

수리 모형을 이용한 조선 산업에서의 블록 적치장 최적 운영 계획 도출*

하병현** · 손정열*** · 조규갑** · †최병천****

A Mathematical Programming Approach for Block Storage Problem in Shipbuilding Process

Byung-Hyun Ha** · Jung-Ryoul Son*** · Kyu Kab Cho** · †Byung-Cheon Choi****

■ Abstract ■

This paper studies the scheduling problem of storing and retrieving assembly blocks in a temporary storage yard. The objective is to minimize the number of relocations of blocks while the constraints for storage and retrieval time windows are satisfied. We present an integer programming model based on multi-commodity network flows, and the three revised models based on the properties of the problem. We show that the revised models are more efficient than the generic model through the numerical experiments.

Keywords : Scheduling, Storage and Retrieval Problem, Mathematical Programming, Shipbuilding Industry

논문접수일 : 2013년 08월 30일 논문게재확정일 : 2013년 11월 07일

논문수정일 : 2013년 10월 29일

* 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

** 부산대학교 산업공학과

*** 대우조선해양 중앙연구소

**** 충남대학교 경영학부

† 교신저자, polytime@cnu.ac.kr

1. Introduction

전통적으로 조선소에서는 프로젝트 쉘(project shop) 형태인 고정위치 배치로 선박을 건조해왔으나, 현재는 보다 효율적인 블록분할공법이 일반화되었다. 블록 분할 공법에서는 전체 선박을 적절한 크기의 블록(block) 단위로 나누어 개별적으로 완성한 후 하나의 선박으로 조립한다. 개별 블록은 주요하게 선각공정과 의장공정, 도장공정을 거쳐 완성된다. 각 공정은 전용 생산 시설에서 이루어지며 블록은 계획된 일정에 따라 생산 시설들 간을 이동하게 된다. 완성된 블록들은 도크에서 탑재공정을 통해 하나의 선박으로 조립되고, 최종적으로 안벽의장공정이 수행된다. 이러한 공법은 잡 쉘(job shop)에서 부분품을 가공하여 조립하는 방식과 유사하다.

본 연구는 정해진 생산 일정 계획 하에서의 블록 적치장(block storage) 운영문제를 다룬다. 규모가 큰 조선소에서는 여러 척의 선박의 건조가 동시에 진행된다. 그에 따라 생산 시설들 사이에는 많은 수의 블록 흐름이 발생하며, 직전 공정이 끝난 블록들은 필요한 경우 다음 공정에 투입될 때까지 적치장에 임시 저장된다. 조선소에는 생산 시설이 다수 분산되어 존재하며 적치장 역시 다수 존재한다. 블록 생산 일정은 선박 건조 완료일을 바탕으로 계획되고, 계획된 생산 일정에 따라 각 블록이 저장될 적치장과 입고하고 출고해야 하는 시간이 정해진다.

우리는 특정 적치장 하나의 운영을 대상으로 한다. 블록 적치장의 형태와 운영 방식은 개별 조선소에 따라, 또 적치장의 여건에 따라 달라진다. 본 연구에서는 직사각형 형태의 적치장을 고려하며, 블록은 적치장의 한 측면만을 통해 입출고된다고 가정한다. 이는 부지가 협소한 국내 조선소에서 많이 사용되는 형태이다. 특히 전체 공간이 행(row) 단위로 구획 지어져 운영되는 적치장을 대상으로 한다.

적치장의 운영을 위해서는 개별블록의 입출고 시점과 저장 위치를 계획해야 한다. 블록은 다양한 크기와 형태를 지닌 거대 중량물로, 전용 운반대위에 올려 특수 차량(shipyard transporter)으로 이

송한다. 블록 이송에는 준비와 마무리 작업을 포함해 많은 시간과 비용이 소요될 뿐 아니라 예외상황의 위험도 항상 존재한다. 일반적으로 조선소에서는 이와 같은 불확실성을 감안하여 생산 일정을 계획한다. 따라서 적치장에서의 입출고 시점은 생산에 차질을 빚지 않도록 일정 기간(time window) 내에서 결정된다.

블록은 적치장에서의 저장 위치에 따라 출고 전에 다른 곳으로 옮기는 재취급(relocation) 작업의 대상이 될 수 있다. 적치장의 안쪽에 있는 블록을 출고하기 위해서 출구 쪽에 위치한 간섭 블록을 치워야하기 때문이다. 조선소에서는 가능한 재취급 작업을 피하려 한다. 본 연구에서는 재취급 회수를 최소화 하는 블록의 입출고 시점과 저장 위치를 도출하는 것을 목적으로 한다.

추가적으로 본 연구에서는 한정된 대상 기간 내의 입출고 요구만을 바탕으로 최적적치장 운영 계획을 수립한다(rolling-horizon planning). 선박의 건조 과정 중 발생할 수 있는 자재 입고 지연, 품질 불량, 기상 변화 등의 불확실성을 고려할 때 장기 생산 계획은 변경될 가능성이 크기 때문이다. 그리고 단기 계획의 변경 가능성 역시 고려하여 간섭 블록이 아닌 경우 미리 재취급해두는 것을 허용하지 않는다. 즉, 특정 블록의 재취급은 안쪽에 있는 블록이 출고될 때만 가능하다. 이와 같은 정책은 컨테이너 터미널의 야적장운영(예를 들어[1, 4]) 등 비슷한 환경에서 많이 사용된다.

조선 산업에서 생산 계획을 위한 많은 연구들이 수행되어 왔으며[5, 7], 통합 생산 계획을 체계화하기 위한 연구가 수행되었다[16]. 그 중 블록 조립 공장의 운영의 문제가 특히 많이 연구되었다[2, 6, 8, 15]. 이는 생산 일정을 고려하여 공간효율의 극대화하는 문제로, 본 연구의 대상인 적치장 운영과 유사한 속성을 가진다. 하지만 조립 공장에서는 크레인을 사용하여 입출고가 비교적 자유로우며, 한 번 놓인 블록은 출고가 되기 전까지 거의 움직이지 않는 특성이 있어 적치장 운영에 직접 적용하기 힘들다.

블록 적치장 운영의 어려운 점과 생산성에 미치

는 영향은 [11]에서 상세히 논의되었으며, 발견적 기법도 같이 제시되었다. 구체적인 운영 계획의 수립을 위해 본 연구와 유사한 환경에서의 적치장 운영 문제가 탐구되었다[12-14, 18]. 이들 연구에서는 근사해 탐색을 위한 발견적 기법이 제시되었다. 동시에 최적 계획을 위한 수리 모형이 수립되었으나, 복잡도로 인해 모두 문제를 형식화하는데 그쳤으며 보다 나은 해의 탐색에 이용되지는 못했다. 여러 방향으로의 이동과 다양한 크기 및 형태의 블록을 고려하는 연구도 수행되었다[9]. 하지만 문제의 높은 자유도로 인해 수리 계획 모형은 도출하지 못하였으며, 터부 탐색(tabu search)을 사용한 근사해법을 제시하였다.

한 방향으로 블록을 입출고하는 운영 환경은 컨테이너 등의 물품을 임시 저장 장소에 쌓아서 보관하는 경우와 유사하며 일반적으로 재취급을 최소화하는 목적을 가진다[3, 4, 10, 19, 20]. 그러나 구체적인 운영방식에서 조선소 적치장과 다르며, 특히 재취급되는 물품들이 저장 장소 외부에 놓여 질 수 없는 점에서 차이점을 가진다. 역시 문제의 복잡도로 인해 수리 모형을 사용하여 최적해를 도출하는 것은 거의 불가능한 것으로 보고되었다.

본 연구에서는 최적 적치장 운영 계획을 수립하기 위한 수리 계획 모형을 제안하고, 상용 solver를 사용하여 실용적인 크기의 문제에 대해 최적해를 구하는 방안을 제시한다. 수리 모형은 다품목 네트워크 흐름(multi-commodity network flow)을 바탕으로 수립되었으며, 특히 재취급 최소화라는 목적에 맞추어 네트워크를 효율적으로 구성하는 방법을 도출한다. 그리고 문제의 특성을 기반으로, 최적해를 보장하면서 동시에 solver가 보다 쉽게 해를 구할 수 있도록 하는 모형 변경 방안을 제시하였다. 변경된 모형의 효과는 수치 실험을 통해 파악하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 적치장 운영 문제를 상세히 기술하고, 제 3장에서는 최적 계획을 위한 기본 수리 계획 모형을 도출한다. 문제의 특성을 바탕으로 수정한 수리 모형을

제 4장에서 제시하고, 제 5장에서 수치 실험을 통해 제안한 모형의 성능을 파악한다. 마지막으로 제 6장에서 결론 및 추후 과제를 제시한다.

2. Problem Description

적치장은 세로 방향으로 n 개의 행으로 나누어져 운영되며, 행은 1부터 n 까지로 구분 된다. 블록의 입출고는 적치장의 오른쪽 측면(각 행의 오른쪽 끝)을 통해 이루어지며, 각 블록은 하나의 행에 다른 블록들과 함께 들어온 순서대로 일렬로 저장된다. 모든 행의 길이 즉 적치장의 가로 길이는 L 이다. 한 행에 저장된 모든 블록들의 길이의 합은 행의 길이 L 보다 클 수 없다. 개별 블록은 둘러싸는 경계 사각형(bounding rectangle)으로 고려되며, 블록 b 의 길이는 l_b 로 나타낸다. 블록은 세로 방향으로 너비가 충분한 행에 저장될 수 있다. 각 행의 너비는 블록들의 크기를 감안하여 미리 결정되어 있다고 가정한다. 본 논문에서는 구체적으로 행의 너비에 대한 제약을 고려하지 않지만, 일반화는 어렵지 않다.

행의 전체 길이와 개별 블록의 길이를 고려해 한 행에 최대 저장할 수 있는 블록의 개수를 m 이라 하자. 그러면 적치장에서 블록을 저장하는 위치(location)는 행의 번호 i 와 안쪽에서부터의 순서 j 로 나타낼 수 있다; 즉, $1 \leq i \leq n$ 과 $1 \leq j \leq m$ 에 대하여 (i, j) 는 행 i 에서 왼쪽으로부터 j 번째 위치를 가리킨다. 행의 안쪽에 있는 블록을 출고 또는 재취급하기 위해서는 바깥쪽에는 간섭 블록이 없어야 한다. 즉 특정 시점에 (i, j) 에 위치한 블록을 출고하는 경우 $j < j' \leq m$ 인 (i, j') 에 위치한 모든 블록은 출고되거나 재취급된다.

전체 계획 기간(planning horizon)은 단위 기간(period)으로 구분된다. 각 단위 기간에서 개별 블록은 입고, 출고, 재취급되거나 저장된 위치에 그대로 있게 된다. 단위기간 1부터 T 까지의 계획을 수립하며, 한 단위 기간 동안 적치장은 다음과 같은 작업 순서로 운영된다 :

- i) 출고될 블록들과 함께 재취급 블록들을 적치장에서 꺼낸다.
- ii) 입고될 블록들과 함께 꺼냈던 재취급 블록들을 적치장으로 넣는다.

시작부터 적치장에 있던 초기 블록들의 집합을 B^0 라 하고, 각 초기 블록 $b \in B^0$ 의 최초 저장 위치를 (i_0^b, j_0^b) 라 하자. 초기 블록들은 시작 시점에 적치장의 안쪽부터 빈틈없이 저장되어 있다고 가정한다. 입고되고 출고될 블록들의 집합을 각각 B^{in} 과 B^{out} 이라 하자. 입고 블록 $b \in B^{in}$ 의 가능 입고 기간들의 집합은 I^b , 출고 블록 $b' \in B^{out}$ 의 가능 출고 기간들의 집합은 $O^{b'}$ 이다. 계획 기간 중 입고돼서 출고도 되는 블록의 경우 입출고 가능 기간은 겹치지 않는다; 즉, 모든 블록 $b \in B^{in} \cap B^{out}$ 에 대하여 $\max I^b < \min O^{b'}$ 이다. $B = B^0 \cup B^{in} \cup B^{out}$ 으로 놓자.

문제 정의를 위해 사용된 기호를 요약하면 다음과 같다 :

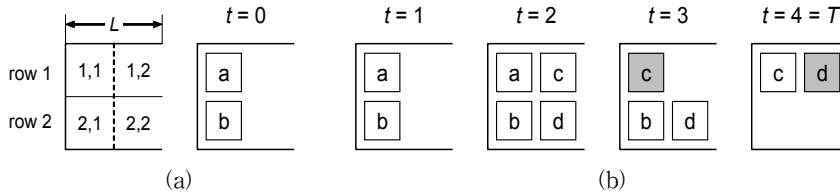
- n : 적치장의 행의 개수
- m : 한 행에 일렬로 적치할 수 있는 블록의 최대 가능 개수
- L : 적치장의 가로(행) 길이
- l_b : 블록 b 의 길이
- (i, j) : 행 i 에서 왼쪽부터 j 번째 위치
- T : 계획 기간의 길이
- B^0 : 초기 블록들의 집합
- (i_0^b, j_0^b) : 초기 블록 $b \in B^0$ 의 최초 저장 위치
- B^{in} : 입고 예정 블록들의 집합

- B^{out} : 출고 예정 블록들의 집합
- I^b : 입고 블록 b 의 가능 입고 기간들의 집합
- $O^{b'}$: 출고 블록 b' 의 가능 출고 기간들의 집합
- $B = B^0 \cup B^{in} \cup B^{out}$

예제 1 : <Figure 1>(a)는 두 행으로 구성된 적치장의 운영 문제의 예를 보여준다. 모든 블록의 길이는 동일하며, 한 행에는 최대 두 개의 블록을 저장할 수 있다($n = m = 2$). 적치장에 저장된 블록은 사각형으로 표시되어 있다. 초기에($t = 0$) 블록 a와 b가 각각 첫 번째 행과 두 번째 행의 가장 안쪽에 저장되어 있다; $B^0 = \{a, b\}$, $(i_0^a, j_0^a) = (1, 1)$, $(i_0^b, j_0^b) = (2, 1)$. 기간 4까지의 계획을 수립하며($T = 4$), 입고 및 출고 예정 블록은 각각 $B^{in} = \{c, d\}$ 와 $B^{out} = \{a, b\}$ 이며, 각 블록의 입출고 가능 기간은 다음과 같다 : $I^c = \{1, 2\}$, $I^d = \{2\}$, $O^a = \{3, 4\}$, $O^b = \{4\}$.

<Figure 1>(b)는 위의 예제에 대해 특정 운영 계획을 실행하는 모습 즉, 각 기간 t 의 끝 시점 곧 $(t+1)$ 의 시작 시점에서의 적치장 상황을 보여준다. 예를 들어, $t = 3$ 인 경우 기간 3에 수행된 작업을 모두 마친 결과이며, 이는 기간 4에서의 작업이 시작되기 전 상황과 같다. 이 계획에서 블록c와 d는 기간 $t = 2$ 에 함께 입고되며, 블록 a와 b는 각각 기간 3과 4에 출고된다. 이는 입출고 가능 기간의 제약을 만족한다.

그림에서 재취급된 블록은 회색으로 표시되어



<Figure 1> An Example of Block Storage Problem, (a) Yard and Initial Blocks, (b) Block Location at the End of Each Period by a Feasible Storage Plan

있다. 예를 들어, 기간 3에 블록 a를 출고하기 위하여 c를 재취급한다. 재취급은 모두 두 번 발생하였으며, 간접 블록이 아닌 경우 재취급하지 않는다는 정책을 고려할 때 이 운영 계획은 최적이다. 참고로, 재취급 블록에 대한 제한이 없으면 더 나은 계획이 가능하다. 즉, 다음과 같이 계획하는 경우 총 재취급 회수는 1이 된다: i) 기간 1에 간접 블록이 아닌 a를 (2, 2)로 재취급하고 블록 c와 d를 각각 (1, 1)과 (1, 2)에 입고함, ii) 기간 4에 a와 b를 출고함.

3. Mathematical Programming Model

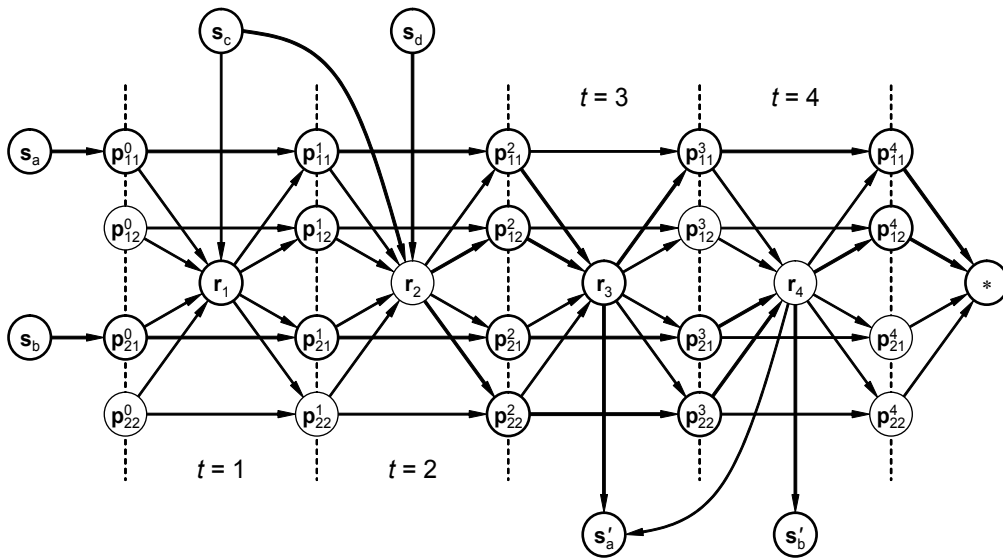
이 장에서는 최적 적치장 운영 계획을 도출하기 위한 정수 계획(integer programming) 모형을 제시한다. 본 모형은 다 품목 네트워크 흐름을 바탕으로 하며, 네트워크 흐름으로 표현하기 힘든 운영 방식을 위해 추가적인 제약을 사용한다.

3.1 Network Flow Model

네트워크 흐름 모형에서 하나의 블록은 하나의

품목에 해당한다. 노드(node)는 블록의 상태와 관련되며 아크(arc)는 블록을 취급하는 활동을 의미한다. 모든 아크는 0 또는 1의 흐름량(flow)을 가진다. 개별 블록에 대한 작업과 블록의 위치 변화는 네트워크상의 흐름으로 나타난다. <Figure 2>는 <Figure 1>의 예제를 네트워크로 표현한 것이다. 앞서 제시된 운영 계획을 나타내는 흐름은 굵은 선으로 표시되어 있다.

블록 b의 입고와 출고를 나타내기 위해 각각 1만큼 유입(inflow)과 유출(outflow)이 있는 노드 s_b 와 s'_b 를 사용한다(예를 들어 <Figure 2>의 s_c 와 s'_d). 노드 p_{ij}^0 는 초기 저장 위치에 대응하며, 기간 $t(\geq 1)$ 에 대하여 노드 p_{ij}^t 는 기간 t 가 끝나는 시점에서의 저장 위치 (i, j) 에 대응한다. 즉, p_{ij}^t 로 유입되는 블록 b의 흐름은 기간 t 의 끝, 곧 기간 $(t+1)$ 의 시작 시점에 b가 위치 (i, j) 에 저장되어 있음을 나타낸다. 한 장소에는 하나의 블록만 보관할 수 있으므로 이들 위치 노드의 용량은 1이다. 계획 시작 시점부터 보관되어 있던 블록은 입고와 동일하게 유입 노드로 표현되어 있으며(예를 들어 s_a), 계획 기간 내에 출고되지 않는 블록의 흐름은 말단



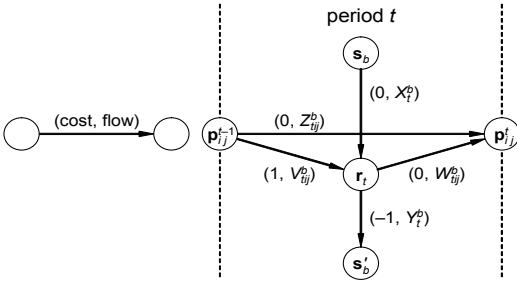
<Figure 2> A Multi-Commodity Network Flow for the Problem in <Figure 1>

노드인 (*)로 향한다.

노드 r_t 로의 흐름은 기간 t 에서 블록의 입고나 출고 또는 재취급의 발생을 의미한다 :

- i) 노드 s_b 에서 r_t 를 거쳐 p_{ij}^t 로 들어가는 블록 b 의 흐름 : b 가 위치 (i, j) 로 입고됨
- ii) 노드 p_{ij}^{t-1} 에서 r_t 를 거쳐 s'_b 로 향하는 블록 b 의 흐름 : 위치 (i, j) 에서 b 가 출고됨
- iii) 노드 p_{ij}^{t-1} 에서 r_t 를 지나 $p_{i'j}^t$ 로 가는 흐름 : 해당 블록이 (i, j) 에서 (i', j) 으로 재취급됨

노드 r_t 를 거치지 않고 p_{ij}^{t-1} 에서 p_{ij}^t 로 직접 향하는 흐름은 (i, j) 에 위치한 블록이 기간 t 에서 움직이지 않고 그대로 있음을 나타낸다.



<Figure 3> The Costs and Variables for Flows of Block b at Period t

각각의 품목별 아크에 대응하는 이진(binary) 결정 변수는 다음과 같다 :

X_t^b : 노드 s_b 에서 r_t 로의 흐름량; 블록 $b \in B^{in}$ 를 기간 $t \in I^b$ 에 입고하는지 여부

Y_t^b : 노드 r_t 에서 s'_b 로의 흐름량; 블록 $b \in B^{out}$ 를 기간 $t \in O^b$ 에 출고하는지 여부

Z_{tij}^b : 노드 p_{ij}^{t-1} 에서 p_{ij}^t 로의 흐름량; 기간 $t \in [1, T]$ 에 블록 b 를 움직이지 않고 위치 (i, j) 에 그대로 두는지 여부

V_{tij}^b : 노드 p_{ij}^{t-1} 에서 r_t 로의 흐름량; 기간 $t \in [1, T]$ 에 블록 b 를 위치 (i, j) 에서 출고 또는 재취급을 위해 움직이는지 여부

W_{tij}^b : 노드 r_t 에서 p_{ij}^t 로의 흐름량; 기간 $t \in [1, T]$ 에 블록 b 를 위치 (i, j) 로 입고 또는 재취급하는지 여부

여기서 $[k, l] = \{k, k+1, \dots, l\}$ 이다. <Figure 3>은 기간 t 에서 블록 b 와 관련된 흐름의 비용과 변수를 보여준다.

본 연구에서는 중간 노드 r_t 를 도입하여 입출고와 위치 노드를 연결한다. 적치장 운영의 목적이 재취급 회수의 최소화이므로 이와 같은 모형화가 가능하다. 만일 중간 노드를 사용하지 않는다면, 입출고와 재취급을 표현하기 위해서는 입출고와 위치 노드의 모든 필요한 쌍을 연결하는 아크가 필요하다. 구체적으로 한 기간에서, 위치 노드를 연결하는 아크의 개수를 n^2m^2 에서 $2nm$ 로 줄일 수 있으며, 특정 블록의 입출고는 nm 개의 아크대신 1개의 아크로 나타낼 수 있다. 그에 따라 관련된 제약식도 단순해지며, 수리 모형의 복잡도가 줄어든다.

3.2 Objective and Constraints

적치장 운영 계획의 목적인 재취급의 최소화는 다음과 같이 표현된다 :

$$\min. \sum_{b \in B} \sum_{t=1}^T \sum_{i=1}^n \sum_{j=1}^m V_{tij}^b - \sum_{b \in B^{out}} \sum_{t \in O^b} Y_t^b \quad (1)$$

네트워크 흐름과 관련된 제약은 흐름 평형(flow balance)과 아크와 노드의 용량 제한이다. 먼저, 흐름 평형 제약은 노드의 유형에 따라 주어진다. 입고와 출고 노드에는 각각 1만큼의 유입과 유출이 있다 :

$$\sum_{t \in I^b} X_t^b = 1, \forall b \in B^{in} \quad (2)$$

$$\sum_{t \in O^b} Y_t^b = 1, \forall b \in B^{out} \quad (3)$$

기간 1에서는 초기 블록을 고려해야 하므로, 초

기 저장 위치에 대응하는 노드 \mathbf{p}_{ij}^0 에서의 흐름 평형은 다음과 같다 : 모든 $b \in B, i \in [1, n], j \in [1, m]$ 에 대하여,

$$Z_{0ij}^b + V_{0ij}^b = \begin{cases} 1, & \text{if } b \in B^0, i = i_0^b, \text{ and } j = j_0^b \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

계획 기간 내의 저장 위치와 관련된 노드 \mathbf{p}_{ij}^t 에 대한 평형은 다음과 같다 : 모든 $b \in B, t \in [1, T-1], i \in [1, n], j \in [1, m]$ 에 대하여,

$$Z_{tij}^b + W_{tij}^b = Z_{(t+1)ij}^b + V_{(t+1)ij}^b \quad (5)$$

개별 블록의 입고 가능 기간과 출고 가능 기간은 겹치지 않으므로, 중간 노드 \mathbf{r}_t 에서의 평형은 다음과 같다 : 모든 $b \in B$ 와 $t \in [1, T]$ 에 대하여,

$$\sum_{i=1}^n \sum_{j=1}^m (V_{tij}^b - W_{tij}^b) = \begin{cases} -X_t^b, & \text{if } b \in B^{\text{in}} \text{ and } t \in I^b \\ Y_t^b, & \text{if } b \in B^{\text{out}} \text{ and } t \in O^b \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

각 위치 노드의 용량 제약은 다음과 같다 :

$$\sum_{b \in B} (Z_{tij}^b + W_{tij}^b) \leq 1, \\ \forall i \in [1, n] \forall j \in [1, m] \forall t \in [1, T] \quad (7)$$

노드의 용량에 제약을 둬으로써 개별 아크 용량에 대한 묶음(bundle) 제약은 따로 두지 않아도 된다. 그리고 마지막 기간 끝 시점에서의 위치 노드에서의 흐름 평형을 고려하지 않아도 무방하다.

네트워크 흐름으로 표현하지 못하는 추가적인 제약이 다음과 같다. 먼저, 한 행에 저장될 수 있는 블록들의 총 길이는 행의 길이를 넘지 못한다 :

$$\sum_{j=1}^m \sum_{b \in B} l_b (Z_{tij}^b + W_{tij}^b) \leq L, \quad \forall i \in [1, n] \forall t \in [1, T] \quad (8)$$

다음으로 간섭 블록은 모두 움직여야 하므로, 특정 기간에 움직이지 않는 블록의 안쪽에는 입고나 출고, 재취급되는 블록이 존재할 수 없다 : 모든 t

$\in [1, T], i \in [1, n], j \in [2, m]$ 에 대하여,

$$\sum_{j=1}^{j-1} \sum_{b \in B} (V_{tij}^b + W_{tij}^b) \leq M \left(1 - \sum_{b \in B} Z_{tij}^b \right) \quad (9)$$

여기서 M 은 충분히 큰 수다. 일반적으로 이와 같은 형태의 제약식에서 작은 M 을 사용하는 것이 문제 해결에 효과적임이 알려져 있다[17]. 제약식 (9)의 좌변의 상한은 주어진 j 에 대하여 $2(j-1)$ 이라는 것을 쉽게 알 수 있다. 따라서 본 연구에서는 M 의 값으로 $2(j-1)$ 을 사용한다.

마지막으로, 출고되는 블록보다 바깥에 위치한 블록만 재취급이 허용된다. 이는 곧 출고되는 블록은 언제나 움직일 수 있지만, 출고되지 않는 경우는 안쪽에 움직이는 블록이 있을 경우에만 움직일 수 있음을 의미한다 : 모든 $b \in B, t \in [1, T], i \in [1, n], j \in [1, m]$ 에 대하여,

$$V_{tij}^b - \sum_{j'=1}^{j-1} \sum_{b' \in B \setminus \{b\}} V_{tij'}^{b'} \leq \begin{cases} Y_t^b, & \text{if } b \in B^{\text{out}} \text{ and } t \in O^b \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

위 제약식에서 좌변의 첫째 항은 블록 b 가 움직이는 여부를 나타내고, 둘째 항은 b 의 안쪽에서 움직이는 블록들의 개수를 나타낸다. 따라서 좌변이 1이 되는 경우, 즉 b 가 움직이는 가장 안쪽 블록이면, b 는 출고되어야 함을 의미한다.

4. Reformulation

본 연구에서 최적 계획을 도출하기 위해 MIP solver를 활용한다. 이 절에서는 적치장 운영 문제의 성질을 고찰하여 최적해를 보장하면서 동시에 MIP solver가 보다 효율적으로 해를 도출할 수 있도록 하는 수정된 모형을 제시한다.

먼저, 입고 및 출고 예정 블록은 각각 입고 후와 출고 전의 기간만 계획에서 고려되면 충분하다. 즉, 모든 입고 블록 $b \in B^{\text{in}}$ 에 대하여 $t \in [1, \min I^b - 1]$ 이면, $Z_{tij}^b = V_{tij}^b = W_{tij}^b = 0$ 이다. 유사하게 모든 출

고 블록 $b' \in B^{out}$ 에 대하여 $t' \in [\max O^{b'}+1, T]$ 이면 $Z_{t'ij}^{b'} = V_{t'ij}^{b'} = W_{t'ij}^{b'} = 0$ 이다. 이와 같이 필요한 결정 변수만 사용하는 수리 모형을 (P1)이라 하자.

다음의 각 절에서는 추가적으로 재취급 가능성과 제약식 대체, 선형 완화를 고려한 수정 모형을 제시한다.

4.1 Arc Elimination(P2)

블록의 재취급은 안쪽에 출고 되는 블록이 있을 경우에만 가능하다. 이와 같은 문제의 특성을 활용하면 네트워크에서 흐름이 발생할 수 없는 아크를 제거하여 문제의 복잡도를 줄일 수 있다.

먼저, 시간 t 에서 특정 초기 블록 $b \in B^0$ 를 고려하자. 만일 b 를 시간 t 까지 출고할 수 없고, b 의 안쪽에도 그때까지 출고할 수 있는 다른 블록이 없다면, 시간 t 에서 b 는 초기저장 위치에서 움직이지 않는다. 즉, 다음의 성질이 존재한다.

성질 1 : 초기 블록 $b \in B^0$ 는 다음 기간 u 까지 움직일 수 없다 :

$$u = \begin{cases} \min \bigcup_{b' \in B'} O^{b'} - 1, & \text{if } B' \neq \emptyset \\ T, & \text{otherwise} \end{cases} \quad (11)$$

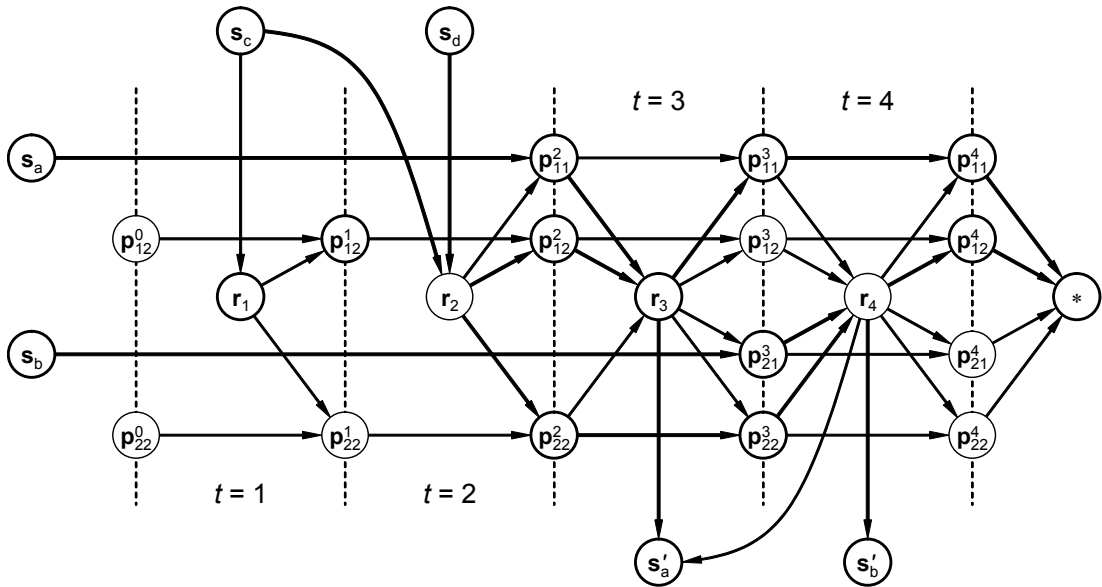
여기서 $B' = \{b' \in B^0 \cap B^{out} | i_0^{b'} = i_0^b, j_0^{b'} \leq j_0^b\}$ 로 b 를 포함하여 b 의 안쪽에 초기 저장되어 있던 블록 중 출고 블록들의 집합이다.

위의 성질에 의해, 그런 블록 b 에 대하여 $t \in [1, u]$ 이면, $Z_{tij}^b = 1$ 로 결정된다. 따라서 흐름 평형과 용량 제한에 의해, 모든 $b' \in B$ 에 대하여 $V_{tij}^{b'} = W_{tij}^{b'} = 0$ 이며, 모든 $b'' \in B \setminus \{b\}$ 에 대하여 $Z_{tij}^{b''} = 0$ 이 된다.

또한 재취급 가능 조건으로부터 다음과 같이 출고 가능 블록이 없는 기간에는 재취급되는 블록도 없음을 쉽게 유추할 수 있다.

성질 2 : 어떤 기간 $t \in [1, T]$ 에 대하여, $t \notin \bigcup_{b \in B^{out}} O^b$ 이면, 모든 $b \in B$ 와 $i \in [1, n], j \in [1, m]$ 에 대하여 $V_{tij}^b = 0$ 이다.

위의 두 성질을 사용하면 일부의 흐름량을 미리



<Figure 4> An Arc-Eliminated Network of <Figure 2>

결정할 수 있다. (P1)에서 그와 같은 결정 변수를 제거한 수리 모형을 (P2)라 한다. 예를 들어 <Figure 4>는 <Figure 2>를 축소된 네트워크를 보여 준다. 성질 1에 의해 행 1과 2에서 각각 기간 2와 3 까지 관련된 아크와 노드가 제거되었으며, 성질 2에 의해 기간 1과 2에서 \mathbf{r}_1 과 \mathbf{r}_2 로 향하는 이동을 나타내는 아크가 삭제되었음을 알 수 있다.

4.2 Constraint Replacement(P3)

모든 결정 변수가 0 또는 1의 값을 가지므로, 식 (9)는 다음으로 대체할 수 있다 :

$$\begin{aligned} \sum_{b \in B} V_{tij'}^b &\leq 1 - \sum_{b \in B} Z_{tij}^b \\ \sum_{b \in B} W_{tij'}^b &\leq 1 - \sum_{b \in B} Z_{tij}^b \end{aligned} \quad \forall j' \in [1, j-1] \quad (12)$$

이와 같은 변경은 MIP solver가 선형 완화(linear relaxation)을 통해 보다 나은 하한(lower bound)을 구할 수 있도록 한다[17]. 참고로 제약식의 개수는 식 (9)를 사용한 경우보다 늘어난다. (P2)에서 위의 제약식들로 대체한 수리 모형을 (P3)라 한다.

식 (9)는 다른 여러 형태로 대체 가능하다. 예를 들어, 다음의 식도 유효한 형태이다 :

$$\begin{aligned} V_{tij'}^b &\leq 1 - \sum_{b' \in B \setminus \{b\}} Z_{tij}^{b'} \\ W_{tij'}^b &\leq 1 - \sum_{b' \in B \setminus \{b\}} Z_{tij}^{b'} \end{aligned} \quad \forall j' \in [1, j-1] \quad \forall b \in B \quad (13)$$

대체하는 식에 따라 제약식 개수의 증가와 하한의 개선 정도가 달라진다. 본 연구에서는 수치 실험을 통해 효과적이라고 판단된 식 (12)를 사용하였다.

4.3 Linear Relaxation(P4)

결정 변수 Z_{tij}^b 와 W_{tij}^b 의 값이 정해지면, 제약식 (5)에 의해 V_{tij}^b 가 결정되며, 제약식 (6)에 의해

X_t^b 와 Y_t^b 역시 결정된다. 따라서 X_t^b , Y_t^b , V_{tij}^b 에 대한 정수 제약이 없어도 이들 결정 변수는 0 또는 1의 값으로 정해진다. 즉 X_t^b , Y_t^b , V_{tij}^b 를 선형으로 완화시켜도 정수 최적해가 보장된다. (P3)를 이와 같이 선형 완화한 모형을 (P4)라 한다.

상용 solver에서는 branch-and-cut 방법을 사용하여 수리 계획을 풀이한다. 정수 제약을 가지는 결정 변수가 다른 경우 도출되는 유효 절단 평면(cut)이 달라져 계산 시간에 영향을 줄 수 있다. 이러한 효과를 살펴보기 위하여 (P4)를 사용하여 정수 제약을 가진 변수의 개수를 줄이는 데 따른 계산 시간의 단축 효과를 확인한다.

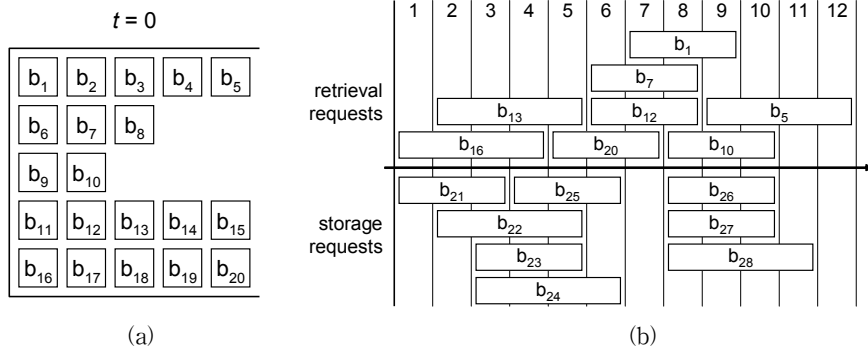
5. Numerical Experiments

본 장에서는 중간 크기의 문제와 실용적인 크기의 문제에 대하여 상용 MIP solver인 ILOG CPLEX 12.4를 사용하여 최적해를 구하고 제안한 수리 모형 (P1)에서 (P4)의 성능을 확인한다. 모든 실험은 Intel Core i7-3610QM 2.3GHz와 16GB RAM을 가지는 하드웨어와 Windows 7(64bit)에서 수행되었다.

5.1 Medium-Size Problem

수치 실험에서 사용한 중간 크기의 문제는 다음과 같다.

예제 2 : 적치장은 다섯 행으로 구성되며, 각 행에는 최대 5개의 블록을 저장할 수 있다. 모두 20개의 초기 블록이 있으며, 각각의 초기 저장 위치는 <Figure 5>(a)와 같다. 입출고 요구 블록은 각각 8개씩이며, 입출고 가능 기간은 (b)에 표시되어 있다. 예를 들어 블록 b_1 은 7부터 9까지의 기간에서 출고해야 하며, 블록 b_{21} 은 1부터 3까지의 기간에서 입고되어야 한다. 모든 블록의 길이는 동일하다고 가정한다. 총 계획 기간 12이다.



<Figure 5> A 5x5-Storage Yard Example, (a) the Initial Block Location, (b) Retrieval and Storage Requests

<Table 1> Experimental results of Example 2

Model	IntVars	Cons	Time
(P1)	20,605	16,080	536.4
(P2)	10,338	9,382	17.9
(P3)	10,338	10,090	13.0
(P4)	6,856	10,090	12.5

<Table 1>은 위의 예제에 대하여 각 수리 모형별 정수 결정 변수의 개수(IntVars)와 제약식의 개수(Cons), 최적해를 구하는 시간(Time in second)

<Table 2> Optimal Solution of Example 2

Period	Retrieval	Storage	Relocation
1	b_{16}		$b_{17}(3, 3)$, $b_{18}(2, 4)$, $b_{19}(3, 4)$, $b_{20}(2, 5)$
2		$b_{21}(5, 1)$	
3	b_{13}		$b_{14}(5, 2)$, $b_{15}(5, 3)$
4		$b_{22}(5, 4)$, $b_{23}(5, 5)$	
5			
6	b_{12} , b_{20}	$b_{24}(4, 2)$, $b_{25}(4, 3)$	
7	b_7		$b_8(4, 4)$, $b_{18}(4, 5)$
8		$b_{26}(2, 2)$	
9	b_1 , b_5		$b_2(2, 3)$, $b_3(2, 4)$, $b_4(2, 5)$
10	b_{10}	$b_{27}(3, 2)$	$b_{17}(1, 1)$, $b_{19}(1, 2)$
11		$b_{28}(3, 3)$	
12			

을 보여준다. 기본모형 (P1)과 비교하여 (P2)에서 결정 변수와 제약식이 줄어들며, (P3)에서는 제약식 대체로 인해 개수가 (P2)에 비해 약 8% 증가한 것을 알 수 있다. (P3)와 비교하여 (P4)는 전체 결정 변수의 개수는 동일하나, 정수 변수의 개수가 약 34% 줄어들었다. 모형의 변화에 따라 계산 시간이 감소하는 것을 알 수 있으며, 재취급 가능성을 고려하는 (P2)의 경우가 이 문제에 대하여 특히 효과적인 것으로 나타났다. (P4)의 계산 시간은 (P1)에 비하여 약 1/3로 줄어들었다.

이 문제에 대한 최소 재취급 회수는 13이며, <Table 2>는 최적 운영 계획을 보여준다. 표에서 입고와 재취급되는 블록은 목표 저장 위치가 같이 표시되어 있다. 예를 들어 기간 1에 b_{17} 은 b_{16} 을 출고하기 위해 재취급되어 (3, 3)의 위치로 옮겨진다.

5.2 Practical Size Problem

국내 한 조선소의 적치장 운영과 유사한 복잡도를 가지는 문제를 임의로 생성하여 각 수리 모형의 성능을 검토하였다.

적치장은 13개의 행으로 이루어져 있으며, 각 행은 최대 7개의 블록을 저장할 수 있다. 총 계획 기간은 12시간이며, 1시간의 단위 기간을 사용한다. 입고와 출고 요청 블록은 각각 20개씩이며, 입출고 가능 기간은 연속된 세 기간 또는 네 기간으로 임의로 정한다. 입고된 블록들 중 계획 기간에 다시

출고되는 블록은 없다. 초기 블록의 위치는 출고 요청 블록들과 독립적으로 임의로 생성된다. 모든 블록의 길이는 동일하다고 가정한다.

적치 능력(이 경우 91개) 대비 계획 기간의 시작 시점에 적치되어 있는 블록의 개수를 초기 적치율이라 하자. 본 연구에서는 초기 적치율을 30%에서 90%까지 변화시켜가며 CPLEX로 최적 계획을 도출하는 데 걸리는 시간을 파악하였다. 각 초기 적치율에 대하여 10개의 문제를 임의로 생성하여 사용하였으며, 계산 시간의 한계는 3,600초로 설정하였다.

<Table 3>은 각 모형으로 한계 시간 안에 해결한 문제의 개수, 즉, 최적해를 구한 문제의 개수를 보여준다. 60%까지의 초기 적치율을 가지는 문제들은 모든 모형으로 해결할 수 있으나, 그 이상의 경우 (P3)와 (P4)만 의미 있는 결과를 보여주었다. 초기 적치율이 80%까지인 경우 두 모형을 사용하여 모두 모든 문제를 해결할 수 있었으나, 90%인 경우 각각 8문제씩만 해결할 수 있었다. 각각의 모형으로 해결하지 못한 문제들 중 한 문제는 동일하였다. 즉, 두 모형이 공통으로 해결한 문제는 7개이다. 참고로 해당 조선소에서는 보통 70%에서 80%의 적치율로 운영된다.

<Table 3> Number of Problem Instances Solved by Each Model(10 Instances in Total)

Utilization	30%	40%	50%	60%	70%	80%	90%
(P1)	10	10	10	10	8	3	3
(P2)	10	10	10	10	8	4	4
(P3)	10	10	10	10	10	10	8
(P4)	10	10	10	10	10	10	8

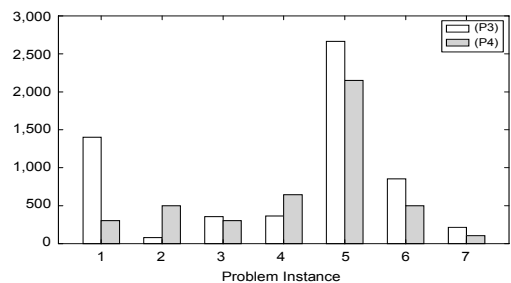
<Table 4>는 (P2)와 (P3), (P4)에 대하여 초기 적치율 80%까지 계산 시간의 평균(Avg)과 최대(Min), 최소(Max)를 보여준다. (P2)의 경우 모든 문제에 대하여 최적해를 구할 수 있었던 60%까지만 계산 시간을 표시하였다. 중간 크기의 문제인 예제 2와 비교할 때, 제약식 (9)를 대체한 (P3)의 효과가 분명함을 볼 수 있다. 세 모형 모두 60%부터 계산

시간이 뚜렷이 증가하며, 증가 정도는 (P3)가 작은 것을 확인할 수 있다. 이 결과는 대상 문제들의 경우 선형 완화가 최적해를 구하는데 도움이 되지 않음을 보여준다.

<Table 4> Computation Time in Sec Using(P3) and (P4)

Utilization	30%	40%	50%	60%	70%	80%	
Avg	(P2)	19	29	112	659		
	(P3)	26	27	32	87	159	359
	(P4)	22	24	29	83	318	516
Min	(P2)	12	15	19	26		
	(P3)	26	27	32	87	159	359
	(P4)	15	19	19	21	19	78
Max	(P2)	38	65	474	2786		
	(P3)	54	45	60	252	480	840
	(P4)	34	32	37	408	1098	1201

<Figure 6>은 90%의 초기 적치율에서 (P3)와 (P4)가 공통으로 해결한 7개의 문제에 대한 계산 시간을 보여준다. 이 7개의 대상 문제에 대해서도 선형 완화가 계산 시간 단축에 도움이 되지 않음을 알 수 있다.



<Figure 6> Computation Time in Sec for 7 Instances with 90% Utilization

6. Conclusions

본 연구는 주어진 생산 일정 계획 하에서의 블록 적치장 운영 문제를 다루었다. 네트워크 흐름을 기반으로 기본적인 수리 모형을 제시했으며, 목적 함수를 고려하여 복잡도를 낮춘 네트워크를 도출

하였다. 또한 문제의 특성을 고려하여 기본 모형을 변경한 수정 모형들을 개발하였다. 수정 모형의 도출을 위해 재취급 가능성과 제약식 대체, 선형 완화를 고려하였다. 그리고 ILOG CLPEX를 사용하여 최적해를 도출하고 모형별 성능을 검토하였으며, 실제 운영에의 적용 가능성을 살펴보았다. 주요한 추후 연구 방향은 다음과 같다.

우리는 적치장의 공간 효율성을 고려하여 한쪽 방향으로만 블록을 입출고하는 적치장을 대상으로 하였다. 그리고 안전을 위해 직선 이동이 선호되는 블록 이송 차량을 고려하여 재취급 블록을 모두 적치장 밖으로 꺼낸 후 다시 넣은 운영 방식을 가정하였다. 적용 범위를 넓히기 위해서는 여러 방향에서의 블록 입출고와 다른 운영 환경을 고려하는 연구가 필요하다. 또한 블록의 각기 다른 형태와 크기를 보다 상세히 고려하는 것도 필요할 것이다.

재취급 블록들은 적치장에 다시 넣기 전 적치장 주변에 임시 보관한다. 하지만 한 번에 너무 많은 재취급 블록이 발생하면, 임시 저장 공간이 부족해질 수 있다. 본 연구에서는 이와 같은 임시 저장 공간의 크기를 고려하지 않았으며, 보다 일반적인 운영 환경에 적용하기 위해서는 임시 공간의 능력을 고려한 연구가 필요하다.

조선 산업에서 블록 물류의 효율성은 생산성에 큰 영향을 미친다. 효율적인 블록 물류계획 수립을 위해서는 본 연구의 대상인 적치 계획과 함께 이송 계획도 수립되어야 한다. 이송 계획에서는 블록의 중량과 시간 제약을 고려하여 이송 장비를 할당하고 처리 순서를 결정한다. 따라서 적치 계획과 이송 계획이 통합되어 운영되는 경우 조선소의 생산성 향상에 크게 기여할 것이다. 나아가 생산 일정 계획의 수립 시 적치 및 이송 계획이 동시에 고려된다면 더 큰 효과를 가져 올 것으로 기대된다.

참 고 문 헌

- [1] Caserta, M., S. Voß, and M. Sniedovich, "Applying the corridor method to a blocks re-location problem," *OR Spectrum*, Vol.33, No.4 (2011), pp.915-929.
- [2] Cho, K.K., K.H. Chung, C. Park, J.C. Park, and H.S. Kim, "A spatial scheduling system for block painting process in shipbuilding," *CIRP Annals*, Vol.50, No.1(2001), pp.339-342.
- [3] Ha, B.-H. and S. Kim, "A* algorithm for optimal intra-bay container premarshalling plan," *Journal of the Korean Institute of Industrial Engineers*, Vol.38, No.2(2012), pp.157-172.
- [4] Kim, K.H. and G.-P. Hong, "A heuristic rule for relocating blocks," *Computers and Operations Research*, Vol.33, No.4(2006), pp.940-954.
- [5] Koh, S.G., "A production schedule with genetic algorithm in block assembly shop," *Korean Management Science Review*, Vol.13, No.1(1996), pp.1-12.
- [6] Koh, S.-G., J.-C. Park, Y.-S. Chon, and C.-M. Joo, "Development of a block assembly scheduling system for shipbuilding company," *IE Interfaces*, Vol.12, No.4(1999), pp.586-594.
- [7] Lee, J.K., K.J. Lee, H.K. Park, J.S. Hong, and J.S. Lee, "Developing scheduling systems for Daewoo Shipbuilding DAS project," *European Journal of Operational Research*, Vol.97, No.2(1997), pp.380-395.
- [8] Lee, K.J., J.K. Lee, and S.Y. Choi, "A spatial scheduling system and its application to shipbuilding DAS-CURVE," *Expert Systems With Applications*, Vol.10, No.3/4(1996), pp.311-324.
- [9] Lee, S., J. Kim, and I. Moon, "Deployment planning of blocks from storage yards using a tabu search algorithm," *Journal of the Korean Institute of Industrial Engineers*, Vol.37, No.3(2011), pp.198-208.

- [10] Lee, Y. and Y.-J. Lee, "A heuristic for retrieving containers from a yard," *Computers and Operations Research*, Vol.37(2010), pp. 1139-1147.
- [11] Park, C. and J. Seo, "A case study on assembly block operations management at shipyard," *Korean Management Science Review*, Vol.23, No.2(2006), pp.175-185.
- [12] Park, C. and J. Seo, "Genetic algorithm of the planar storage location assignment problem," *Journal of the Korean Institute of Industrial Engineers*, Vol.35, No.2(2009), pp. 129-140.
- [13] Park, C. and J. Seo, "Mathematical modeling and solving procedure of the planar storage location assignment problem," *Computers and Industrial Engineering*, Vol.57, No.3 (2009), pp.1062-1071.
- [14] Park, C. and J. Seo, "Comparing heuristic algorithms of the planar storage location assignment problem," *Transportation Research Part E*, Vol.46, No.1(2010), pp.171-185.
- [15] Park, K., K. Lee, S. Park, and S. Kim, "Modeling and solving the spatial block scheduling problem in a shipbuilding company," *Computers and Industrial Engineering*, Vol. 30, No.3(1996), pp.357-364.
- [16] Park, M.-H., W.-S. Lee, Y.-S. Ock, and T.-E. Lee, "A review of Korean shipbuilding industry and engineering research," *IE Interfaces*, Vol.8, No.2(1995), pp.5-20.
- [17] Pochet, Y. and L.A. Wolsey, *Production Planning by Mixed Integer Programming*. Springer, New York, 2006.
- [18] Son, J.-R. and K.-K. Cho, "Design for block logistics system of storage location assignment in shipbuilding," In *Proceedings of the 12th Annual International Conference on Industrial Engineering : Theory, Applications and Practice*, Cancun, Mexico, 2007.
- [19] Wan, Y., J. Liu, and P.-C. Tsai, "The assignment of storage locations to containers for a container stack," *Naval Research Logistics*, Vol.56(2009), pp.699-713.
- [20] Won, S.H. and K.H. Kim, "Allocating storage spaces for temporary inventories considering handling, transportation, and storage capacities," *Journal of the Korean Operations Research and Management Science Society*, Vol.31, No.3(2006), pp.11-25.