

컨테이너 터미널에서 장치장 현황정보를 고려한 안벽크레인 일정계획 방법

A Quay Crane Scheduling Method Considering Congestions in the Yard of Container Terminals

정다훈*, 박영만** 이병권*, 류광렬***, 김갑환*

- * 부산대학교 산업공학과
- ** 해군사관학교 경영과학과
- *** 부산대학교 컴퓨터공학과

{youngu, ymanpark, errorplus, krryu, kapkim}@pusan.ac.kr,

Abstract

컨테이너 터미널에서의 생산성은 본선작업의 효율성과 밀접한 관련이 있다. 효율적인 본선작업을 위한 첫 번째 단계는 선석에서 컨테이너를 취급하는 장비인 안벽크레인(Quay Crane : QC)에 대하여 효율적인 작업일정계획을 수립하는 것이다. 일반적으로 안벽크레인의 작업일정계획은 선박내의 장치계획에 관한 정보와 장치장의 장치현황정보등 많은 정보를 활용해야 하며, 선박의 안정성과 크레인의 작업용이성 등 다양한 제약사항을 고려하여야 한다. 따라서 본 연구에서는 인공지능기법 중 최적에 가까운 해를 찾아주는 탐색기법인 GRASP(Greedy Randomized Adaptive Search Procedure)를 적용하여 실제 현장에서 적용 가능한 안벽크레인 작업일정계획기법을 제안하였다. 또한, 컨테이너 터미널의 실제 자료를 수집하여 제안된 메타휴리스틱 알고리즘의 성능을 분석하였다.

1. 서론

컨테이너 터미널은 하역, 보관, 환적, 운송이 동시에 이루어지는 대표적인 종합 물류 기지라고 할 수 있다. 최근 컨테이너 선박의 대형화는 물동량을 크게 증가 시켰으며, 이로 인해 항만 간에 선사 및 화물 유치경쟁이 치열하게 전개되고 있다. 이와 같은 항만 경쟁에서 앞서 나가기 위해서 항만운영자들은 컨테이너 터미널의 생산성을 향상시키기 위하여 노력하고 있다.

본 연구에서 다루는 컨테이너 터미널은 선석, 장치장, 게이트로 그 영역이 나뉜다. 그림 1에서 보는 바와 같이, 컨테이너 터미널에서의 작업은 선박에 실려 있는 수입컨테이너를 내려서 장치장에 적재하는 양하작업과 장치장에 적재되어있는 수출컨테이너를 선박에 적재하는 적하작업, 장치장에 적재되어있는 수입 컨테이너를 외부트럭이 게이트를 통해 인출하는 반출작업, 게이트를 통해 외부트럭으로부터 수출 컨테이너를 받아 장치장에 적재하는 반입 작업으로 구성된다. 선박과 관련된 양

하작업을 흔히 본선작업이라 한다.

선박이 컨테이너 터미널에 머무르는 평균시간인, 선박체류시간은 컨테이너 터미널의 생산성을 측정하는 중요한 요소 중 하나이다. 선박체류시간의 많은 부분은 양·적하작업시간에 의해 결정된다. 따라서 양·적하작업시간을 줄임으로써 선박체류시간이 줄어들고, 컨테이너 터미널의 생산성을 향상시킬 수 있다. 효율적인 본선작업을 위한 첫 번째 단계는 선석에서 컨테이너를 취급하는 장비인 안벽크레인에 대하여 효율적인 작업일정계획을 수립하는 것이다.

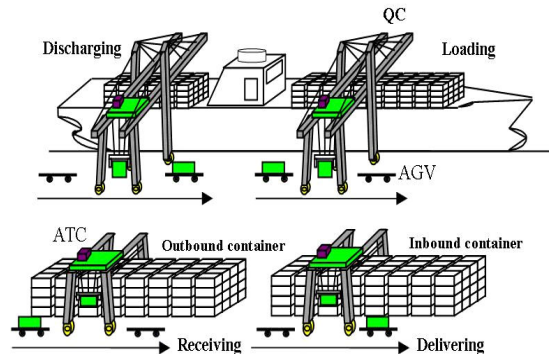


그림1. 컨테이너 터미널에서의 작업

안벽크레인 일정계획은 선박의 접안시간과 양·적하작업물량이 주어졌을 때, 선박의 선적지시서 (Stowage Plan)와 야드맵(Yard Map)을 바탕으로 안벽크레인의 가용시간을 고려하여 효율적인 안벽크레인의 작업일정계획을 수립하는 것이다. 컨테이너 터미널에 접안하는 선박마다 안정성에 대한 고려정도, 안벽크레인간의 간섭, 장치장 현황등 다양한 고려사항으로 인하여 효율적인 일정계획을 작성하는데 어려움이 있다.

본선작업시에 트랜스퍼크레인은 안벽크레인의 작업지연이 발생하지 않도록 작업순서를 결정해야 한다. 그러나 트랜스퍼크레인의 작업순서결정은 이

문제자체로서도 해결하기 어려운 문제이다. 지금까지의 연구들은 대부분 안벽크레인 일정계획시 장치장의 장치현황 및 작업부하정보, 트랜스퍼크레인의 작업순서 및 간섭현상등을 고려하지 못하였다. 본 연구에서는 트랜스퍼크레인의 작업순서결정은 장치현황정보를 이용한 Greedy 휴리스틱을 사용하였으며 이를 바탕으로 작업부하 및 트랜스퍼크레인간의 간섭을 고려하여 안벽크레인 일정계획을 수립하였다.

Daganzo (1989)는 여러 대의 선박의 작업을 수행하기 위한 안벽크레인 일정계획문제를 처음으로 다루었다. 또한, Peterkofsky와 Daganzo (1990)는 여러 대의 선박에 대하여 출항시간과 안벽크레인 할당 수를 결정하는 알고리즘을 제시하여 선박의 지연을 최소화하는 문제를 다루었다. 박영만과 김갑환 (2004)은 안벽크레인 일정계획문제를 메타 휴리스틱기법인 GRASP와 Branch and Bound 기법을 적용하여 문제를 해결하였으나, 장치장현황은 이미 결정 되어 있다고 가정하고 선박의 물량만을 고려하였다. 김갑환과 강진수(2004)는 안벽크레인 스케줄과 야드맵을 바탕으로 개별 컨테이너의 적하작업순서를 결정하기 위해 빔 서치기법을 적용하여 문제를 해결 하였다. 김갑환과 김기영(1999)은 트랜스퍼크레인의 작업순서를 결정하는 문제를 다루었다.

본 연구의 구성은 다음과 같다. 2장에서는 안벽크레인의 일정계획문제를 설명하였으며, 3장에서는 장치장의 현황정보가 고려된 안벽크레인 일정계획문제를 해결하기 위해서 GRASP를 적용한 알고리즘을 제시하였다. 4장에서는 수치 실험을 수행하고, 분석결과를 제시하였다. 마지막으로 5장에서는 결론 및 향후 연구방향을 제시하였다.

2. 안벽크레인 일정계획 문제

안벽크레인 일정계획문제의 목표는 본선작업의 완료시간을 최소화하기 위해서 각 안벽크레인의 양적하작업의 순서를 결정하는 것이다. 안벽크레인 일정계획은 선사로부터 제공되는 선적지시서(Stowage Plan)와 장치장에서 컨테이너들의 장치현황을 보여주는 야드맵(Yard Map)의 정보를 이용하여 수립한다. 그림 2와 그림 3은 선박에 대한 선적지시서와 야드맵을 보여주고 있다.

선적지시서는 선박내 각 베이의 횡단면에 여러 개의 색깔로 표시된 cell들의 집합으로 구성되어 있다. 각 cell에는 양적하 되어야 할 컨테이너의 속성 즉, 목적항, 크기, 형태등을 나타내고 있다. 이러한 속성들을 가진 컨테이너들의 집합을 "Group"이라고 한다. 야드맵은 장치장에서 컨테이너들의 Group별 장치현황정보를 보여준다. 보통 선적지시서와 야드맵에는 동일한 속성을 가지는 컨테이너들이 서로 인접한 slot에 위치되어 있는데 이는 양적하작업의 효율성을 위한 것이다.

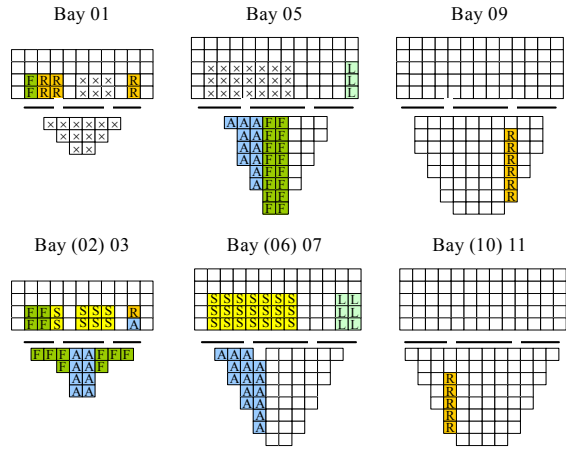


그림2. 선박의 Stowage Plan

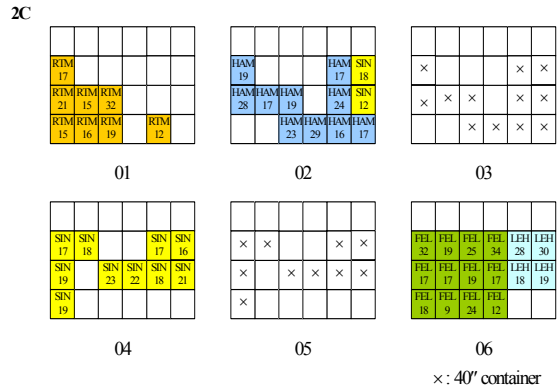


그림3. 야드맵에서 컨테이너의 분포

본 연구에서는 선박과 장치장의 컨테이너들에 대하여 선박클러스터(Ship-cluster)와 장치장클러스터(Yard-cluster)를 생성하여 이들의 작업순서를 결정함으로써 안벽 및 트랜스퍼크레인의 작업일정계획을 수립하였다. 여기서 클러스터란 동일한 베이에 속성이 같은 동일 group 컨테이너들의 묶음을 말한다. 선박클러스터는 선적지시서를 바탕으로 동일한 Ship-bay, Hold 또는 Deck에 위치하고, 양하 또는 적하, 목적항, 크기, 형태 등의 동일한 속성을 가진 컨테이너들의 묶음이다. 이러한 선박클러스터는 안벽크레인의 작업계획단위라고 할 수 있다. 장치장클러스터는 야드맵에서 동일선박을 대상으로 동일한 Yard-bay에 위치하고, 양하 또는 적하, 목적항, 크기, 형태 등의 동일한 속성을 가진 컨테이너들의 묶음이다.

안벽크레인 일정계획문제는 몇 가지 특징이 있다. 먼저, 보통 양적하작업을 위해서 다수의 안벽크레인이 동시에 작업을 수행한다. 이러한 경우는 일정계획분야에서 동일한 기능을 가진 m 대의 기계가 동시에 작업을 수행하는 경우와 일치하므로 m Parallel Machine 일정계획문제의 한 형태라고 할 수 있다. 뿐만 아니라 동일한 Ship-bay에서 양적하작업이 수행되어야만 할 때, 양하작업이 적하작업보다 먼저 수행 되어야만 한다. 적하작업의 경우, 동일한 작업위치에서는 Hold의 작업을 먼저 수행하

안벽크레인 일정 계획

QC 1 (작업 시간: 09:00~10:25)							QC 2 (작업 시간: 09:00~11:10)						
작업 순서	선박 클러스터	컨테이너 개수	작업시간	장치장 클러스터	컨테이너 개수	작업시간	작업 순서	선박 클러스터	컨테이너 개수	작업시간	장치장 클러스터	컨테이너 개수	작업시간
1	1	8	9:00~9:20	2	8	9:00~9:20	1	5	35	9:20~10:55	8	10	9:20~9:45
											9	5	9:45~10:00
											10	5	10:00~10:15
2	2	10	9:20~9:40	3	10	9:20~9:40	2	4	5	10:55~11:10	5	10	10:15~10:40
											4	5	10:40~10:55
											2	2	10:55~11:02
3	3	10	9:40~10:25	1	10	9:40~10:00	6	3	3	11:02~11:10	7	10	10:00~10:25
				7	10	10:00~10:25							

그림4. 안벽크레인 일정계획의 예

고 Deck의 작업을 수행하여야 한다. 반대로 양하작업인 경우, 동일한 작업위치에서는 Deck의 작업을 먼저 수행하고 Hold의 작업을 수행하여야 한다. 이처럼 안벽크레인 일정계획문제는 일반적인 일정계획문제에서 작업대상들간의 선후행 관계가 있는 경우이다. 또한 작업을 수행하는 장비인 안벽크레인이 이동하면서 작업을 수행하기 때문에 각 안벽크레인간의 간섭현상이 발생하지 않도록 해야 한다. 즉, 안벽크레인의 작업 대상인 선박클러스터가 인접했을 때 서로 다른 안벽크레인이 동시에 작업을 수행할 수 없다. 따라서 안벽크레인 간에는 약간의 거리를 두고 작업을 수행 해야만 한다. 따라서 본 연구에서 다루는 안벽크레인 일정계획문제는 작업간의 선후행관계가 있고 작업을 수행하는 기계간 간섭이 있는 *m Parallel Machine* 일정계획문제의 한 형태로 정의할 수 있다.

본 연구에서 안벽크레인 일정계획문제를 해결하는데 있어서 고려된 제약조건은 다음과 같다.

- (1) 안벽크레인의 가용한 시간 내에서 작업을 수행하여야 한다.
- (2) 선박내의 인접한 작업이나 특정한 작업에 대해서는 동시에 작업이 이루어질 수 없다.
- (3) 장치장의 장치현황에 따라서 특정한 작업은 동시에 작업이 이루어질 수 없다.
- (4) 작업을 수행하는 안벽크레인간의 간섭이 발생하지 않아야 한다.
- (5) 특정한 작업 간에는 선후행관계를 만족시켜야 한다.
- (6) 장치장에서의 트랜스퍼크레인 이동 및 간섭 등 작업부하를 고려하여야 한다.

위에서 제시한 제약조건에서 알 수 있듯이, 안벽크레인의 작업일정계획문제는 선박의 특징, 장치장 현황, 취급장비의 특징등 다양한 고려사항이 존재한다. 기존에는 안벽크레인 일정계획을 수행함에 있어 선박 및 안벽크레인의 간섭등은 반영된 반면 장치장 현황정보 및 트랜스퍼크레인의 이동및 간섭은 충분히 반영이 되지 않았다. 안벽크레인의 작업시간은 장치장에서 작업하는 트랜스퍼 크레인의 작업시간에 많은 영향을 받는다. 본 연구에서는 안벽크레인일정계획 수립시 장치장 작업현황 정보를 적용하기위해서, 안벽크레인이 작업할 하나의 선박클러스터를 결정할 때 장치장에서 선박클러스터의 대상이 되는 장치장클러스터를 결정하여 트랜스퍼크레인의 간섭등 작업시간을 고려한 안벽 크레인의

총 작업시간을 계산 하였다.

그림 4는 본 연구에서 작성한 안벽크레인 작업 일정계획표이다. 그림 4에서는 안벽크레인파 트랜스퍼크레인이 작업을 수행할 선박클러스터와 장치장클러스터를 나열하고 있으며, 각 클러스터에 포함된 컨테이너 개수, 안벽크레인파 트랜스퍼크레인의 작업 시간을 보여주고 있다.

다음 장에서는 인공지능기법 중 최적에 가까운 해를 찾아주는 탐색기법인 GRASP(Greedy Randomized Adaptive Search Procedure)를 적용하여 실제 현장에서 적용 가능한 안벽크레인 작업일정계획기법을 제안하였다.

3. GRASP를 이용한 안벽크레인 일정 계획

GRASP(Greedy Randomized Adaptive Search Procedure)를 통한 최적해 탐색기법은 1980년대 후반에 set covering problem등 계산복잡도가 높은 문제를 풀기 위한 해법으로 처음 개발 되었다. GRASP는 반복적인 임의추출기법으로써 각 반복단계에서 하나의 해를 제공한다. 모든 GRASP 반복과정 후 유지되는 해가 최종해가 된다. GRASP는 해를 생성하는 첫 번째 단계(Solution Construction Phase)와 생성된 해를 개선시키는 두 번째 단계(Solution Improvement Phase)로 구성되어있다. 그림 5는 본 연구에서 제시하는 안벽크레인 일정계획 알고리즘의 pseudo-code이다.

```

procedure Grasp()
1  InputInstance();
2  Repeat
3      Phase 1: ConstructInitialSolution(Solution);
4      Phase 2: LocalSearch(Solution);
5      UpdateSolution(Solution, BestSolution);
6  Until stopping criterion = true
7  return(BestSolution);
end grasp
    
```

그림5. 전체 알고리즘의 pseudo-code

그림 5의 첫 번째 행은 초기에 문제를 정의하는 부분으로서 필요한 변수의 선언 및 해를 초기화하는 부분이다. 두 번째 행부터 여섯 번째 행까지가 GRASP 기법의 주 내용이다. 즉, 안벽크레인이 작업 가능한 작업을 나열하여 Greedy function과 난수를 이용하여 선택클러스터를 선택하고, 선택된 선택클러스터와 동일한 그룹속성을 가진 장치장클러스터 중에서 Greedy 휴리스틱을 이용해서 트랜스퍼크레인의 작업을 선택하는 방법으로 모든 선택클러스터와 장치장클러스터의 작업순서를 결정한다. 네 번째 행과 다섯 번째 행에서 지역적 탐색기법을 이용하여 해를 개선시킨 다음 기존의 좋은 해와 비교하여 최선의 해를 찾는다. 마지막 행에서 가장 좋은 해를 되돌려주는 형태를 가진다. 안벽크레인 일 정계획에서 사용한 GRASP 탐색의 종료 조건은 최대반복수행 횟수 및 해의 개선이 연속해서 특정한 횟수만큼 개선이 없는 경우(MNI)에 종료하며, 그때 까지 가장 좋은 해를 선택한다.

각 단계별 세부적인 해법 절차를 설명하기 위해 다음과 같은 기호를 사용하였다.

- i QC 및 TC의 색인($i = 1, 2, \dots, T$)
- Q_i QC_i 의 작업 완료시간
- T_i TC_i 의 작업 완료시간
- F_i QC_i 의 현재 위치
- $f_i^{(BD)}$ TC_i 의 현재 위치(블럭, 베이)
- l_j 선택클러스터 j 의 위치
- $y_k^{(BD)}$ 장치장클러스터 k 의 위치
- SC_j 선택클러스터 j 의 컨테이너 개수
- YC_k 장치장클러스터 k 의 컨테이너 개수
- p_j 선택클러스터 j 의 처리시간
- Ω 전체 선택클러스터들의 집합
- Ψ 동시에 수행될 수 없는 선택클러스터들의 집합
- Φ 선택클러스터의 선후행관계를 나타내는 집합
- Π 전체 장치장클러스터들의 집합

3.1 해의 생성 단계 (Phase 1)

해의 생성단계의 단계별 세부적 절차는 다음과 같다.

단계1. 안벽크레인 중에서 최소 Q_i 를 가지는 안벽크레인을 선택한다. 만약 안벽크레인의 Q_i 가 동일한 경우 임의적으로 안벽크레인을 선택한다. QC_i 가 선택되었다고 하자.

단계2. 선택된 QC_i 가 작업할 수 있는 작업가능목록을 작성한다. 이를 집합 F 라고 하자. 작업의 선후행 관계 및 안벽크레인간 간섭등 제약조건에 해당하는 선택클러스터들은 F 에서 제거된다. 또한 선택클러스터 $j \in F$ 가 다음 조건을 만족하면 F 에서 제거된다. $1/|I_i^c - I_j| < r \max 1/|I_i^c - I_l|, l \in F$, 여기서 r 은 0과 1사이의 상수이다.

단계3. 나열된 선택클러스터 $j \in F$ 중에서 QC_i 위치와 가까운 위치의 작업이 선택될 확률이 크도록 다음과 같은 확률값을 부여한다.

$$(1/|I_i^c - I_j|) / \sum_{l \in F} (1/|I_i^c - I_l|)$$

단계4. 난수를 발생시켜 QC_i 가 작업할 하나의 선택클러스터 $j^* \in F$ 를 선택한다.

단계5. 선택된 선택클러스터와 동일한 그룹정보를 가진 장치장클러스터들 중에서 TC_i 의 작업을 선택한다. 세부절차는 다음과 같다.

단계5-1. 단계4에서 선택된 선택클러스터 $j^* \in F$ 와 동일한 그룹정보를 가진 장치장클러스터들의 작업후보목록을 작성한다. 작업후보목록을 집합 G_{j^*} 라고 하자.

단계5-2. 장치장클러스터 $k \in G_{j^*}$ 중에서 TC_i 의 현재 위치와의 거리가 가장 짧은 장치장클러스터 k^* 를 선택한다. 선택된 장치장클러스터 k^* 를 집합 S_{j^*} 에 포함시킨다. S_{j^*} 를 선택클러스터 j^* 에 할당된 장치장클러스터들의 집합이라 하자.

단계5-3. TC_i 의 현재 위치를 장치장클러스터 k^* 의 위치 $y_{k^*}^{(BD)}$ 로 설정한다. 선택된 장치장클러스터 k^* 가 가진 컨테이너개수 YC_{k^*} 는 선택클러스터 j^* 가 요구한 컨테이너개수 SC_{j^*} 를 만족 시켜야한다. 즉, 조건 $YC_{k^*} \geq SC_{j^*}$ 을 만족하면 $YC_{k^*} = YC_{k^*} - SC_{j^*}$ 으로 변경하고 단계6으로 진행한다. 만일 조건을 만족하지 않으면 장치장클러스터 k^* 를 G_{j^*} 에서 제거하고, $SC_{j^*} = SC_{j^*} - YC_{k^*}, YC_{k^*} = 0$ 으로 변경한 다음 단계5.2로 간다.

단계6. 장치장부하정보를 수집하고, 수집된 정보를 이용하여 트랜스퍼크레인의 이동시간, 작업시간, 간섭을 고려하여 TC_i 의 총 작업완료시간 T_i 을 계산한다. 그림 6은 트랜스퍼크레인간 간섭을 보여주고 있다. 즉, 그림6에서 TC_2 의 작업이 현재 진행 중인 TC_1 의 작업위치나 그 주변에 있게 될 경우 트랜스퍼크레인간의 간섭이 발생하게 된다. 이러한 경우 본 연구에서는 트랜스퍼크레인간의 간섭을 해결하기 위해 TC_2 는 기다리는 전략(waiting strategy)을 사용하였다.

단계7. QC_i 의 작업완료시간을 TC_i 의 작업완료시간 T_i 와 QC_i 의 작업완료시간 Q_i 와 비교하여 다음과 같이 결정한다.

$$Q_i = \max \{ Q_i, T_i \}$$

단계8. 할당해야할 선택클러스터가 존재하면 단계1로 간다. 모든 선택클러스터들이 할당되었으면 해의 생성단계는 종료되며 해의 개선 단계로 진행한다.

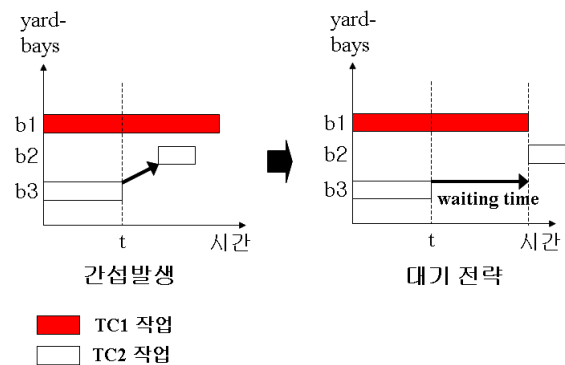


그림6. TC간 간섭과 해결의 예

3.2 해의 개선 단계 (Phase 2)

해의 개선 단계는 생성된 해로부터 지역적 탐색법을 사용하여 해를 개선시키는 단계이다. 본 연구에서는 해의 개선방법으로 일정계획문제에서 많이 사용하고 있는 2-opt 방법을 사용하였다. 해의 개선 단계의 각 단계별 세부적인 절차는 다음과 같다.

단계0. $j = 0$

단계1. $i = j + 1$. 만약 j 가 총 안벽크레인수보다 크면 종료한다. 그렇지 않으면 QC_j 를 선택한다.

단계2. 선택된 QC_j 에 할당된 선박클러스터들 중에서 두 개의 선박클러스터를 임의로 선택하여 작업 순서를 바꾼다(2-opt 방법). 이때 TC 에 할당된 장치장클러스터들의 작업순서도 바꿔 줘야 한다. 즉, TC_j 에 할당된 모든 장치장클러스터들을 대상으로 Phase 1의 단계5부터 단계7까지의 과정과 동일하게 Greedy 휴리스틱과 sweeping 전략을 적용하여 안벽크레인 작업완료시간 Q_j 를 계산한다. 선박클러스터의 쌍 중에서 가장 개선이 많은 두 선박클러스터의 순서를 변경하고 더 이상 개선되는 선박클러스터의 쌍을 찾을 수 없을 때까지 단계 2를 반복한다. 더 이상 개선되는 선박클러스터의 쌍이 없으면 단계1로 간다.

간단한 예제를 통하여 제안된 알고리즘의 Phase 1과 Phase 2의 과정을 살펴보면 다음과 같다. 표1과 표2는 예제에서 적용된 선박클러스터와 장치장클러스터의 정보이며 표3은 알고리즘 수행결과를 보여 주고 있다.

표1. 선박클러스터의 예

항목	1	2	3	4	5
p_j	720	900	1800	450	3150
SC_j	8	10	20	5	35
그룹	A	B	B	A	C
I_j	1	3	5	7	8

$\Psi = \{(1,2), (2,3), (3,4), (4,5)\}$, $\Phi = \{ \}$, $T = 2$, $I_1^c = 0$, $I_2^c = 6$, $r = 0.4$, ship-bay간 이동시간=10

표2. 장치장클러스터의 예

항목	1	2	3	4	5	6	7	8	9	10
$y_k^{(B,b)}$	(1,3)	(1,1)	(1,2)	(1,7)	(1,9)	(2,1)	(2,3)	(2,5)	(2,6)	(2,9)
YC_k	10	10	10	5	10	3	10	10	5	5
그룹	B	A	B	C	C	A	B	C	C	C

$\Pi = \{1,2,3,4,5,6,7,8,9,10\}$, $I_1^{(B,b)} = (1,0)$, $I_2^{(B,b)} = (2,0)$, yard-block간 이동시간=100, yard-bay 간 이동시간=10, 컨테이너 한 개당 TC작업시간=80

Phase 1

단계1. QC_1 이 선택된다. ($Q_1=0, Q_2=0$)

단계2. $\Omega = \{1,2,3,4,5\}$, $F = \{1,2,3\}$

$$1/|I_1^c - I_1| = 1/1 > 0.4 \times \max\{1/1, 1/2, 1/3\} = 0.4$$

$$1/|I_1^c - I_2| = 1/2 > 0.4 \times \max\{1/1, 1/2, 1/3\} = 0.4$$

$$1/|I_1^c - I_3| = 1/3 < 0.4 \times \max\{1/1, 1/2, 1/3\} = 0.4$$

$$F = \{1,2\}$$

단계3. 확률값 부여

$$\{(1/1)/(1/1 + 1/2), (1/2)/(1/1 + 1/2)\} = \{2/3, 1/3\}$$

단계4. 난수 0.52가 발생하였다고 하자. 선박클러스터 1이 선택된다.

$$Q_1 = 10 + 720 = 730, Q_2 = 0, I_1^c = 1, I_2^c = 5$$

단계5. TC_1 의 작업을 선택한다.

단계5.1. 작업목록을 작성한다.

$$\Pi = \{1,2,3,4,5,6,7,8,9,10\}, G_1 = \{2,6\}$$

단계5.2. 가장 거리가 짧은 작업을 선택한다.

$$\min\{|I_1^{(1,0)} - y_2^{(1,1)}|, |I_1^{(1,0)} - y_6^{(2,1)}|\} = \min\{10, 110\} = 10$$

$$S_1 = \{2\}$$

장치장클러스터 2가 선택된다.

단계5.3. $YC_2 = 10 \geq SC_1 = 8$ 이므로 선박클러스터 1이 요구한 컨테이너수가 충족된다. 단계6으로 간다.

$$YC_2 = 2, SC_1 = 0, I_1^{(B,b)} = (1,1), I_2^{(B,b)} = (2,0)$$

단계6. $T_1 = 10 + 640 = 650, T_2 = 0$

단계7. $\max\{Q_1, T_1\} = \max\{730, 650\} = 730$

$$Q_1 = 730, Q_2 = 0$$

단계1. QC_2 가 선택된다. ($Q_1=730, Q_2=0$)

단계2. $\Omega = \{2,3,4,5\}$, $F = \{3,4,5\}$

$$1/|I_2^c - I_3| = 1/1 > 0.4 \times \max\{1/1, 1/1, 1/2\} = 0.4$$

$$1/|I_2^c - I_4| = 1/1 > 0.4 \times \max\{1/1, 1/1, 1/2\} = 0.4$$

$$1/|I_2^c - I_5| = 1/2 > 0.4 \times \max\{1/1, 1/1, 1/2\} = 0.4$$

$$F = \{3,4,5\}$$

단계3. 확률값 부여

$$\left\{ \begin{array}{l} (1/1)/(1/1 + 1/1 + 1/2), (1/1)/(1/1 + 1/1 + 1/2), \\ (1/2)/(1/1 + 1/1 + 1/2) \end{array} \right\} = \{2/5, 2/5, 1/5\}$$

단계4. 난수 0.71이 발생하였다고 하자. 선박클러스터 4가 선택된다.

$$Q_1 = 730, Q_2 = 10 + 450 = 460, I_1^c = 1, I_2^c = 6$$

단계5. TC_2 의 작업을 선택한다.

단계5.1. 작업목록을 작성한다.

$$\Pi = \{1,2,3,4,5,6,7,8,9,10\}, G_4 = \{2,6\}$$

단계5.2. 가장 거리가 짧은 작업을 선택한다.

$$\min\{|I_2^{(2,0)} - y_2^{(1,1)}|, |I_2^{(2,0)} - y_6^{(2,1)}|\} = \min\{110, 10\} = 10$$

$$S_4 = \{6\}$$

장치장클러스터 6이 선택된다.

단계5.3. $YC_6 = 3 \leq SC_4 = 5$ 이므로 선박클러스터 4가 요구한 컨테이너수를 충족시키지 못한다. 단계5.2로 간다.

$$YC_6 = 0, SC_4 = 2, I_1^{(B,b)} = (1,1), I_2^{(B,b)} = (2,1), G_4 = \{2\}$$

$$T_1 = 650, T_2 = 10 + 240$$

단계5.2. 가장 거리가 짧은 작업을 선택한다.

$$\min\{|I_2^{(2,1)} - y_2^{(1,1)}|\} = \min\{100\} = 100$$

$$S_4 = \{6,2\}$$

장치장클러스터 2가 선택된다.

단계5.3. $YC_2 = 2 = SC_4 = 2$ 이므로 선박클러스터 4의 남은 컨테이너수를 모두 충족시킨다.

$$YC_2 = 0, SC_4 = 0, I_1^{(B,b)} = (1,1), I_2^{(B,b)} = (1,1), G_4 = \{ \}$$

단계6. TC_2 는 블럭간 이동을 고려하여 시간 350에서 작업이 가능하지만 현재 TC_1 이 시간 730까지 1블럭 1베이에서 QC_1 의 작업을 수행하고 있으므로 TC_2 는 730까지 대기한 후 작업을 시작할 수 있다. 따라서 TC_2 의 작업완료시간은 다음과 같다.

$$T_2 = \max(250 + 100 + 0, 730) + 160 = 890$$

단계7. $\max\{Q_i, T_i\} = \max\{450, 890\} = 890$

$$Q_1 = 730, Q_2 = 890$$

단계1. QC_1 이 선택된다. ($Q_1 = 730, Q_2 = 890$)

...

단계8. 모든 선박클러스터 및 장치장 클러스터를 할당한 안벽크레인 일정계획의 결과는 다음과 같다. 이때 총 작업완료시간은 4120이다.

$$QC_1: (1, 3, 2), TC_1: (2, 3, 1, 7), Q_1 = 3590$$

$$QC_2: (4, 5), TC_2: (6, 2, 4, 5, 8, 9, 10), Q_2 = 4120$$

Phase 2

단계1. QC_1 이 선택된다.

단계2. 2-opt 방법으로 가능한 순서쌍은 다음과 같다.

$$QC_1: (3, 1, 2), TC_1: (3, 1, 2, 7), Q_1 = 3530, Q_2 = 4140$$

$$QC_1: (2, 3, 1), TC_1: (3, 1, 7, 2), Q_1 = 3510, Q_2 = 4250$$

$$QC_1: (1, 2, 3), TC_1: (2, 3, 1, 7), Q_1 = 3510, Q_2 = 3950$$

가장 큰 개선을 가진 작업순서는 선박클러스터 3과 2를 바꾸는 것으로 작업완료시간은 3950이며 QC_1 및 TC_1 의 작업순서는 다음과 같다.

$$QC_1: (1, 2, 3), TC_1: (2, 3, 1, 7), Q_1 = 3510, Q_2 = 3950.$$

다음으로 가능한 작업 순서쌍은 아래와 같다.

$$QC_1: (2, 1, 3), TC_1: (3, 2, 1, 7), Q_1 = 3570, Q_2 = 4250$$

$$QC_1: (3, 2, 1), TC_1: (3, 1, 7, 2), Q_1 = 3530, Q_2 = 4140$$

더 이상 개선이 없으므로 단계 1로 간다.

단계1. QC_2 이 선택된다.

단계2. 가능한 순서쌍은 다음과 같다.

$$QC_2: (5, 4), TC_2: (8, 9, 10, 5, 4, 2, 6), Q_1 = 3470, Q_2 = 3610$$

작업완료시간이 3610으로 선박클러스터 4과 5를 바꾸는 것이 개선이 있다. 더이상 QC_2 의 경우 개선이 없으므로 Phase 2를 종료한다.

표3. Phase 1과 Phase 2의 결과

구분	선박 클러스터	컨테이너 개수	장치장 클러스터	컨테이너 개수
QC_1	1	8	2	8
	2	10	3	10
	3	20	1	10
QC_2	5	35	7	10
			8	10
			9	5
			10	5
			4	5
	4	5	5	10
			2	2
			6	3

4. 수치 실험

본 연구에서 제시한 안벽크레인 일정계획기법의 성능을 시험하기 위해서 실제 선박의 선적지시서와 야드맵에서 생성된 선박클러스터와 장치장클러스터를 가지고 실험을 수행 하였다. 본 연구에서 제시된 알고리즘은 C언어로 구현을 하였으며, 펜티엄4 2.4GHz PC에서 실험을 수행하였다. 수치 실험에 사용된 문제는 2문제이며, 세부 정보는 표 4와 같다. 표4에서 보는 것처럼, 선박클러스터 수는 두 문제 모두 58개이며, 장치장클러스터 수는 각각 80, 86개이다. QC와 TC의 수는 각각 3대씩으로 설정하였다.

GRASP의 종료조건은 MTI와 MNI가 된다. MTI는 최대반복수로 GRASP의 전체수행 반복수를 의미한다. 반면, MNI는 해의 개선이 없을 때의 반복수를 의미한다. 즉, MNI가 10이라는 것은 GRASP 반복절차에서 해가 연속해서 10번 개선되지 않을 때 종료한다는 것을 의미한다. 따라서 MTI보다는 MNI가 프로그램의 수행시간에 더 많은 영향을 줄 수 있다. 수치 실험에서는 MTI를 500으로 설정하였으며, MNI를 각각 5, 10, 20으로 설정하여 실험을 수행 하였다. 한편, 실험에 영향을 주는 요소로써 r 값이 있다. r 값은 0과 1사이의 값이므로 해의 생성 단계에서 작업가능목록의 크기를 결정한다. r 값이 큰 경우에는 작업가능목록이 줄어들게 된다. 본 실험에서는 r 값을 0.95, 0.7, 0.5로 각각 설정하였고, MNI와의 조합을 통해서 총 9가지 실험조건에 대하여 실험을 수행하였다.

표4. 실험에서 사용된 예제

내용	P1	P2
QC와 TC의 수	3	3
선박클러스터의 수	58	58
장치장클러스터의 수	80	86
컨테이너의 수	746	872
그룹의 수	9	8
블록의 수	10	10
블록당 베이 수	25	25

표5는 MNI와 r 값의 조합을 통해 실험한 결과를 보여주고 있다. 하나의 실험조건에 대하여 총 10회 반복하여 실험을 수행하였다.

표5. 실험조건별 평균 Makespan과 평균CPUtime

r	MNI	평균 Makespan (초)		평균 CPUtime (초)	
		P1	P2	P1	P2
0.5	5	37432	38984	231	317
	10	36305	37998	450	679
	20	35312	37432	1669	1669
0.7	5	36374	38721	262	408
	10	35315	37828	685	987
	20	33939	36992	1228	1791
0.95	5	33186	43425	323	497
	10	32886	40630	712	1302
	20	32541	38546	1272	2314

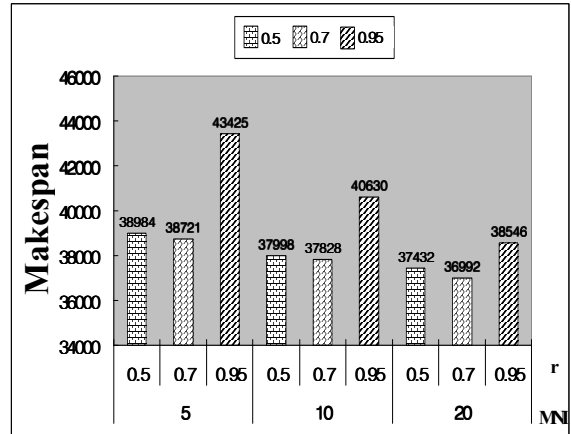


그림9. r값과 MNI의 조합에 따른 Makespan비교(P2)

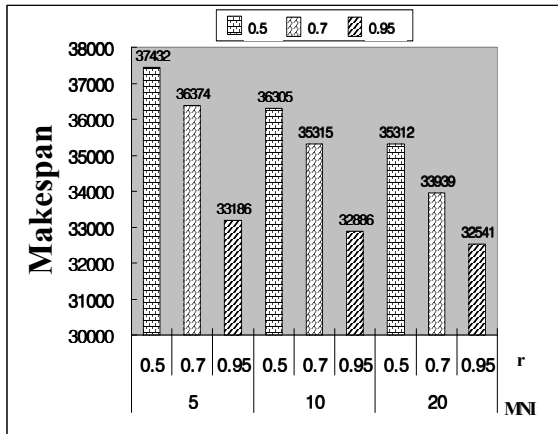


그림7. r값과 MNI의 조합에 따른 Makespan비교(P1)

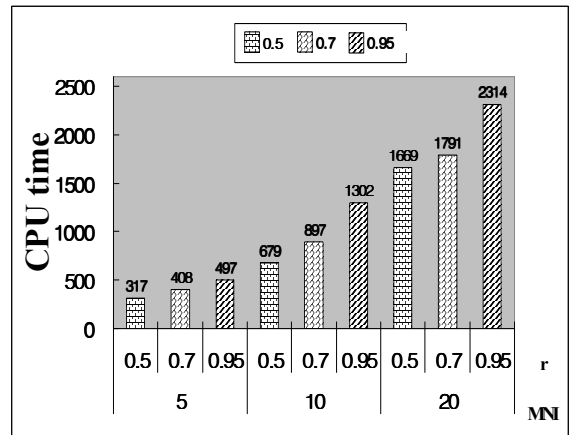


그림10. r값과 MNI의 조합에 따른 CPUtime비교(P2)

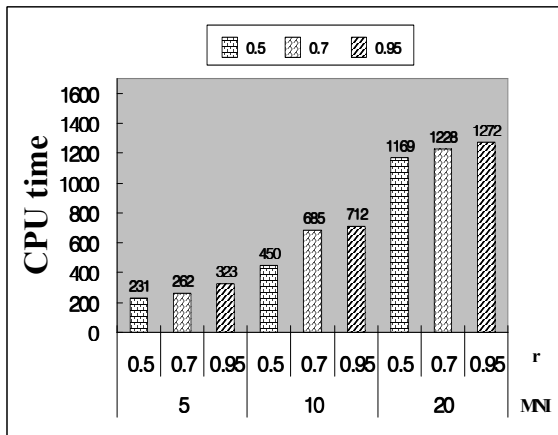


그림8. r값과 MNI의 조합에 따른 CPUtime비교(P1)

실험결과에서, Makespan은 MNI 값이 증가함에 따라 감소하는 경향을 보였다. 그림 7은 MNI와 r값이 증가함에 따라 해가 개선되는 것을 보여준다. 반면에, 그림 9에서는 P2의 경우 r값 0.7에서 가장 Makespan이 작았고 .95에서 Makespan이 증가하는 경향을 발견 하였다. 한편, 그림 8과 그림 10에서 보는 것처럼, MNI와 r값이 증가함에 따라 CPUtime이 증가하는 경향을 보였다.

5. 결론

본 연구에서는 컨테이너 터미널에서 중요한 역할을 수행하는 안벽크레인에 대한 작업일정계획문제를 다루었다. 안벽크레인 작업일정계획을 결정할 때 장치장의 현황 정보를 고려하였으며, 인공지능 기법 중 최적에 가까운 해를 찾아주는 탐색기법인 GRASP를 적용하여 실제 현장에서 적용 가능한 안벽크레인 작업일정계획 기법을 제안하였다. 또한, 제안된 안벽크레인 작업일정계획 알고리즘의 성능을 분석하기 위해 컨테이너터미널의 실제 자료를 수집하여 실험을 수행 하였다. 실험결과에서 r값이 0.7이상일때 Makespan이 감소하는 경향을 보였고, MNI가 증가함에 따라 CPUtime이 증가함을 알 수 있었다.

향후 연구에서는 TC간 간섭에 대한 다양한 해결 전략을 적용할 것이다. 또한, 해의 개선 단계에서 좀 더 효율적인 지역적 탐색기법의 개발이 필요

하다.

Acknowledgement

본 연구는 해양수산부의 '지능형 항만물류 시스템 기술개발' 사업의 일환으로 수행되었습니다.

참고문헌

- Daganzo, C. F. (1989), The Crane Scheduling Problem, Transportation Research 23B, 159-175.
- Kim, K.H. and Park Y. M. (2004), A crane scheduling method for port container terminals, European Journal of Operational Research, 156, 752-768.
- Kim, K.H., Kang, J.S., and Ryu, K.R., A beam search algorithm for the load sequencing of outbound containers in port container terminals, OR Spectrum, Vol. 26, pp93-116. 2004.
- Kim, K.H., Kim, K.Y., An Optimal Routing Algorithm for a Transfer Crane in Port Container Terminals, Transportation Science, Vol. 33, No. 1, pp17~33. 1999
- Peterkofsky, R. I. and Daganzo, C.F. (1990), A Branch and Bound Solution Method for the Crane Scheduling Problem, Transportation Research, 24B(3), 159-172.