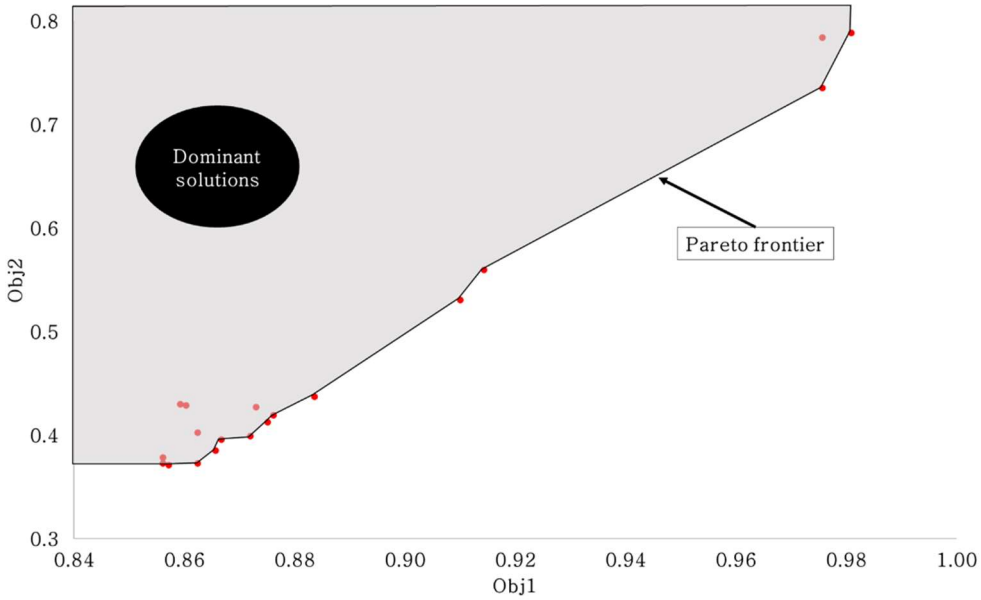


Figure 6-2 Pareto frontier by non-dominant solutions of weight case

제 7 장 사례 연구

제 7.1 절 수립 대상 호선 척 수에 따른 결과분석

본 장에서는 구현 및 검증이 완료된 CP모델을 기반으로 다양한 데이터에 따른 선표 수립 결과를 분석한다. 각 사례는 데이터의 크기에 따라 구분되며 데이터의 크기는 미정선의 척 수 증가에 따라 결정된다. 사례 별 동일 탐색 시간에 대한 각 목적함수의 변화 및 추가적인 KPI(Key Performance Indicator)에 대한 분석을 진행하며 이를 통해 현실 적용 가능성을 확인한다.

분석 대상 데이터로 선대 정보는 Table 6-7를 사용하며, 호선 및 선종-선대 별 우선순위 정보는 각각 Table 7-1, Table 7-2와 같다. 작은 크기의 문제는 고정호선 5척, 계약호선 59척, 미정선 0척으로 구성되어 있으며 큰 크기의 문제는 고정호선 12척, 계약호선 59척, 미정선 143척으로 구성되어 있다. 두 데이터 모두 구성 선종은 총 12개이다. 본 연구에서 제안하는 CP모델을 기반으로 선표 수립 최적화를 진행하며, 문제 사이즈에 따른 모델 내부에 생성되는 변수, 제약조건의 개수를 확인하고, 탐색시간에 따른 해의 개수, 목적함수 값의 개선 성능 등을 분석한다.

Table 7-1 Ship data for KPI analysis

Parameter	Value	
	Small size problem	Big size problem
Number of fixed ships	5	12
Number of contracted ships	59	59
Number of expected ships	0	143
Number of shiptype	12	12

Table 7-2 Berth priority by shiptype for KPI analysis

Ship type	Berth ID					
	AD	BD	CD	DD	ED	FD
ST1	1	3	3	5	3	3
ST2	0	0	0	7	10	10
ST3	0	0	0	5	10	10
ST4	5	0	0	0	0	10
ST5	5	10	10	7	3	3
ST6	5	10	10	7	3	3
ST7	3	10	0	5	3	3
ST8	10	5	10	3	3	3
ST9	10	5	0	3	5	5
ST10	3	10	0	8	3	3
ST11	5	3	0	0	10	7
ST12	5	3	0	0	10	7

두 문제에 대한 탐색 결과는 Table 7-3와 같다. 2개 데이터에 대해 동일한 탐색시간을 부여했으며, 모델 내부의 변수의 개수는 약 3배 차이 났으며, 제약조건의 개수는 약 6배 차이가 났다. 늘어난 호선의 척 수에

비해 변수 및 제약조건의 개수 증가량이 매우 큰 것은 모델의 복잡성을 나타낸다고 할 수 있다. 각 문제의 목적함수를 비교해보면, 거의 유사한 출력 값을 확인할 수 있다. 다만, 전체 목적함수를 구성하는 각각의 목적함수 값을 살펴보면 작은 크기의 문제에서는 Obj_1 최적화 값이 높았고 큰 크기의 문제에서는 Obj_2 최적화 값이 좋았는데 이는 호선의 구성 비율이 다르기도 했지만 목적함수가 서로 상충되며 trade off를 발생시키기 때문이다. 각 목적함수의 trade off는 각 문제의 상세 분석을 통해 확인할 수 있다.

우선, 작은 크기의 문제에 대해, 모든 제약조건이 반영되었고, 약 1,800초의 탐색 시간동안 132개의 해를 탐색하였다. Figure 7-1에 탐색된 모든 해들에 대해 2개의 목적함수 Obj_1, Obj_2 각각의 값 변화와 Obj값의 변화를 도식화 하였다. Obj는 탐색 과정에 따라 꾸준한 증가 추세를 보였음을 확인할 수 있었다. 모델을 구성하는 각각의 목적함수의 추이를 보면 21번째 부근의 해들에 대해 목적함수 값의 trade off 가 발생한 것을 확인할 수 있다. 해당 목적함수 값을 추출하면 Table 7-4와 같다. Obj는 꾸준히 개선되고 있지만, Obj_1는 악화되고 있는 것을 확인할 수 있다. 이는 Obj_1와 Obj_2가 서로 trade off를 일으켰기 때문에 발생하는 구간이며 명백한 원인이 존재한다. 유사한 작업 종료일 최적 시점을 갖는 호선들에 대해 동일한 선대에 가장 높은 선호도를 갖는다면, 어느 호선은 납기일에 대한 개선을 포기하고 선대 선호도를 선택할 수 있고 선대 선호도를 포기하고 납기일에 대한 개선을 선택할 수도 있다. 목적함수 값의 trade off는 후자를 선택한 경우에 발생하며 이에 대한 명백한 조정은 앞서 검증과정에서 언급했던 가중치 선택을 통해 이루어질 수 있다.

Table 7-3 Optimization results for KPI analysis

Evaluation factor	Small size problem	Big size Problem
Scheduling time (s)	1,800	1,800
Number of variables	6,368	17,253
Number of constraints	601,116	3,413,516
Number of solutions	132	210
Number of possible L/C for a day constraint	O	O
L/C day constraint	O	O
Spatial constraint	O	O
Key event constraint	O	O
Workload constraint	O	O
Obj_1	0.869	0.821
Obj_2	0.348	0.319
Obj	0.521	0.502

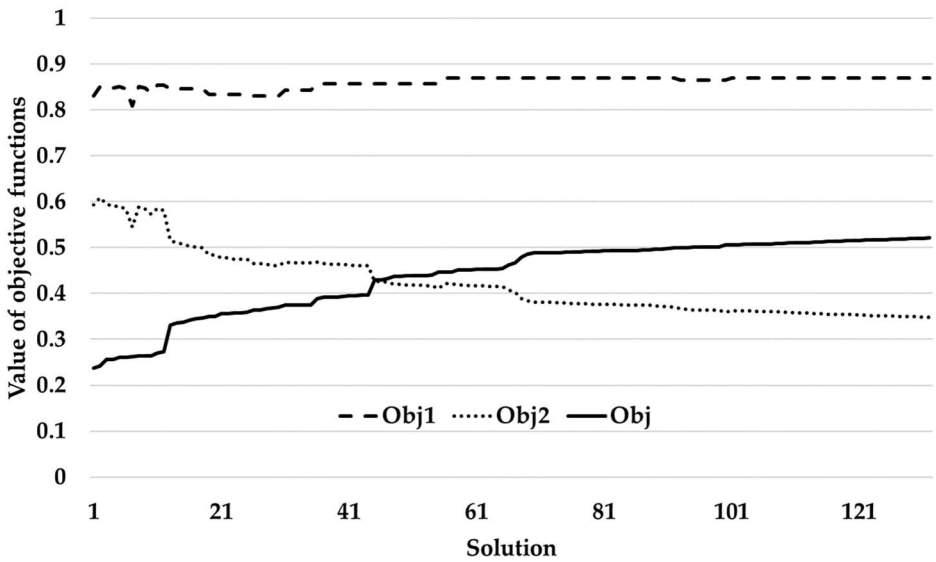


Figure 7-1 Value of objective functions(small size problem)

Table 7-4 Solutions in the range of objective function value trade off

Solution	Obj_1	Obj_2	Obj
12	0.855	0.581	0.274
13	0.846	0.515	0.331
14	0.846	0.510	0.336
15	0.846	0.508	0.338
16	0.846	0.504	0.342
17	0.846	0.501	0.345
18	0.846	0.500	0.346

다음으로 큰 크기의 문제 탐색 결과에 대해 분석을 진행하면, 모든 작은 크기의 문제와 마찬가지로 모든 제약조건이 반영되었고, 약 1,800초의 탐색 시간동안 210개의 해를 탐색하였다. Figure 7-2은 목적함수 별 값의 변화를 해에 따라 도식화 하였다. 작은 크기의 문제와 달리 큰 문제의 탐색 결과에서 trade off는 발생하지 않았다. 이는 호선의 구성 비율과도 연관이 있을 수 있는데 Obj_1의 경우, 호선의 종류와 관계없이 배치 대상의 모든 호선으로부터 영향을 받는 요소이다. 그러나 Obj_2의 경우, 미정선은 계약 납기일 없이 배치 대상 기간이 모든 시점이기 때문에 계약호선에 의해서만 영향을 받는 값이다. 2개 데이터는 동일한 계약호선 구성에 미정선의 척 수 차이가 가장 큰데, Obj_1에 영향을 주는 호선만 증가하였기 때문에 가중치가 높아지는 것과 같은 유사한 효과를 불러 일으켰다고 할 수 있다.

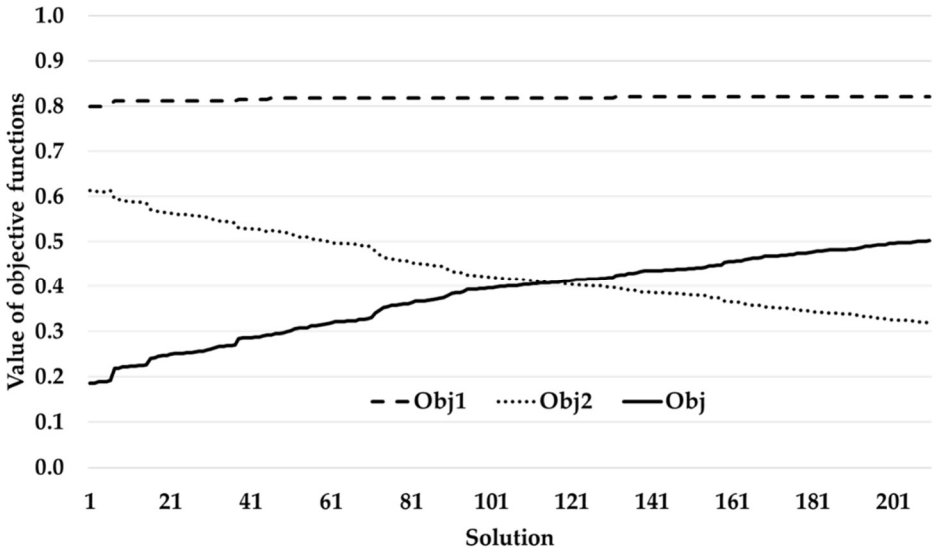


Figure 7-2 Value of objective functions(big size problem)

제 7.2 절 실험요약 및 기타 지표 분석

본 연구에서 주목할 만한 항목은 복잡한 제약조건의 구현과 목적함수의 성능이다. 제약조건 구현에 대한 사항은 검증 과정을 통해 확인하였으며, 목적함수 개선 추이는 사례연구를 통해 진행하였다. 검증 과정에서 실제 현업의 배치 결과와 비교하며 기본 제약조건에 대해 검토를 진행하였고 부하 항목에 대해서는 모델의 자체 비교를 통해 확인할 수 있었다. 실제 선표 결과를 분석할 때, 추가적으로 확인하는 지표들이 존재하는데, 공기 증가율과 선대 연속 배치 비율이 있다. 공기 증가율은 표준 데이터로 주어진 선종 별 작업 공기에 대한 배치를 위해 불가피하게 증가시킨 작업공기의 비율이고, 선대 연속 배치 비율은 호선이 선표에 배치될 때, 빈 시점 없이 연속적으로 배치되는 비율을 의미한다. 본 연구에서 제시하는 모델의 변수는 interval variable로 정의되며 공기의 가변성을 포함한다. Figure 7-3과 같이 key event 중 F/L 일치 제약조건을 만족시키기 위해 CP모델은 탐색 과정 중 project2 호선의 F/L이전 공기를 늘려야 한다. 모든 호선들에 대해 결과 적으로 총 증가한 공기의 일자를 파악할 수 있는 지표가 선대 공기 증가율이다.

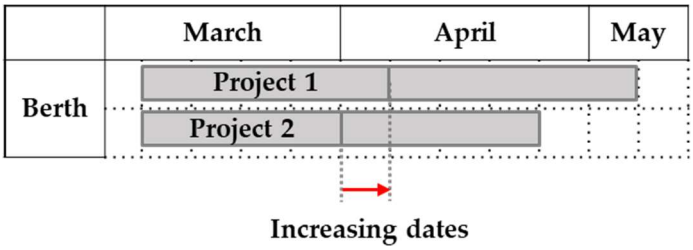


Figure 7-3 Example of increased leadtime occurrence

선대 공기 증가율을 계산하는 식은 다음과 같다.

$$R^I = \frac{\sum sizeOf(z_j)}{\sum(presenceOf(x_{ij}) \times p_{ij})}, \quad \forall i \in M, \forall j \in J \tag{12}$$

Equation (12)는 전체 호선의 R^I(선대 공기 증가율)을 계산하는 식이다. 분모의 식은 각 호선-선대 별 변수의 존재성과 호선-선대 별 공기를 곱한 값의 총 합을 나타내며 이를 통해 계획 대상 호선들의 표준 최소 공기 총 합을 계산한다. 호선-선대 별 변수를 사용하는 이유는 호선의 최소 공기가 선대에 따라 모두 다르기 때문이다. 분자의 식은 탐색 결과에 따른 호선 변수의 공기의 총 합을 나타내었으며 증가율은 1이상의 실수로 계산된다. 공기가 증가하는 대표적인 요인으로 앞서 언급만 key event 제약조건 이외에도 2가지가 더 존재한다. 첫번째로, 부하의 capacity 제약조건을 만족시키기 위해 부하일자를 조정하는 과정에서 호선 별 선대 공기를 조절하면, 호선 별 K/L 일자를 조정할 수 있기 때문에 공기 증가가 발생할 수 있다. 다른 한가지 원인은, 계약호선의 경우, 목적함수에 납기 준수 점수가 포함되며 납기일에 최대한 근사하기 위해 공기를 증가시키게 된다. 두가지 경우 모두 실제 선표 최적 관점에서 비효율적인 결과를 발생시키기 때문에 공기 증가율은 감소되어야 한다. 그러나 해당 값 최소화를 위해 목적함수에 포함하게 되면 tandem배치 관점에서 손실이 증가하게 되기 때문에 모델 개선 과정 이전에 실험을 통해 증가율 추이를 분석해야 한다.

두번째 평가요소는 선대 연속 배치 비율이다. 선표 배치 완료 기간을 기준으로 호선이 연속적으로 배치된 기간에 대한 비율을 통해 산출하며 선대 자원의 유휴시간을 분석하는 지표이다. Figure 7-4의 (a)와 (b)는 선대 선호도 관점에서 동일한 목적함수 값을 가지고 있기 때문에 CP모델 관점에서는 동일한 수준의 해로 인식한다. 그러나 실제

분석에서 (a)보다는 (b)의 배치가 효율적인 배치인데 이는 자원이 남는데 호선 배치가 되지 않았다면 유휴시간으로 포함되기 때문이다.

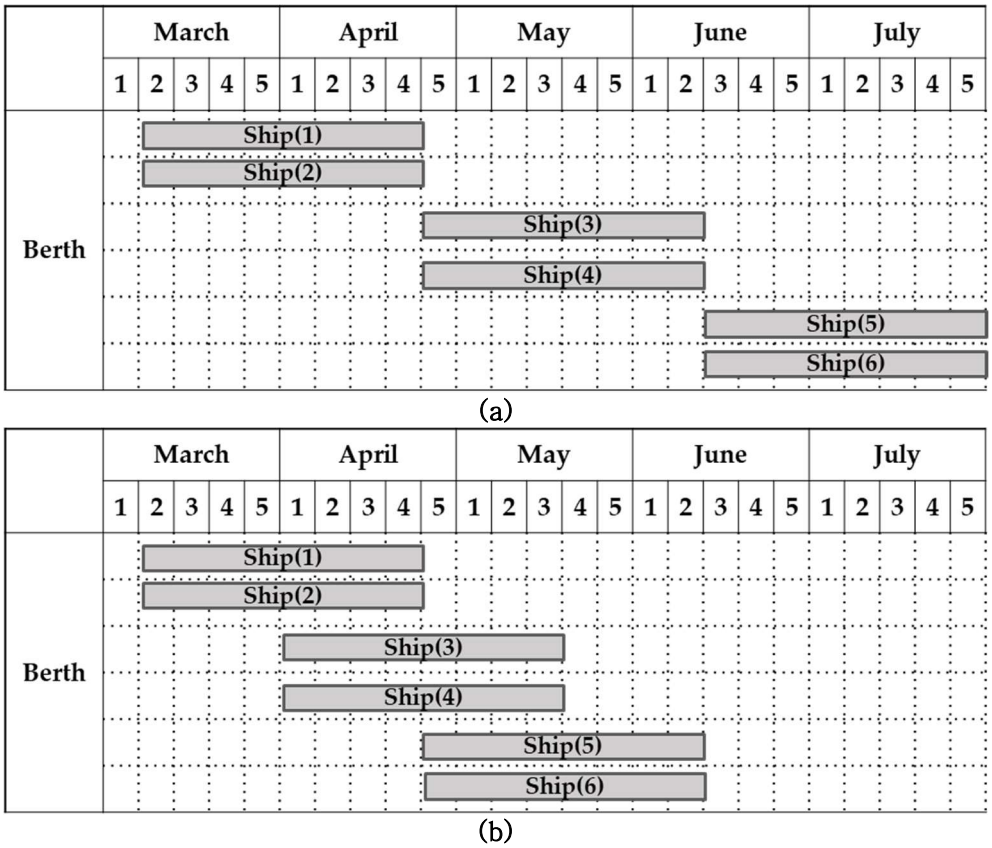


Figure 7-4 Example of berth plan for continuous batch:(a)worse case, (b)better case

유휴시간의 증가는 가능 자원의 낭비로 이어져 생산성 저하로 이어질 수 있기 때문에 유휴시간에 대한 분석은 필수이다. 선대 연속 배치 비율을 계산하는 식은 다음과 같다.

$$R_i = \frac{\sum\left(sizeOf\left(x_{ij}\right)\right)}{Max\left(endOf\left(x_{ij}\right)\right) - Min\left(startOf\left(x_{ij}\right)\right)} \quad (13)$$

$$,\forall\, i \in M,\ \ \forall\, j \in J,\ \ presenceOf\left(x_{ij}\right) = 1$$

Equation (13)은 선대 i의 연속 배치 비율을 계산하는 식이다. 분모의 식은 선대 별로 정의된 선대-선종 별 변수 중 가장 빠른 시작시점과 가장 늦은 마감시점의 차이를 계산하여 선대 별 배치 기간을 의미한다. 이 때, 선대-선종 별 변수는 optional 속성이 존재하기 때문에 존재하지 않을 수 있으며 해당 변수들은 제외된다. 분자의 식은 탐색 결과 선대 i에 배치된 호선 공기의 총 합을 의미하며 항상 분모의 값에 비해 작거나 같은 값을 갖는다.

2가지 데이터에 대한 기타 지표 분석 결과는 Table 7-5과 같다. 선대 공기 증가율의 경우, small size problem에 비해 big size problem에서 우수한 결과를 보이는데 이는 선표의 배치 가능 시점이 너무 넓어, 무의미한 공기 증가가 발생할 수 있는 가능성에 대하여 추가적인 호선 물량이 빈 시점을 점유하면서 개선시켰음을 의미한다. 선대 연속 배치 비율의 경우, big size problem에서 다소 낮은 성능의 결과를 보였는데, 이는 고정호선의 구성 호선이 다르기 때문으로 해석할 수 있다. 고정호선은 배치가 완료된 호선으로 계약 납기일이 선표 수립 시점에 비해 이른 경우가 대부분이지만 미정선의 경우, 최소 수립 시점 보다 미래의 시점이 배치 가능 시점으로 고려되기 때문에, 연속 배치가 불가능한 경우가 늘어날 가능성이 매우 높다. 또한 parallel-tandem 이

가능한 선대의 경우, 폭이 넓은 호선이 배치되면 한 척의 호선만 배치가 되는데, 계산 식은 parallel-tandem이 가능한 선대의 모든 시점에 대해 병렬 배치를 전제로 계산이 되기 때문에 선대 연속 배치 비율은 본 연구의 모델과는 무관하지만 명확한 기준을 바탕으로 평가할 수 있는 연구가 필요함을 알 수 있었다.

Table 7-5 Indices according to size of problem

Content	Value	
	Small size problem	Big size problem
$R^I(\%)$	119.382	111.714
$R_A(\%)$	68.457	62.225
$R_B(\%)$	14.233	77.598
$R_C(\%)$	0	85.625
$R_D(\%)$	93.333	64.767
$R_E(\%)$	69.814	77.322
$CR_F(\%)$	58.985	72.211

본 장에서는 사례연구를 통해 모델의 성능을 확인하였고 기타 지표에 대한 영향도를 분석하였다. 수립 속도 관점에서 기존의 선표 수립 과정에 소요되는 시간에 비해 우수한 성능을 나타내었고 기존의 선표 수립 과정과 유사한 모델로 구현이 되었기 때문에 실적 데이터를 기반으로 다양한 실험 및 비교를 진행해 볼 수 있음을 확인할 수 있었다.

제 8 장 결론

본 논문에서, 조립 공정의 부하를 고려하는 선표 수립 최적화 모델 구현에 대한 연구를 진행하였다. 기존의 선표 수립 과정은 수립된 선표에 대해 최적해라는 보장을 하기 어려웠고, 수립하는데 약 1~2 주라는 적지 않은 시간이 소요되었다. 계획자의 경험과 암묵지에 따라 수립되기 때문이고, 과정이 분리되어 반복되기 때문이라는 점에 원인이 있다고 판단하였고 이를 해결하고자 명확한 제약조건과 목적함수를 정의하고 부하 분석 과정과 호선 배치 과정을 통합하는 모델을 구현하였다.

선표 수립 최적화 모델은 복잡한 제약조건을 반영하는데 특화된 제약 프로그래밍을 사용하여 구현되었으며, 검증과정에서 실적 결과와 비교한 결과, 우수한 성능 및 제약조건이 모두 반영되는 것을 확인할 수 있었다. 또한, 사례연구를 통하여 다양한 크기의 문제에 대해 목적함수 최적화 성능을 평가하였고, 실제 계획자의 관점으로 수립된 선표를 평가하기 위해 기타 평가 지표를 도입하였다. 기타 평가 지표로 선대 공기 증가율과 선대 연속 배치 비율을 사용하였으며 모델 관점에서 평가 지표에 대한 개선점이 발견되지는 않았다. 그러나 평가 지표 자체에 대한 명확한 기준을 세운다면 모델에 추가적으로 반영할 수 있으며 이는 실제 데이터와 비교 분석을 통하여 진행할 수 있다.

기존의 선표 수립 최적화 연구들이 많이 진행되어 왔지만 선표의 호선 배치 시점을 결정하여 매칭하는 방법을 사용하거나 부하 분석과 호선 배치가 분리된 연구들이 대부분이었다. 본 연구에서는 선표 수립 최적화를 위해 동적으로 호선 배치 시점을 결정할 수 있고, 호선 배치와 부하 분석이 동시에 진행되었으며 2가지 사항들에 대해 기존의 연구와

가장 큰 차별점을 나타낸다.

본 연구에서, 선표 수립 최적화 모델을 구현하였고 성능 평가에 대한 부분도 검증 및 사례연구를 통해 완료하였다. 그러나 실적 데이터와의 비교는 검증을 통해 한 번 밖에 진행하지 못했다. 때문에 향후에 CP모델을 바탕으로 다양한 실적 데이터와의 비교가 필요하고 이를 통해 모델의 타당성을 높일 수 있다. 또한 사례연구에서 추가적으로 분석했던 기타 지표들에 대해, 명확한 기준을 수립하여 수립 이후 분석이 아닌 수립 과정에서의 최적화를 통해 세부적인 과정까지 통합하는 모델을 구현할 예정이다. 더 나아가, 중기 계획의 부하만을 고려하는 것이 아니라, 후행 공정인 안벽 공정과의 연결과 같은 연구도 진행하여 단계적인 생산계획이 아닌 통합 생산 계획으로 범위를 넓혀 가며 전체 공정에 대한 최적 계획을 수립하는 모델을 구현할 계획이다.

참고 문헌

Afzalirad, M., & Rezaeian, J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions. *Computers & Industrial Engineering*, *98*, 40-52.

Backer, B. D., Furnon, V., Shaw, P., Kilby, P., & Prosser, P. (2000). Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of heuristics*, *6*, 501-523.

Barták, R. (1999). Constraint programming: In pursuit of the holy grail. Proceedings of the Week of Doctoral Students (WDS99),

Cappadonna, F., Costa, A., & Fichera, S. (2013). Makespan minimization of unrelated parallel machines with limited human resources. *Procedia Cirp*, *12*, 450-455.

Chen, J.-F. (2005). Unrelated parallel machine scheduling with secondary resource constraints. *The International Journal of Advanced Manufacturing Technology*, *26*, 285-292.

Chen, J.-F., & Wu, T.-H. (2006). Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. *Omega*, *34*(1), 81-89.

Costa, A., Cappadonna, F., & Fichera, S. (2013). A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, *69*, 2799-2817.

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial intelligence*, *49*(1-3), 61-95.

Edis, E. B., & Oguz, C. (2012). Parallel machine scheduling with flexible resources. *Computers & Industrial Engineering*, *63*(2), 433-447.

Goel, V., Slusky, M., van Hoeve, W.-J., Furman, K. C., & Shao, Y. (2015). Constraint programming for LNG ship scheduling and inventory management. *European journal of operational research*, *241*(3), 662-673.

Ham, A. M., & Cakici, E. (2016). Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches. *Computers & Industrial Engineering*, *102*, 160-165.

Ju, S., & Woo, J. H. (2021). Integration of long-term planning and mid-term scheduling of shipbuilding. *Production Planning & Control*, 1-19.

Kwak, D. H., Cho, Y. I., Choe, S. W., Kwon, H. J., & Woo, J. H. (2022). Optimization of long-term planning with a constraint satisfaction problem algorithm with a machine learning. *International Journal of Naval Architecture and Ocean Engineering*, *14*, 100442.

Laborie, P., & Rogerie, J. (2008). Reasoning with Conditional Time-Intervals. FLAIRS conference,

Laborie, P., Rogerie, J., Shaw, P., & Vilím, P. (2018). IBM ILOG CP optimizer for scheduling: 20+ years of scheduling with constraints at IBM/ILOG. *Constraints*, *23*, 210-250.

Novas, J. M. (2019). Production scheduling and lot streaming at flexible job-shops environments using constraint programming. *Computers & Industrial Engineering*, *136*, 252-264.

Piersma, N., & van Dijk, W. (1996). A local search heuristic for unrelated parallel machine scheduling with efficient neighborhood search. *Mathematical and Computer Modelling*, *24*(9), 11-19.

Qin, T., Du, Y., Chen, J. H., & Sha, M. (2020). Combining mixed integer programming and constraint programming to solve the integrated scheduling problem of container handling operations of a single vessel. *European journal of operational research*, *285*(3), 884-901.

Smith, B. M. (1995). A tutorial on constraint programming.

Villa, F., Vallada, E., & Fanjul-Peyro, L. (2018). Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource. *Expert Systems with Applications*, *93*, 28-38.

Wang, I.-L., Wang, Y.-C., & Chen, C.-W. (2013). Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics. *Flexible Services and Manufacturing Journal*, *25*, 343-366.

Wang, T., Meskens, N., & Duvivier, D. (2015). Scheduling operating theatres: Mixed integer programming vs. constraint programming. *European journal of operational research*, *247*(2), 401-413.

Yunusoglu, P., & Topaloglu Yildiz, S. (2022). Constraint programming approach for multi-resource-constrained unrelated parallel machine scheduling problem with sequence-dependent setup times. *International Journal of Production Research*, *60*(7), 2212-2229.

김광욱. (1991). 선박설계· 생산기술의 전산화 실태. *대한조선학회지*, *28*(3), 54-59.

김기동, 우상복, & 한형상. (2001). 자원 제약을 고려한 조선산업에서의 탑재 일정계획에 관한 연구. *산업공학 (IE interfaces)*, *14*(3), 218-226.

김태호. (2020). *조선소 강재 적치장의 선별 작업 개선을 위한 강화학습 적용* 서울대학교 대학원].

류지성, 김홍태, 박진형, 이병로, & 신종계. (2004). 제약만족기법 기반의 최대부하감소를 통한 탑재네크워크의 부하평준화. *대한조선학회 논문집*, *41*(5), 55-62.

민상규, 이민우, 문치웅, 정귀훈, & 박주철. (2000). 조선 탑재일정의 부하 평준화를 위한 유전 알고리듬. *산업공학 (IE interfaces)*, *13*(2), 225-233.

박주철, 옥철영, 이태억, 정동수, & 이강렬. (1995). 조선기본계획수립시스템의 개발. *산업공학 (IE interfaces)*, *8*(2), 65-75.

배희철, 박경철, 차병철, & 문일경. (2007). 유전자 알고리즘을 활용한 조선 소조립 공정 일정계획. *IE Interfaces*, *20*(1), 33-40.

안남수, 강자영, 홍순익, 최화용, & 박성수. (2007). 조선산업에서 안벽 배치 계획 문제에 관한 해법. *대한산업공학회 추계학술대회 논문집*, 362-370.

여성주, 정인성, 구락조, 배재호, & 왕지남. (2008). 효율적인 선박 안벽배치를 위한 휴리스틱 Rule 생성. *대한산업공학회 춘계공동학술대회 논문집*, 758-765.

유병철, & 이민우. (1996). 0-1 정수계획법을 이용한 일정계획 시스템 개발. *한국공업경영학회지*, *19*(38), 139-146.

윤덕영, & 김근철. (1993). 공정계획/일정계획 지원 시스템 개발을 위한 개념 정립. *대한조선학회지*, *30*(4), 37-40.

이봉현, & 서윤호. (2020). 조선 블록 조립과 트랜스포터 통합 일정계획 알고리즘 개발. *한국생산관리학회지*, *31*(4), 427-443.

허영주, 조정헌, 정문호, 윤성태, 이창호, 최형림, 박춘희, & 류광렬. (1998). 계층적 접근에 의한 계획선표 최적화 시스템. *한국지능정보시스템학회 학술대회논문집*, 385-393.

Abstract

Optimization of long-term planning based on constraint programming considering mid-term plan workload

-Berth planning optimization model considering block assembly workload-

Kiyoung Cho
Naval architecture and ocean engineering
The Graduate School
Seoul National University

The berth scheduling process pertains to the long-term scheduling within the shipyard's production plans and directly influences the annual operations of the shipyard. Due to its nature as a shipbuilding schedule spanning 3 to 6 years or even longer, the berth scheduling involves a large number of target ships, and it often requires a significant amount of time for planning, mainly relying on heuristic judgment by planners in most shipyards. Moreover, to consider the load of preceding processes, the method of planning multiple berths scheduling cases and subsequently calculating the berth case-specific loads is employed. However, this approach becomes a factor that increases the planning time. As a result, the berth scheduling is currently being done based on trial and error

without a specific algorithm, and this limitation arises due to the constraints of the search space, which prevents the guarantee of the optimality of the resulting schedules.

In this paper, we have developed a constraint programming-based model for optimizing berth scheduling by defining the constraints and objective functions. The model aims to derive the optimal berth schedules that satisfy these defined constraints and objective functions. Constraint programming is based on logical operations, enabling the straightforward modeling of complex constraints, and it leverages graph theory, making it a suitable method for solving scheduling problems effectively. The berth scheduling optimization problem is highly complex, consisting of numerous intricate constraints and covering an extensive time frame, making it a large-scale problem. Due to these challenges, we employed the constraint programming methodology to address this optimization problem effectively.

The proposed model in this paper includes five key constraints, including daily launching frequency, possible launching dates for each ship type, shipyard space, daily workload of assembly processes, and key event constraint. By adding the daily workload constraint of the processes, the berth scheduling and the workload analysis of preceding processes have been integrated into a single model, avoiding them being treated as separate tasks. Utilizing constraint programming, the model efficiently searches for the optimal solutions, enabling fast berth scheduling.

The model considers two main objective functions: maximizing

workstage efficiency and ensuring contract deadline compliance for berth scheduling. To validate the model's effectiveness, we compared the model's results with actual performance data and verified the constraints using the same dataset. Additionally, we conducted a weighted analysis of multiple objective functions to derive Pareto optimal solutions, demonstrating the versatility of the model for berth scheduling. After ensuring the validity of the constraints and objective functions, we conducted a case study on various ship configurations and target ship quantities. During the case study, we conducted an exploration of solutions, analyzing the objective function values of the obtained results. Additionally, we performed an analysis of metrics such as ship leadtime increasing rate and consecutive assignment ratio by berth to gain a comprehensive understanding of the berth scheduling solutions generated by the model.

In this paper, we propose a constraint programming-based model for optimizing berth scheduling. The developed model overcomes the limitations of planner-based berth scheduling, which heavily relies on implicit knowledge and experience and is constrained by planning time, to guarantee optimal solutions. Moreover, the model integrates dynamic optimization for the target period, which was not addressed in previous studies, along with load analysis, offering a comprehensive approach to berth scheduling optimization. Through this research, we have established a theoretical foundation for berth scheduling optimization and achieved practical implementation in optimizing berth scheduling.

Keywords : Berth scheduling, Ship production planning, Constraint satisfaction problem, Constraint programming, Optimization, Long-term planning, Integrated production planning
Student Number : 2021-23872