

다중(Multiple) 야드 크레인의 효율적인 운영을 위한 점수기반(Score-based) 작업배정(Dispatching) 로직

김도경¹, 이기주², 정동진³, 채준재^{4*}

^{1,2,3,4}한국항공대학교 항공교통물류학과

An Efficient Score-based Dispatching Logic for Operating Multiple Yard Cranes

Do-kyung Kim¹, Keyju Lee², Dongjin Jeong³, Junjae Chae^{4*}

^{1,2,3,4}School of Air Transport, Transportation and Logistics, Korea Aerospace University

The need for automation in ports is growing, and a strategy gaining attention involves automating through retrofitting methods to enhance productivity while maintaining the flexibility of rubber-tired gantry cranes (RTGC) in the port. RTGCs offer greater flexibility than rail mounted gantry cranes (RMGC) as they are not fixed to specific container blocks and can move to adjacent blocks. Therefore, an operational strategy is required that can adapt to situations such as changes in the number of cranes for efficient operation. In this study, a score-based dispatching logic is proposed to efficiently allocate tasks for port yard cranes in real-time. In this logic, scores are calculated as the sum of appropriate weighted values for various factors, and tasks with the best scores are prioritized for execution. Comparing the performance metrics with other dispatching rules through AutoMod simulation, the superiority of the score-based dispatching logic is confirmed. Also, it is found that inter-block movements of RTGCs can improve key performance indicators in port yards, such as turnaround time.

Keywords: Container terminal automation, Yard crane dispatching rule, Rubber tyred gantry crane, Multiple yard blocks, Multiple yard cranes

논문접수일 : 2023.10.20. 논문수정일 : 2023.12.05. 게재확정일 : 2023.12.15.

1. 한국항공대학교 항공교통물류학과 석사과정

2. 한국항공대학교 항공교통물류학과 박사과정

3. 한국항공대학교 항공교통물류학과 석사과정

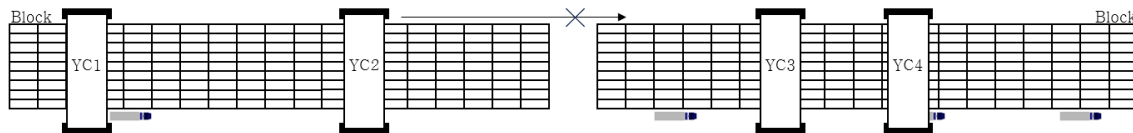
4*. 한국항공대학교 항공교통물류학부 교수, Corresponding Author: jchae@kau.ac.kr

1. 서론

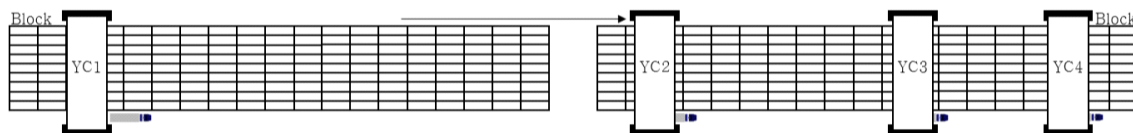
1.1 연구 배경

우리나라 전체 무역 증량의 99% 이상이 해운을 통하고 있으며, 이 중 60% 이상이 컨테이너에 의해 운송되고 있다(Y. F. Liu & Lee, 2019). 컨테이너 물동량은 지속적으로 증가하고 있으며(Ministry of Oceans and Fisheries, 2021a), 이에 따라 컨테이너항만 자동화에 대한 수요 또한 증가하고 있다.

항만의 핵심 장비인 야드 크레인에는 크게 RMGC(rail mounted gantry crane)와 RTGC(rubber tired gantry crane)로 구분할 수 있다. 레일을 기반으로 야드 블록 내에서만 운영되는 RMGC는 자동화에 유리하지만 지형적인 제약(지진 등)에 취약하다는 특징을 갖는다. 타이어 기반의 RTGC는 상대적으로 이동속도가 느리지만 야드의 건설비용이 낮고, 블록 간 이동과 같은 유연한 운영을 통해 활용률을 높일 수 있다는 장점이 있다(Sha et al. 2017). 아래 Figure 1은 두 가지 유형 크레인의 운영 특성 차이를 예시로 나타낸다.



(a) Rail mounted gantry cranes generally do not travel through blocks



(b) Rubber tired gantry cranes travel through blocks

Figure 1. Operational difference of rail mounted gantry cranes and rubber tired gantry cranes

1.2 연구 필요성 및 목적

주로 오래된 항만시설에는 RTGC가 배치되어 있다. RTGC는 최근 자동화 항만에 설치되는 RMGC에 비해 낮은 생산성 지표를 나타내지만, 크레인을 특정 블록에 고정하지 않는 유연한 운영이 가능하다는 장점이 있다. 이러한 유연성을 극대화하기 위해서는 RTGC의 자동화가 적극적으로 고려되어야 한다. 우리나라는 경제성장기인 1970~1980년대에 항만시설을 집중적으로 건설하였으며, 2030년에는 30년 이상 사용된 노후 항만시설물 비율이 50%를 넘어서게 된다(Ministry of Oceans and Fisheries, 2021b). 노후 항만의 RTGC 자동화를 준비하기 위해 하드웨어적 자동화시스템 기술연구가 진행된 바 있으며(Global Logistics Research Institute, 2009), 최근에는 운영 및 소프트웨어 측면까지를 포함한 기술연구가 진행되고 있다(Ministry of Oceans and Fisheries, 2022).

세계적인 터미널운영시스템 공급사 Navis(2023)가 자동화 터미널의 신규건설보다는 기존시설을 자동화 개조(retrofit)하는 것이 더욱 합리적일 수 있음을 지적한 것(Port Technology International, 2020)을 포함하여, 최근에는 비교적 적은 비용으로 RTGC 등의 기존 장비를 개조하는 자동화 전략이 주목받고 있다(Korea Maritime Institute, 2020). 또한, 중국, 중동지역 등의 해외지역에서도 안전성, 운영 일관성, 생산성 확보를 위한 RTGC 자동화에 관심을 가지면서(Navis Collaboration Center, 2019), 효율적인 자동화 RTGC 운영로직에 대한 연구가 요구되고 있다.

본 연구에서는 자동화 RTGC의 특성인 크레인의 블록 간 이동과 블록 당 크레인의 운영 대수의 유연성을 효율적으로 활용하기 위한 점수기반 작업배정 로직(score-based dispatching logic)을 제시한다. RTGC의 운영 특성은 RMGC의 운영에서 일반적으로 두 대의 크레인이 하나의 야드 블록에서 고정적으로 사용되는 방식과 대비된다. 본 연구에서는 점수기반 작업배정 로직의 성능을 확인하기 위해 AutoMod 시뮬레이션을 구성한다. 두 개의 컨테이너 블록에서 최소 한 대, 최대 네 대의 크레인이 동시에 투입된 환경을 검토하며, 타 작업배정 규칙과의 비교를 통해 점수기반 작업배정 로직의 우수성을 확인한다.

2. 문헌연구

2.1 야드 크레인 스케줄링(scheduling)

야드 크레인 스케줄링은 복잡도가 매우 높은 NP-완전 문제이며(Ng & Mak, 2005), 문제의 크기(작업의 수와 계획기간의 길이)가 커지면 가능해(feasible solution) 조차 합리적인 시간 내에 찾아낼 수 없다. 야드 크레인 스케줄링 최적화(optimization) 문제를 해결하기 위한 연구에서는 일반적으로 혼합정수계획 수리모형을 구성한 뒤, 분기한정법과 같은 최적(exact) 알고리즘, 또는 지역최적해를 찾아낼 수 있는 발견적(heuristic) 해법을 제시한다. 야드 크레인 스케줄링은 두 대의 크레인 즉, 트윈(twin) 야드 크레인의 작업 형태가 가장 일반적으로 연구되고 있다. 그러나 RTGC의 유연한 이동성은 특정 컨테이너 블록의 범위 내에서 두 대 이상의 크레인이 동시에 작업하는 상황을 가능하게 하며(Sha et al., 2017), 크레인 간 상호간섭(충돌)을 고려해야 하는 다중(multiple) 야드 크레인의 환경에서는 스케줄링 문제의 복잡도가 더욱 증가한다(Chu et al., 2019).

Briskorn & Zey(2018)는 트리플 교차(crossover) 크레인(아래 Figure 2 참조)의 작업시간을 최소화하기 위해 혼합정수계획을 구성한 뒤, 크레인 간 상호 간섭을 고려할 수 있는 그래픽모델(graphical model)을 활용한 발견적(heuristic) 해법을 제시하였다. 문제 해결을 위한 제한시간(1시간) 이내에 최대 16개의 작업을 수행하기 위한 최적해를 도출하였으며, 발견적 해법을 적용하여 최대 32개 작업에 대한 지역최적해를 도출하였다. 이어지는 연구(Briskorn & Zey, 2020)에서는 컨테이너 이송작업시간 최소화를 위한 분기한정법 알고리즘을 제안하였으며, 1시간 이내에 최대 10개 작업에 대한 최적해와 최대 25개 작업에 대한 지역최적해를 도출하였다. Chu et al.(2019)은 두 컨테이너 블록에서 교차 불가능한(non-crossing) 세 대의 크레인(아래 Figure 2 참조)에 대하여 충돌을 방지하고 작업 대기시간을 최소화하기 위한 혼합정수계획 모형을 구성한 뒤, 근사해를 도출하기 위한 유전 알고리즘을 제시하였다. 해당 연구에서는 2시간의 문제해결 제한시간 이내에 최대 12개 작업에 대한 최적 작업 스케줄을 도출하였다.

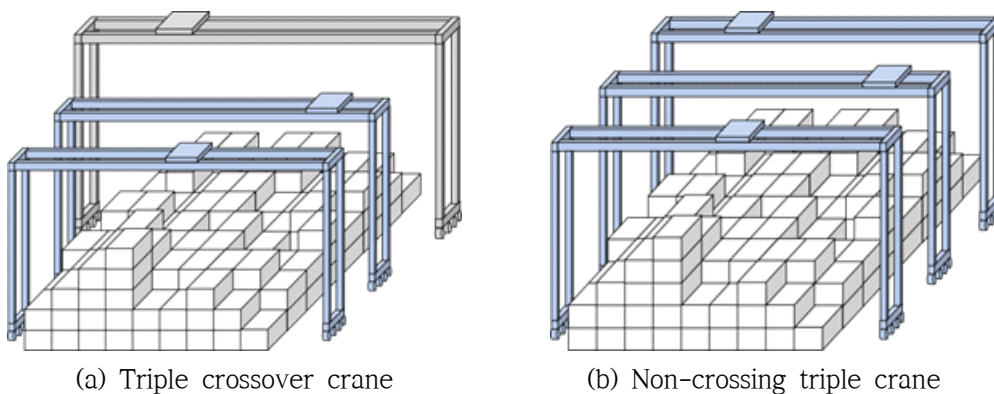


Figure 2. Typical configuration of triple crossover cranes and triple non-crossing cranes

X. Guo et al.(2008)은 한 개의 컨테이너 블록에 한 대의 야드 크레인이 투입된 환경에서 작업 대기시간을 최소화하기 위한 전략을 제시하였다. 해당 연구에서는 전체 작업을 소규모 계획 단위(planning window)

로 분할한 뒤 각 단위 내에서의 최적 작업계획을 결정하고 이를 이어 붙여 전체 작업계획을 완성하였으며, 성능 검증을 위해 선착순(first come first served : FCFS) 및 최근접(nearest first served : NFS) 작업배정 규칙과의 비교를 진행하였다. Huang et al.(2012)의 연구에서도 한 개의 블록 내에 배치된 한 대의 야드 크레인을 가정하였다. 해당 연구에서는 작업 대기시간을 최소화하기 위한 수정된(modified) A* 알고리즘 제시하였으며, 선착순 및 최근접 작업배정 규칙과 비교하여 알고리즘의 성능을 검증하였다. 이어지는 연구(Huang et al., 2014)에서는 기존 연구의 방법론을 발전시켰으며, 작업의 대기시간 감소 효과 및 이로 인한 선박의 항만 체류시간 감소 효과를 함께 확인하였다.

2.2 야드 크레인 작업배정 규칙(dispatching rule)

작업배정 규칙은 서비스 대기 중인 일련의 작업 중에서 다음으로 처리할 작업을 선택하기 위해 사용되며, 최적을 보장하지 않는 실시간 규칙의 특성상 그 성능은 작업장 구성형태와 운영 조건에 따라 달라질 수 있다(Blackstone et al., 1982).

Petering et al.(2009)의 연구에서는 야드 크레인의 작업배정 규칙(선착순, 최근접 등)에 따른 안벽 크레인의 시간당 평균 작업횟수(gross crane rate) 지표 변화를 모의실험을 통해 확인하였다. 해당 연구에서는 블록에 장치되었던 컨테이너를 선박에 적재하는 작업의 우선순위를 높이는 것이 성능지표 개선으로 이어지는 것을 확인하였다. X. Yu et al. (2018)의 연구에서는 야드 크레인의 작업배정 규칙에 따른 터미널의 운영 비용을 성능지표로 사용하였다. 작업배정 규칙으로 각 작업에 배정된 크레인과의 거리, 외부트럭 및 내부트럭의 대기시간 등을 고려하였으며, 모의실험을 통해 크레인으로부터 근접한 작업을 우선적으로 고려하는 전략이 운영비용을 감소시키는 것으로 분석하였다.

P. Guo et al. (2019)의 연구에서는 두 개의 컨테이너 블록이 존재하며 각 블록에 두 대의 트윈 야드 크레인이 배치된 환경에서 야드 크레인의 작업배정을 효율화하고자 하였다. 해당 연구에서는 두 블록 사이에 중계영역(handshake area)을 두어 컨테이너의 블록 간 이동이 가능하도록 하는 특수한 상황을 고려하였다. 중계영역은 동시에 한 대의 크레인만 점유할 수 있어 간섭(충돌)방지를 고려한 작업배정 규칙이 요구된다. 해당 연구에서는 전체 작업의 완료시간 최소화를 목표로 간섭방지 규칙과 작업별 우선순위 규칙을 조합한 다양한 작업배정 규칙의 성능을 검토하였다.

2.3 연구 차별점

야드 크레인 스케줄링 문헌연구를 통해 다중 야드 크레인의 스케줄링 접근법은 복잡도가 높아 즉각적으로 해를 도출하는 데에는 무리가 있음을 확인할 수 있다. 또한, 사전에 알려진 작업정보를 기반으로 도출된 최적의 작업계획은 동적(dynamic)으로 변화하고 불확실성(uncertainty)이 존재하는 항만 환경(W. Liu et al., 2021; D. Yu et al., 2019)에서 높은 빈도로 재계획(rescheduling)되어야 한다는 문제가 존재한다. 다중 야드 크레인의 운영을 위한 현실적인 접근법으로 실시간으로 즉각적인 대응이 가능한 작업배정(dispatching) 규칙을 고려해볼 수 있다.

야드 크레인 작업배정 규칙과 관련한 문헌연구에서는 선착순 또는 최근접 규칙과 같이 단순한 방법에 집중하지만, 그 성능이 충분히 현실 적용 가능한 수준인 것으로 검증되었다. 특히, 문헌을 통해 작업 종류에 따른 우선순위나 크레인 간 간섭(충돌) 해결을 위한 여러 규칙을 다양하게 조합하는 것으로 개선된 작업배정 규칙이 제시될 수 있음을 확인할 수 있다.

기존의 야드 크레인 작업배정 규칙 연구에서 선착순, 최근접 등의 규칙을 배타적으로 선별하여 조합했던 것과 달리, 본 연구에서는 각 배정 규칙에 가중치를 적용한 점수기반 작업배정(score-based dispatching) 로직을 제시하였다. 야드 크레인과의 작업(트럭) 사이의 거리, 트럭이 컨테이너 블록에 도착한 시간, 각 작업의 예상 소요시간과 같은 요소들은 모두 효율적인 작업배정이라는 의사결정에 어느 정도 영향력을 갖는다고 볼 수 있다. 본 연구에서는 각 요소들에 대한 가중치를 결정하여 조합함으로써 현실적인 작업배정 규칙을 제시하였다. 야드 크레인 작업배정을 위한 점수기반 로직은 기존에는 없던 새로운 방향의 연구라고 할 수 있다.

또한, 기존의 연구에서 고정된 개수의 블록과 크레인 대수를 가정했던 것과 달리, 본 연구에서는 컨테이너 블록을 두 개로 고정하지만, 투입 크레인의 대수는 최소 한 대부터 최대 네 대까지의 상황을 검토하였다. 시뮬레이션 실험을 통해 투입 크레인의 대수에 따른 점수기반 작업배정 로직의 각 요소들에 적용될 최적 가중치를 제시하였으며, 이를 통해 점수기반 작업배정 로직이 다양한 환경에서 모수 조정을 통해 일반적으로 적용될 수 있음을 보였다. 본 연구에서는 RTGC 크레인을 사용하여 블록 간 이동이 가능한 상황을 가정하였으나, 블록 간 이동이 가능한 경우와 불가능한 경우의 성능지표를 비교하기 위한 시나리오를 추가적으로 구성하였다.

3. 문제 정의 및 가정 사항

3.1 문제 환경

1) 항만의 구조적 특성

컨테이너항만의 구조는 Figure 3에서 확인할 수 있듯 유럽형(European, perpendicular) 레이아웃과 아시아형(Asian, horizontal) 레이아웃으로 구분할 수 있다(Carlo et al., 2014). 본 연구에서는 안벽과 컨테이너 블록이 평행한 아시아형 레이아웃을 사용하였으며, 양측이 아닌 단측 I/O(input and output) 레인을 가정하였다.

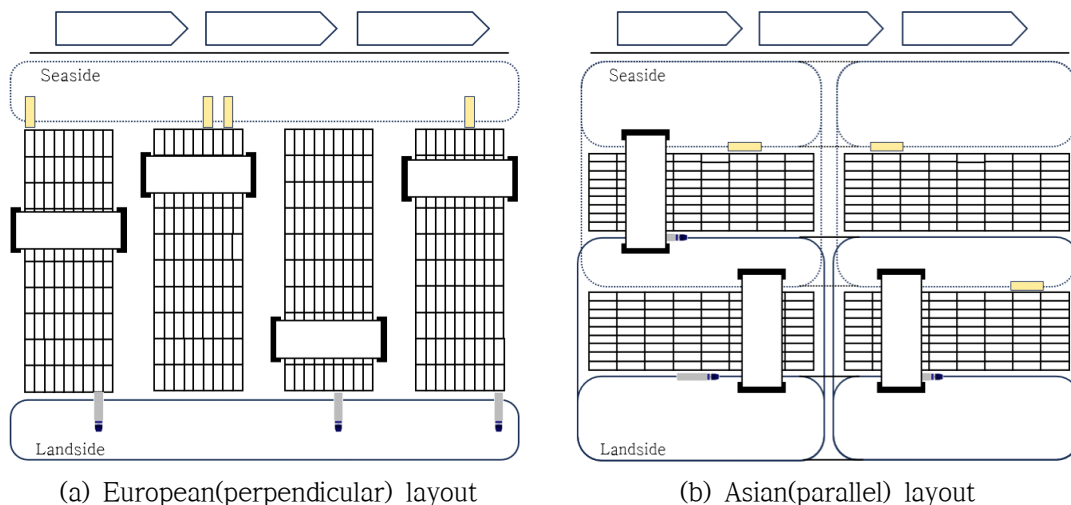


Figure 3. Typical container port terminal layouts

본 연구에서는 아래 Figure 4에 나타난 것과 같이 50-베이(bay), 9-열(row), 5-층(tier) 형태의 컨테이너 블록을 사용하며, 투입된 모든 야드 크레인 작업범위는 두 개의 컨테이너 블록으로 가정하였다. 20ft 컨테이너는 한 개의 베이, 40ft 컨테이너는 두 개의 베이 공간을 차지하도록 하였으며, 서로 다른 크기의 컨테이너가 동일한 베이에 함께 적치될 수 없도록 하였다. 냉동, 위험물 컨테이너 등은 별도의 적치 공간(블록)을 가정하여 본 연구에서는 고려하지 않았다.

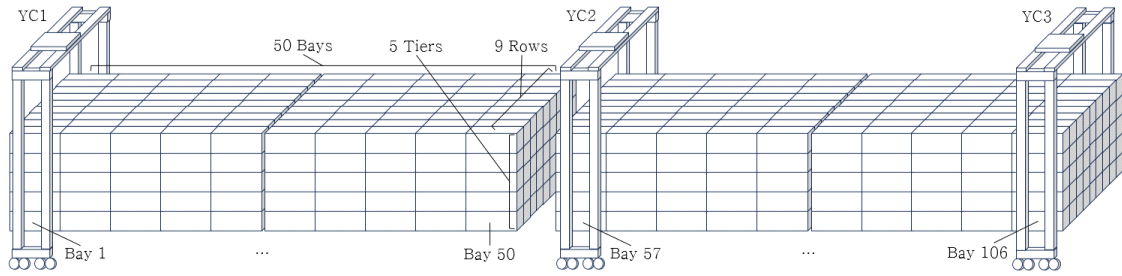


Figure 4. An example of three non-crossing yard cranes and in two container blocks

2) 야드 크레인의 특성

본 연구에서는 두 개의 컨테이너 블록 내에 최소 한 대부터 최대 네 대까지의 RTGC 야드 크레인이 투입된 상황을 가정한다. 각 야드 크레인은 서로 교차(cross)할 수 없으며, 충돌방지를 위해 최소 두 개의 베이의 안전거리를 유지해야 한다. 각 야드 크레인은 RTGC의 유연한 이동성을 반영하여 두 컨테이너 블록 간 이동이 가능하도록 하였다. 각 야드 크레인은 한 번에 하나의 작업만을 처리할 수 있으며, 작업을 마친 크레인은 다음 작업배정을 위해 종료위치에서 대기한다. 야드 크레인에 적용된 속도 및 가속속 정보는 Table 1에 정리된 바와 같다. 야드 크레인은 움직임은 베이 이동(x 축의 gantry drive), 로우 이동(y 축의 trolley drive), 층 이동(z 축의 hoist drive)로 구분되며, x 축과 y 축을 동시에 움직이는 것이 가능하지만, z 축의 움직임은 x 와 y 축이 정렬된 이후에 개별적으로 실행한다.

Table 1. RTGC specifications on speed, acceleration, and deceleration

		Category	
		Speed(m/min)	Accel.Decel.(m/sec ²)
Gantry Drive	Rated load	90	0.25
	No load	130	0.24
Trolley Drive		70	0.33
Hoist Drive	Rated load	25	0.28
	No load	50	0.28

3) 작업의 특성

본 연구에서 사용한 작업 생성분포 및 작업특성 비율 등은 대한민국 소재의 한 컨테이너 항만에서 수집한 데이터를 근거로 한다. 컨테이너를 블록 내부에 적치하는 반입 작업은 지수분포(exponential distribution)를 따라 생성되며, 반입된 컨테이너의 장치기간(dwell time)이 지난 뒤 반출작업이 생성된다. 컨테이너 터미널의 혼잡 시간대와 비혼잡 시간대를 현실적으로 모사하기 위해 반입 작업 생성 지수분포의 모수(평균)가 시간대에 따라 변경되도록 하였다. 예를 들어, 두 컨테이너 블록 내에 세 대의 야드 크레인 투입된 시나리오에서는 최대 혼잡시 평균적으로 610초, 최소 혼잡시 5,583초에 하나의 반입 작업이 생성된다. 각 컨테이너의 장치기간은 경험적(empirical) 분포(최소 0.25일, 최대 28.66일 수준)를 따른다. 반입 및 반출 작업은 해측작업(우선순위 높음)과 육측작업(우선순위 낮음)으로 한 번 더 구분되며, 해측 대 육측작업의 비율은 (63.5 : 36.5)로 설정하였다.

반출작업 대상 컨테이너 상단에 존재하는 컨테이너들을 이동시키는 재취급(re-handling) 작업은 동일 베이 내에서만 수행되며, 빈 공간이 존재하는 가장 인접한 로우를 이동위치로 선택한다. 각 베이 내에는 4개 이상의 여유공간(buffer)을 두어 동일 베이 내 재취급이 항상 가능하도록 하였다. 반입 또는 반출을 위해 컨테이너 야드에 도착한 차량은 한 번에 하나의 컨테이너만 실을 수 있는 것으로 가정하였다.

3.2 목적 함수

본 연구에서 제시하는 점수기반 작업배정 로직은 차량의 평균 턴타임(turn time 또는 turnaround time : TT)을 최소화하는 것을 목표로 한다. 본 연구에서 사용되는 차량은 해측작업을 위한 야드 차량(yard truck)과 외부차량(road truck)이 있으며, 평균 턴타임 계산 시 두 종류의 차량이 모두 고려된다. 본 연구에서는 턴타임을 아래 Figure 5에 예시된 것과 같이 왼쪽 블록 끝의 시작지점(start)으로부터 오른쪽 블록 끝이 종료지점(end)까지 소요된 시간으로 정의한다. 차량의 턴타임에는 이동에 소요된 시간, 목적지(베이) 도착 후 크레인의 작업을 대기한 시간, 그리고 크레인이 작업한 시간이 포함된다. 평균 턴타임 외에도 추가적인 성능지표로 최대 턴타임을 함께 확인한다.

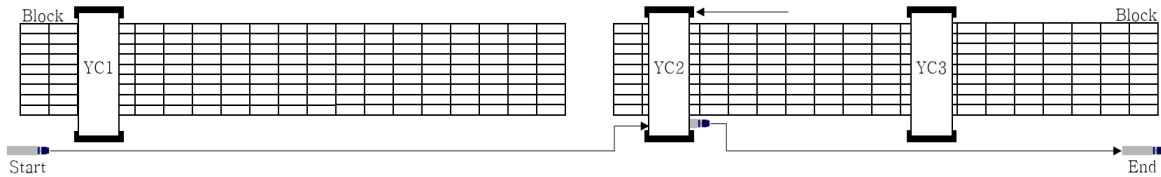


Figure 5. Explanation on how turnaround time is measured

4. 점수기반 작업배정 로직

4.1 점수의 구성

크레인의 성능에 영향을 미치는 다양한 변수를 추출하여 항만에서의 크레인 성능을 평가하기 위한 실험을 진행했다. 해측과 육측의 우선순위, 현재 크레인과 작업 간의 거리, 초기 크레인 위치와 작업 간의 거리, 작업 소요 시간, 리핸들링 시간, 작업 대기 개수, 대기 시간 등의 변수를 추출하여 목적함수로 설정한 작업 턴타임을 최소화하는 데 어떤 변수가 긍정적인 영향을 미치는지 확인하였다. 실험 결과, 크레인과 작업 간의 거리, 대기 시간, 작업 소요 시간 변수가 작업 턴타임을 유의미하게 줄이는 데 긍정적인 효과를 나타내는 것으로 확인되었으며, 이들을 점수 로직의 구성 요소로 선택했다.

본 연구에서 제시하는 다중 야드 크레인의 점수기반 작업배정 로직에 사용되는 표기법은 아래 Table 2에 제시된 바와 같다. 거리(D)의 단위는 베이의 개수를 사용하며, 이를 제외한 모든 단위는 시간(초)으로 관리한다. 거리(D)는 특정 크레인으로부터 특정 작업까지의 거리(베이 개수)를 의미한다. 대기시간(WT)은 작업이 생성된 시간과 현재시간의 차이로 계산되며, 작업소요시간(PT)은 특정 크레인에 기할당된 작업을 포함하여 새로운 작업이 할당 되었을 때 계산(예상)되는 작업소요시간 즉, 새로 할당된 작업이 종료되기까지의 예상 소요시간을 의미한다. 재취급시간(RT)은 모의실험 데이터를 기반으로 최소값, 최빈값, 최대값을 추출하여 삼각분포로 적용하였다. 본 연구에서는 작업의 턴타임(TT)을 최소화하는 것을 목표로 한다.

Table 2. Notations used for constructing the score-based dispatching logic

Notation		Definition
D	Distance	Number of bays from a crane's position to a job
AT	Arrival Time	Vehicle arrival time to the system (container blocks)
WT	Wait Time	Current time minus a job's AT
ST	Start Time	Time of job start (crane arriving at the vehicle's bay position)
RT	Re-handling Time	Expected time for re-handling for a job $RT \sim tri(40, 90, 458) \times (\text{number of containers stacked on top})$
PT	Processing Time	Expected processing time of a job using a crane $PT_{curr} = RT + PT_{prev}$
CT	Completion Time	Time of job completion (vehicle leaving the container blocks)
TT	Turnaround Time	$CT - AT$

본 연구의 점수기반 작업배정 로직에서 점수(score)는 아래 수식(1)에 나타난 것과 같이 거리(D), 대기시간(WT), 작업소요시간(PT) 변수(factor)에 대한 함수로 정의된다. 각 변수는 정규화(min-max scaling normalization)를 적용하여 0과 1 사이의 값으로 변환한 뒤 점수를 계산하는 데 사용하였다. 수식에 나타난 것과 같이 각 변수에는 가중치(α, β, γ)가 곱해져 더해지며, 점수가 낮을수록 유리한 작업으로 판단한다. 다양한 환경에서의 최적 가중치 조합 (α, β, γ)는 시뮬레이션 실험을 통해 도출한다.

$$Score(D, WT, PT) = \alpha \times D + \beta \times (1 - WT) + \gamma \times PT \quad (1)$$

4.2 점수의 계산 및 작업배정

새로운 작업이 생성되거나 야드 크레인의 작업이 완료되었을 때, 각 야드 크레인에 할당할 다음 작업을 결정하기 위해 점수기반의 작업배정 로직을 사용한다. 점수 계산에 의한 작업배정은 크게 두 가지 과정을 통해 실행된다. 첫 번째는 작업을 크레인에 할당하는 과정이며, 두 번째는 크레인에 할당된 작업 중 가장 점수가 좋은(값이 작은) 작업을 선택하는 과정이다.

점수를 계산하는 시점의 크레인 위치를 기반으로 컨테이너 블록의 영역을 아래 Figure 6에 예시된 것과 같이 구분할 수 있다. 예시된 그림에서 1번 영역(Area1)에 존재하는 작업은 1번 야드 크레인(YC1)만 접근할 수 있으며, 4번 영역(Area4)의 작업은 3번 야드 크레인(YC3)만 접근할 수 있다. 본 연구에서는 비교차(non-crossing) 형태의 크레인을 가정하므로, 별도의 점수 계산 없이 1번 영역의 작업은 1번 크레인으로, 4번 영역의 작업은 3번 크레인으로 할당한다. 반면, 두 대의 크레인이 접근할 수 있는 2번 영역(Area2)과 3번 영역(Area3)에 존재하는 작업은, 점수 계산을 통해 일차적으로 야드 크레인에 할당한다.

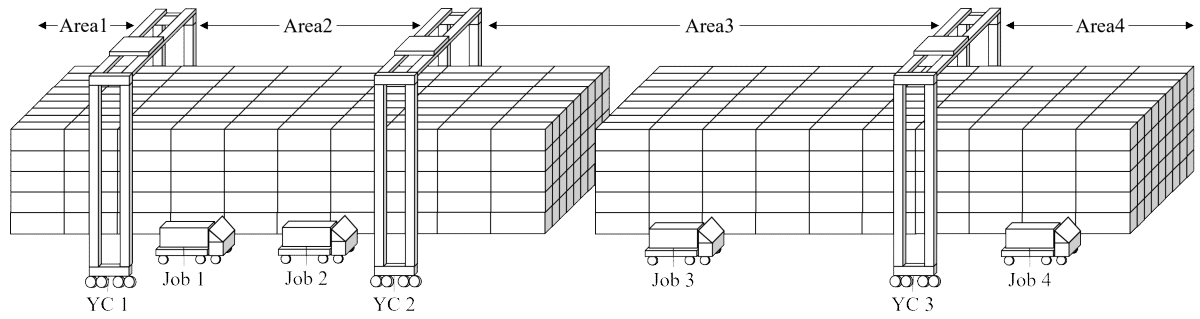


Figure 6. An example state of 2-block 3-crane environment

앞서 크레인에 작업을 할당하는 과정에서는 다수의 작업이 동일한 크레인에 할당되는 것이 허용된다. 이후, 각 크레인에 할당된 여러 작업 중에서 가장 우수한(값이 작은) 점수를 갖는 작업을 선택하도록 한다. 아래 Table 3에 제시된 예시는 Figure 6과 같은 상황에서 세 대의 야드 크레인이 모두 가용상태일 때를 가정한다. 해당 표에서는 각 크레인에 작업이 할당되는 과정과 할당된 작업 중 실행할 작업을 선택하는 과정을 확인할 수 있다. 각 변수(D , PT , WT)는 각 min-max 정규화가 적용되어 0과 1 사이의 값으로 나타난다. 표에서 확인할 수 있듯, 크레인에 의해 구분된 영역(Area)과 작업의 생성 위치(Bay)에 따라 일부 점수가 계산되지 않는 작업-크레인(Job-YC) 조합이 존재한다. 최종 점수를 기반으로, 1번 야드 크레인(YC1)은 1번 작업(Job1)을, 2번 야드 크레인(YC2)은 2번 작업(Job2)을, 3번 야드 크레인(YC3)은 4번 작업(Job4)을 실행한다. 남아 있는 작업(Job3)은 다음 점수계산 시점(가용 상태의 야드 크레인이 생겨났을 때 등)에 점수기반 로직에 의해 크레인에 배정된다.

Table 3. An example of normalized score calculations with $(\alpha, \beta, \gamma) = (1, 1, 1)$

Job	Area	Bay	Normalized score calculation												Execute now
			YC1			YC2			YC3			YC assign	WT	Final score	
			D	PT	Score	D	PT	Score	D	PT	Score				
1	2	15	0.052	0.219	0.271	0.513	0.585	1.098	-	-	-	1	0	0.271	TRUE
2	2	35	0.564	0.514	1.055	0.154	0.291	0.446	-	-	-	2	0.235	0.681	TRUE
3	3	65	-	-	-	0.615	0.516	1.131	0.410	1.107	1.517	2	0.601	1.732	False
4	4	95	-	-	-	-	-	-	0.154	0.841	0.995	3	0.002	0.997	TRUE

5. 실험

5.1 시뮬레이션 구성

본 연구에서는 Figure 7에 제시된 시뮬레이션 흐름도에 따라 AutoMod 시뮬레이션(아래 Figure 8 참조)을 구성하였다. 먼저, 컨테이너 반입 작업은 시간대별로 구분된 지수분포에 의해 생성된다. 반입된 컨테이너에는 임의의 장치기간이 설정되며, 장치기간 이후에는 반출작업이 생성된다. 생성된 반입 및 반출작업은 작업 목록(working list)에 포함되며, 점수기반 작업배정 로직은 작업 목록 내에 존재하는 작업들을 대상으로 실행된다. 점수의 계산 및 작업배정의 실행은 새로운 작업이 작업 목록에 포함된 시점 또는 야드 크레인이 가용상태(idle)인 시점에 수행된다.

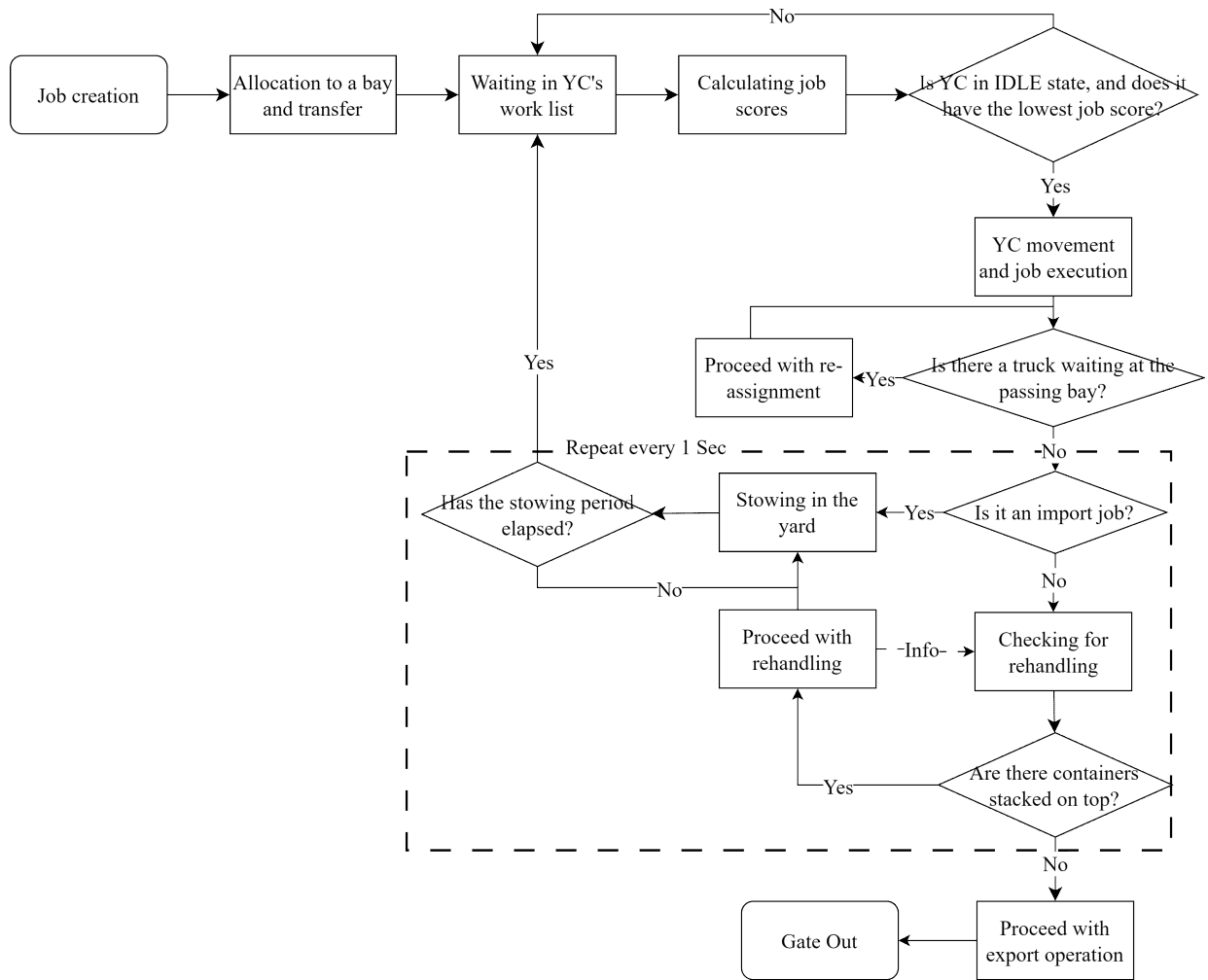
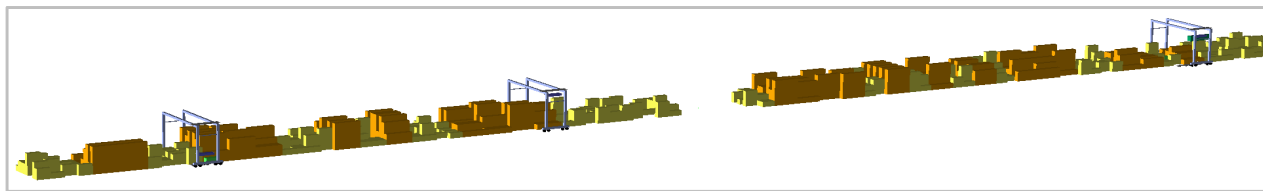


Figure 7. Flow chart for the AutoMod simulation model



(a) Top view of the 2-block 3-crane simulation model



(b) Bird's eye view of the 2-block 3-crane simulation model

Figure 8. AutoMod simulation model constructed for experiments

작업의 재할당(re-assign)은 특수한 조건에서만 실행된다. 첫 번째로, 야드 크레인이 특정 작업을 수행하기 위해 이동 중에 새로운(점수 계산 시 작업 목록에 존재하지 않았던) 작업의 차량이 크레인의 경로 상에 발견된 경우, 새로운 작업을 먼저 수행 후 기존의 목표 작업을 수행한다. 두 번째로, 야드 크레인이 특정 작

업을 수행하기 위해 특정 베이에 도착했으나, 목표 차량이 다른 작업을 대기하는 차량에 의해 가로막힌 경우, 먼저 도착해 있던 차량의 작업을 먼저 수행 후 기존의 목표 작업을 수행한다. 목표 컨테이너 상단에 다른 컨테이너가 쌓여있다면 재취급(re-handle) 작업을 수행한다. 재취급 작업은 동일 베이 내에서만 수행되며, 빈 공간이 존재하는 가장 인접한 로우를 이동위치로 선택한다.

실험을 통해 야드 블록 내에 존재하는 컨테이너 장치량(반입 컨테이너의 최대장치기간은 28.66일 수준)을 분석하여 30일의 워업(warm-up) 기간을 설정하였다. 한 번의 실험 복제(replication)에는 총 60일 기간(워업 30일과 실험 30일)에 대한 시뮬레이션을 실행하였고, 총 5번의 복제를 진행하였다.

본 실험을 진행한 환경은 13th Gen Intel(R) Core(TM) i5-13600K, 3500Mhz, 14 Cores, 20 Logical Processors이고, 32GB 메모리를 사용하였다. AutoMod® x64 14.5을 이용하여 시뮬레이션을 구성하였으며, 총 실험 기간은 평균 125.365초 이다.

Table 4. Experimental environment and time

	Specification
Processor	13th Gen Intel(R) Core(TM) i5-13600K, 3500Mhz, 14 Cores, 20 Logical Processors
Memory	32GB
Simulation	AutoMod® x64 14.5
CPU time	125.365 sec

5.2 실험 시나리오

본 연구에서 제시하는 점수기반 작업배정 로직의 효율성을 확인하기 위해 구성한 실험 시나리오는 아래 Table 5에 정리된 바와 같다. 점수기반 작업배정 로직(Score)의 비교 대상으로는 선착순(FCFS), 최근접(NFS), 최단공정시간(shortest processing time : SPT) 작업배정 규칙을 사용하였다. 선착순 작업배정 규칙에서는 처음 작업 목록에 포함된 시간 순서대로 작업을 실행한다. 최근접 작업배정 규칙은 작업 목록에 존재하는 작업 중 야드 크레인으로부터의 거리가 가까운 작업을 우선적으로 실행한다. 최단공정시간 작업배정 규칙은 작업 목록에 존재하는 작업 중 소요시간이 가장 짧은 것으로 예상되는 작업을 먼저 처리한다.

Table 5. Scenarios and subscenarios with different number of cranes within two container blocks

Dispatching rules		Number of cranes		Cranes switch blocks				Cranes do not switch blocks	
		1	2	3	4	(1, 1)	(2, 2)		
FCFS	First come first served	Scenarios are created for each matching combination							
NFS	Nearest first served								
SPT	Shortest processing time								
Score	Score-based dispatching logic to minimize average turnaround time	Scenarios are created for each matching combination Further subscenarios are created to find the best weight combination $\alpha = [0, 0.25, 0.5, 0.75, 1]$ $\beta = [0, 0.25, 0.5, 0.75, 1]$ $\gamma = [0, 0.25, 0.5, 0.75, 1]$							
	Score-based dispatching logic to minimize maximum turnaround time								

모든 실험 시나리오에서 컨테이너 블록의 개수는 두 개로 동일하지만, 사용되는 야드 크레인의 대수는 한 대부터 네 대까지의 시나리오를 생성하였다. RTGC의 운영 특성과 같이 야드 크레인의 블록 간 이동이 허용되는 경우(switch blocks)와 그 반대의 경우(do not switch blocks)를 구분하였으며, 후자의 경우 각 컨테

이너 블록에서 한 대씩의 야드 크레인을 사용하는 경우(1, 1)와 두 대씩의 야드 크레인이 사용하는 경우(2, 2)로 시나리오를 구성하였다.

점수기반 작업배정 로직은 이를 구성하고 있는 각 변수(D, PT, WT)에 대한 가중치(α, β, γ)에 따라 본 연구에서 최소화하고자 하는 평균 및 최대 턴타임(TT) 성능지표가 달라진다. 이에 따라, 점수기반 작업배정 로직에 대해서는 크레인의 대수(1대~4대) 및 특성(블록 간 이동 가능 또는 불가능), 그리고 목표 성능지표(평균 또는 최대 턴타임)에 따른 가중치 조합을 탐색한다. 각 점수 변수는 0~1 사이의 숫자로 정규화된 상태이며, 이들에 곱해지는 각 가중치의 설정범위 또한 0~1 사이에서 탐색을 진행하였다.

5.3 실험 결과

1) 점수기반 작업할당 로직의 크레인 대수 및 가중치에 따른 턴타임 변화

아래 Table 6~9은 한 대 또는 세 대의 야드 크레인이 투입된 시나리오에서 점수기반 작업배정 로직의 각 변수에 대한 가중치(α, β, γ)를 달리하여 도출한 평균 턴타임과 최대 턴타임 결과 값을 제시하고 있다. 표에서 녹색은 낮은 턴타임(긍정적 결과), 적색은 높은 턴타임(부정적 결과)을 나타낸다. 거리(D)에 대한 가중치가 전혀 없는 경우($\alpha = 0$), 평균 및 최대 턴타임이 크게 증가하는 것으로 확인되었으며, 이를 최적 가중치 조합을 탐색하는 과정에서 제외하였다.

한 대의 크레인이 투입된 시나리오에서의 평균 턴타임을 최소화하기 위한 최적 가중치는 거리에만 집중한 ($\alpha = 1, \beta = 0, \gamma = 0$)임을 확인하였다(Table 6 참조). 최대 턴타임은 대기시간(WT)에 대한 가중치인 β 가 0일 때 값이 증가하고 1에 가까울수록 감소하는 경향을 나타냈으며, 최대 턴타임이 최소화되는 최적 가중치는 α 와 β 값이 동일하면서 $\gamma = 0$ 인 상황에서 나타난다(Table 7 참조).

세 대 크레인의 시나리오에서는 평균 턴타임을 최소화하는 최적 가중치가 ($\alpha = 1, \beta = 0.75, \gamma = 0.5$)로 도출되었으며, 평균 턴타임을 최소화하는 가중치 조합의 1순위부터 3순위까지 0의 가중치를 갖는 변수는 없었다(Table 8 참조). 즉, 해당 시나리오에서 세 개의 변수가 균형적으로 고려되어야만 평균 턴타임을 최소화할 수 있다. 특히, 변수 중 작업소요시간(PT)을 무시했을 때($\gamma = 0$) 평균 턴타임을 크게 증가하는 것으로 확인된다(Table 8 참조). 최대 턴타임을 최소화하는 가중치는 ($\alpha = 1, \beta = 1, \gamma = 0.25$)로 나타났다. 대기시간(WT)의 가중치 β 값이 너무 작거나 작업소요시간(PT)의 가중치 γ 값이 너무 큰 경우 최대 턴타임을 최소화하는 목표에 악영향을 주는 것으로 확인된다(Table 9 참조). 세 대의 크레인을 운영하는 시나리오에서는 대기시간 변수가 주요한 영향을 미치며, 거리 및 작업소요시간은 비슷한 수준의 중요도를 갖는다.

참고로, 해당 실험에서는 가중치들을 일정한 비율로 조합했을 때 동일한 결과를 도출하고 있음을 확인할 수 있다. 예를 들어, 크레인이 1대일 때, (α, β, γ)가 (0.25, 0.25, 0.25), (0.5, 0.5, 0.5), (0.75, 0.75, 0.75), (1, 1, 1)과 같은 비율을 갖는 경우에는 모두 675로 동일한 평균 턴타임을 나타낸다. 현재처럼 값을 특정하고 구간을 나누는 방식으로 실험을 진행하면 많은 경우 동일한 결과를 얻을 수 있으며, 전통적인 할당 방법과 유사한 결과를 얻을 수 있다. 더불어, 현재의 가중치 조합 탐색 방법은 특정 범위 내의 가중치만을 고려한 것이었지만, 향후 연구에서는 확률 경사하강법(stochastic gradient descent)과 유전 알고리즘(GA)을 도입하여 가중치 탐색 방법을 실험하고, 다양한 가중치 조합에 대한 성능을 평가하며 각 변수의 효과와 영향을 비교하고자 한다.

Table 6. Average turnaround time for different (α, β, γ) in 2-block 1-crane scenario (unit: sec)

$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1
0	615	637	722	712	821	0	615	636	637	669	722	0	615	638	632	637	677	0	615	618	636	627	637
0.25	655	675	728	731	733	0.25	656	643	645	670	682	0.25	638	628	650	655	654	0.25	642	634	620	628	645
0.5	757	766	775	764	793	0.5	655	696	675	675	728	0.5	649	647	664	669	675	0.5	656	645	643	648	645
0.75	803	806	831	774	898	0.75	694	713	745	709	783	0.75	655	675	671	675	690	0.75	657	667	649	667	650
1	828	836	823	883	917	1	757	728	766	749	775	1	697	711	716	730	721	1	655	670	696	662	675

(a) $\alpha = 0.25$ (b) $\alpha = 0.5$ (c) $\alpha = 0.75$ (d) $\alpha = 1$

Table 7. Maximum turnaround time for different (α, β, γ) in 2-block 1-crane scenario (unit: sec)

$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1
0	5090	6191	7836	6737	8428	0	5090	5923	6191	7117	7836	0	5090	7396	6509	6191	7140	0	5090	5546	5923	5458	6191
0.25	3345	4254	6548	4708	4564	0.25	3595	3373	4309	4737	5011	0.25	4074	3701	4108	4124	4737	0.25	3973	4456	4528	4124	4875
0.5	4546	4890	4378	4028	5225	0.5	3345	3483	4254	3783	6548	0.5	3679	3973	4441	3537	3960	0.5	3595	3626	3373	3741	4309
0.75	4964	5123	4790	5407	5467	0.75	3498	4509	3914	3882	5257	0.75	3345	3521	3356	4254	4618	0.75	3441	3542	3904	4934	4128
1	4822	5210	4888	6185	5793	1	4546	3444	4890	6843	4378	1	4282	4360	3546	4238	4679	1	3345	3542	3483	3466	4254

(a) $\alpha = 0.25$ (b) $\alpha = 0.5$ (c) $\alpha = 0.75$ (d) $\alpha = 1$

Table 8. Average turnaround time for different (α, β, γ) in 2-block 3-crane scenario (unit: sec)

$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1
0	401	388	388	391	393	0	401	381	388	386	388	0	401	383	386	388	385	0	401	387	381	389	388
0.25	398	384	388	391	393	0.25	403	385	385	386	389	0.25	400	385	390	383	387	0.25	406	383	383	386	383
0.5	402	384	384	389	393	0.5	398	382	384	388	388	0.5	401	384	388	383	386	0.5	403	384	385	387	385
0.75	400	385	386	391	391	0.75	396	382	382	387	388	0.75	398	383	387	384	389	0.75	397	387	379	387	384
1	400	380	388	385	394	1	402	380	384	388	384	1	404	385	383	382	385	1	398	384	382	381	401

(a) $\alpha = 0.25$ (b) $\alpha = 0.5$ (c) $\alpha = 0.75$ (d) $\alpha = 1$

Table 9. Maximum turnaround time for different (α, β, γ) in 2-block 3-crane scenario (unit: sec)

$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1	$\beta \backslash \gamma$	0	0.25	0.5	0.75	1
0	3379	2449	2548	2897	2772	0	3379	2341	2449	2590	2548	0	3379	2581	2289	2449	2502	0	3379	2960	2341	2415	2449
0.25	2089	2119	2703	2916	2772	0.25	2825	2316	2654	2421	2675	0.25	2891	2300	2921	2463	2499	0.25	2949	2471	2393	2657	2837
0.5	2097	2267	2488	2687	2335	0.5	2089	2342	2119	2851	2703	0.5	2516	2301	2351	2140	2487	0.5	2825	2228	2316	2790	2654
0.75	2221	2384	2159	2292	2499	0.75	2331	1994	2352	2324	2344	0.75	2089	2197	2375	2119	2394	0.75	2392	2295	2160	2600	2958
1	2130	2003	2159	2347	2512	1	2097	1980	2267	2272	2488	1	2385	2389	2161	2307	2312	1	2089	1973	2342	2087	2691

(a) $\alpha = 0.25$ (b) $\alpha = 0.5$ (c) $\alpha = 0.75$ (d) $\alpha = 1$

2) 점수기반 작업할당 로직의 크레인 대수에 따른 최적 가중치

본 연구에서 구성한 모든 시나리오에 대한 점수기반 작업배정 로직 가중치를 탐색한 결과는 아래 Table 10에 요약된 바와 같다. 최소화하려는 성능지표(평균 또는 최대 턴타임)에 따라, 그리고 크레인의 대수(1대~4대)에 따라 다른 가중치를 적용하는 것이 유리하다는 것을 표에서 확인할 수 있다. 사용하는 야드 크레인

인 대수가 한 대인 경우, 평균 턴타임을 줄이기 위해 작업과 크레인 사이의 거리(D)에 집중해야며, 최대 턴타임을 최소화하고자 하는 경우 거리(D)와 함께 대기시간(WT)을 같은 비율로 고려하면 된다. 두 대 이상의, 즉 다중 야드 크레인이 사용되는 경우에는 평균 및 최대 턴타임을 최소화하기 위해서는 거리(D), 대기시간(WT), 작업소요시간(PT)의 세 가지 변수를 모두 사용하는 것이 유리한 것으로 확인하였다.

Table 10. Best combination of (α, β, γ) found for score-based dispatching logic in different scenarios

Objective function	Number of cranes					
	Cranes switch blocks				Cranes do not switch blocks	
	1	2	3	4	(1, 1)	(2, 2)
Minimize average turnaround time	(1, 0, 0)	(1, 0.25, 0.5)	(1, 0.75, 0.5)	(0.25, 0.25, 0.75)	(0.75, 1, 0)	(0.5, 0.75, 0)
Minimize maximum turnaround time	(1, 1, 0)	(0.5, 1, 0.25)	(1, 1, 0.25)	(0.5, 1, 0.25)	(0.75, 0, 0.25)	(1, 0.75, 0.25)

3) 점수기반 작업할당 로직과 세 가지 작업배정 규칙 비교

본 연구에서 제시하는 점수기반 야드 크레인 작업배정 로직의 성능을 확인하기 위해 세 가지 작업배정 규칙과의 비교를 진행하였으며, 결과를 Figure 9에 제시하였다. 비교 대상으로는 선착순(FCFS), 최근접(NFS), 최단공정시간(SPT) 작업배정 규칙을 사용하였으며, 점수기반 작업배정 로직에는 앞서 Table 10에 정리된 상황별 최적 가중치를 적용하였다. Figure 9에는 점수기반 작업배정 로직의 두 가지 가중치 조합 즉, 평균 턴타임 최소화 가중치를 사용한 결과(S_{avg})와 최대 턴타임 최소화 가중치를 사용한 결과(S_{max})를 함께 표시하였다.

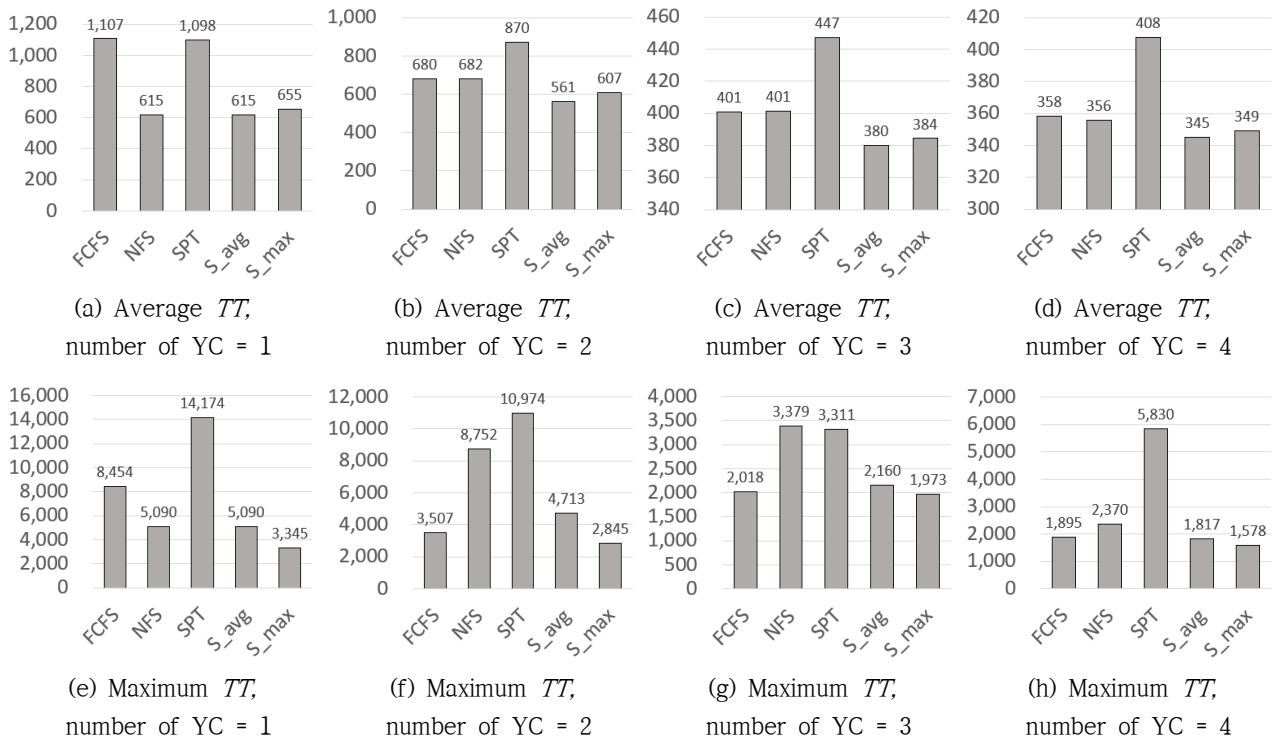


Figure 9. Score-based dispatching logic performance comparison against three dispatching rules

점수기반 작업배정 로직에서 한 대의 야드 크레인을 사용하여 평균 턴타임을 최소화하고자 하는 경우, 거리(D)에 대한 가중치만 사용하게 된다. 이에 따라, 평균 턴타임을 최소화 가중치를 적용한 결과(S_{avg})는

최근접(NFS)규칙에 의한 결과와 동일하게 나타났다. 그러나 다중 크레인이 사용될 경우(2대~4대), 점수기반 작업배정 로직이 다른 규칙에 비해 낮은 평균 턴타임을 도출하였다. 최대 턴타임을 최소화하는 가중치 조합을 적용한 경우에서도 점수기반 작업배정 로직의 효율성을 확인할 수 있다. 크레인의 대수에 상관없이 점수기반 작업배정 로직이 가장 짧은 최대 턴타임을 기록하였다. 특히, 크레인의 대수가 네 대일 때에는 평균 또는 최대 턴타임 가중치에 상관없이(S_avg 또는 S_max) 타 배정규칙에 비해 짧은 최대 턴타임을 도출하였다.

최대 턴타임을 최소화하는 가중치 조합을 사용한다면 평균 턴타임 최소화 가중치를 사용했을 때에 비해 증가된 평균 턴타임을 도출할 수 있다. 이러한 차이는 투입된 크레인의 대수가 많아질수록 줄어드는 경향이 확인된다(야드 크레인이 두 대일 때 약 8%, 세 대 또는 네 대일 때에는 약 1%의 차이).

4) 블록 간 이동 허용 여부에 따른 턴타임 변화

동일한 대수의 야드 크레인이 투입되더라도 크레인의 블록 간 이동 허용 여부에 따라 턴타임 성능지표가 달라질 수 있다. 본 연구에서는 블록 간 이동이 가능한 RTGC의 유연한 이동성의 장점을 정량적으로 확인하기 위한 추가적인 시나리오를 구성하였다. 실험을 위해 각 컨테이너 블록에서 한 대씩의 야드 크레인만을 고정적으로 사용하는 경우(1, 1)와 두 컨테이너 블록을 오갈 수 있는 두 대의 야드 크레인을 사용하는 경우를 비교하였으며, 두 배의 야드 크레인이 사용된 (2, 2)와 네 대의 야드 크레인의 상황을 비교하였다. 실험 결과를 Figure 10에서 확인할 수 있다.

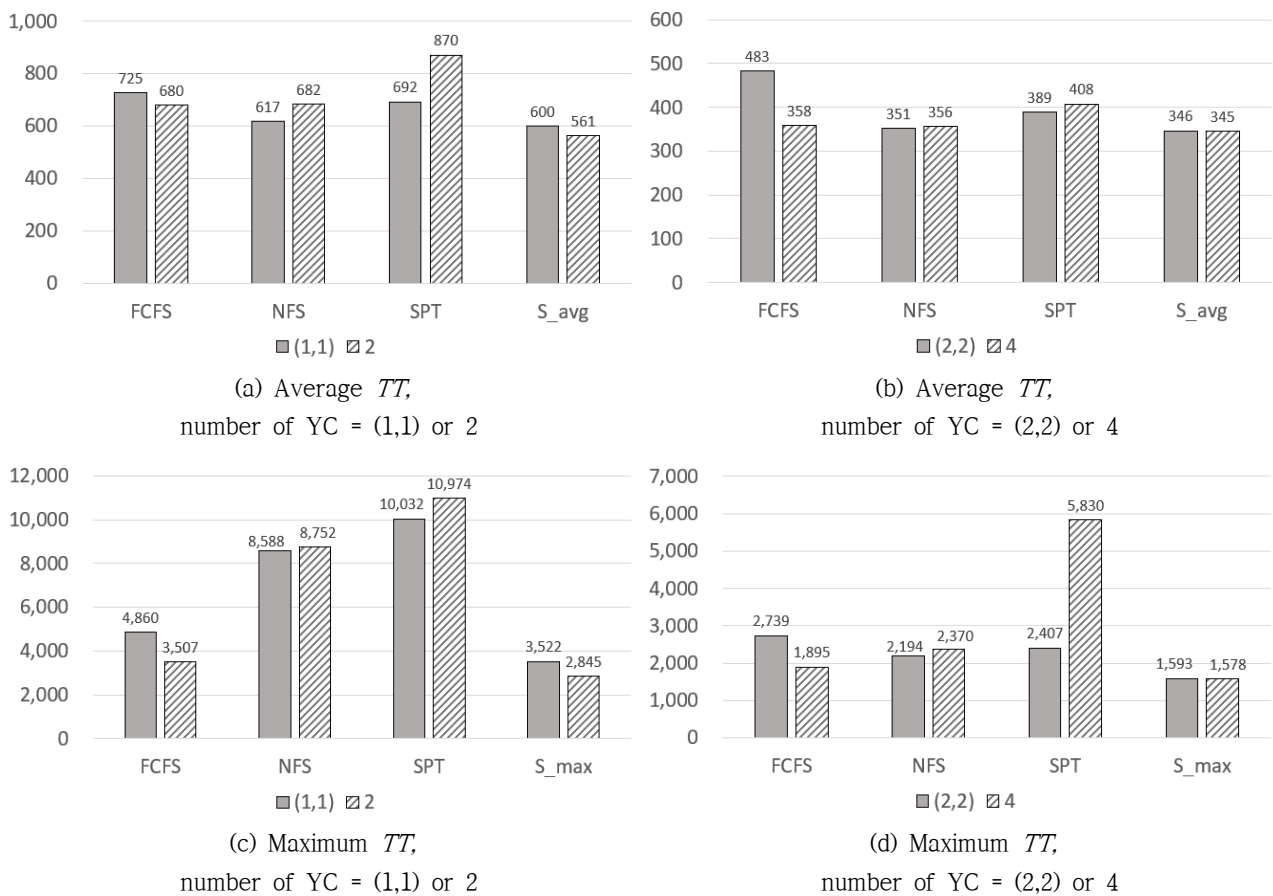


Figure 10. Performance comparison of scenarios, where cranes can switch blocks or cannot switch blocks

선착순(FCFS) 규칙 또는 점수기반 작업배정 로직이 사용되는 상황에서는 야드 크레인의 블록 간 이동이 가능해졌을 때 평균 및 최대 턴타임을 감소시킬 수 있는 것으로 확인되었다. 총 두 대의 크레인이 존재할 때 블록 간 이동은 선착순 규칙 하에서 평균 턴타임을 6.3%, 최대 턴타임을 27.8% 감소시켰으며, 점수기반 작업배정 로직 하에서는 평균 턴타임 6.4%, 최대 턴타임 8.2% 수준의 감소 효과가 나타났다. 총 네 대의 크레인을 가정했을 때에도 블록 간 이동이 선착순 규칙 및 점수기반 작업배정 로직에서 효과적인 것으로 확인되었다. 크레인이 네 대일 때 선착순 규칙에서는 평균 턴타임을 25.9%, 최대 턴타임을 34.5% 감소하였으며, 점수기반 작업배정 로직은 평균 및 최대 턴타임을 아주 약간(1% 미만) 감소시킬 수 있었다. 한편, 최근접(NFS) 및 최단작업시간(SPT) 규칙에서는 블록 간 이동을 허용하지 않는 편이 더 효율적인 운영 방식인 것으로 확인되었다.

6. 논의 및 결론

RTGC는 유연한 이동성(블록 간 이동)을 포함하여 여러 장점을 갖지만, 대부분 디젤 내연기관을 사용하고 있기 때문에 RMGC 대비 에너지 소모량과 유지비용이 높다는 단점이 있다(Sha et al., 2017). 그럼에도, RTGC 시장의 복합연간성장률이 4.7%(2022년에서 2030년)로 조사되는(Global Industry Analysts Inc., 2023) 등, 전 세계 항만에서 RTGC를 계속해서 운영할 것으로 예상된다. 이를 위해 기존 RTGC를 개조하는 자동화 전략(Korea Maritime Institute, 2020)과 함께, 전기 또는 하이브리드 전환을 통해 에너지 소모량과 탄소 배출량 감축 방안이 활발하게 연구되고 있다(Ding et al., 2021; Kusakana, 2021; Lin et al., 2022). RTGC의 특성을 반영한 야드 크레인 작업배정 로직 효율화를 도모하는 본 연구 또한 이러한 추세에 합치하는 것으로 판단된다.

본 연구에서는 항만 야드 크레인을 효율적으로 실시간 배정하기 위한 점수기반 작업배정(score-based dispatching) 로직을 제시하였다. 문헌연구를 통해 동적으로 변화하고 불확실성이 높은 항만의 환경에서는 최적의 작업계획(scheduling)에 비해 즉각적인 작업배정(dispatching) 방식이 현실적인 접근법임을 확인하였다. 본 연구는 RMGC와 비교되는 RTGC의 유연한 이동성(크레인의 블록 간 이동)을 적극적으로 검토하기 위해 두 개의 컨테이너 블록 내에 투입되는 크레인 대수가 최소 한 대부터 최대 네 대까지인 상황을 고려하였으며, 야드 크레인의 블록 간 이동 가능 여부에 따른 성능지표 차이를 확인하였다.

점수기반 작업배정 로직에서는 각 작업이 각 야드 크레인에 의해 실행되었을 때의 점수를 계산하며, 가장 우수한 점수를 갖는 작업이 선택되어 실행된다. 점수의 계산에는 야드 크레인과 작업 간의 거리, 작업의 대기시간, 예상 작업소요시간의 세 가지 변수(factor)가 사용된다. 각 변수에 개별 가중치를 곱한 합산으로 점수가 계산되며, 가중치는 다양한 시나리오에서 상황에 맞는 최적의 조합을 찾아 사용하게 된다.

각 변수의 Min-Max 정규화에는 실험을 통해 얻은 고정된 변수를 사용하였다. 그러나 항만 환경이 변화함에 따라 정규화 변수가 조정되어야 할 필요가 있다. 향후 연구에서는 민감도 분석 및 대안 정규화 기법에 대한 조사를 통해 데이터 정확도를 향상시키고자 한다.

점수기반 작업배정 로직의 성능을 확인하기 위해 AutoMod 시뮬레이션 실험을 진행하였다. 성능지표로는 평균 턴타임과 최대 턴타임을 사용하였으며, 비교 대상으로는 선착순, 최근접, 최단작업시간 작업배정 규칙을 선정하였다. 시뮬레이션 실험 결과, 다중 크레인(2대~4대)의 모든 시나리오에서는 점수기반 작업배정 로직의 성능이 타 규칙에 비해 우수한 것으로 확인되었다. 추가적인 실험을 통해 크레인의 블록 간 이동(RTGC의 특성)이 가능한 경우, 그렇지 않은 경우에 비해 성능지표를 개선할 수 있다는 사실을 확인하였다. 점수기반 작업배정 로직에서도 블록 간 이동에 의한 개선효과를 확인할 수 있었으며, 크레인의 대수가 증가할수록 개선의 정도는 감소하였다.

본 연구에서는 선착순, 최근접, 최단시간 작업배정 규칙과 함께 점수기반의 작업할당 로직을 비교했다. 그러나 동적인 항만 상황에 더 효과적으로 대응하기 위해 새로운 작업할당 로직들이 도입되고 있다. 향후 연구에서는 플래닝 윈도우 기반의 작업할당 로직과 A* 검색 알고리즘과 같은 다양한 알고리즘들을 점수기반의 작업배정 방식과 비교하여 성능을 평가해보고자 한다.

참고문헌

- Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20(1), 27-45. <https://doi.org/10.1080/00207548208947745>
- Briskorn, D., & Zey, L. (2018). Resolving interferences of triple-crossover-cranes by determining paths in networks. *Naval Research Logistics*, 65(6-7), 477-498. <https://doi.org/10.1002/nav.21819>
- Briskorn, D., & Zey, L. (2020). Interference aware scheduling of triple-crossover-cranes. *Journal of Scheduling*, 23(4), 465-485. <https://doi.org/10.1007/s10951-019-00634-6>
- Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014). Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research*, 235(2), 412-430. <https://doi.org/10.1016/j.ejor.2013.10.054>
- Chu, F., He, J., Zheng, F., & Liu, M. (2019). Scheduling multiple yard cranes in two adjacent container blocks with position-dependent processing times. *Computers and Industrial Engineering*, 136, 355-365. <https://doi.org/10.1016/j.cie.2019.07.013>
- Ding, Y., Yang, Y., Heilig, L., Lalla-Ruiz, E., & Voss, S. (2021). Deployment and retrofit strategy for rubber-tyred gantry cranes considering carbon emissions. *Computers and Industrial Engineering*, 161. <https://doi.org/10.1016/j.cie.2021.107645>
- Global Industry Analysts Inc. (2023). Rubber Tired Gantry (RTG) Cranes - Global Strategic Business Report. <https://www.researchandmarkets.com/reports/5304200/rubber-tired-gantry-rtg-cranes-global>
- Global Logistics Research Institute. (2009). Development of high-efficiency port cargo handling system technology.
- Guo, P., Wang, L., Xue, C., & Wang, Y. (2020). Dispatching Rules for Scheduling Twin Automated Gantry Cranes in an Automated Railroad Container Terminal. *Arabian Journal for Science and Engineering*, 45(3), 2205-2217. <https://doi.org/10.1007/s13369-019-04176-z>
- Guo, X., Huang, S. Y., Hsu, W. J., & Low, M. Y. H. (2008). Yard crane dispatching based on real time data driven simulation for container terminals. 2008 Winter Simulation Conference, 2648-2655. <https://doi.org/10.1109/WSC.2008.4736380>
- Huang, S. Y., Guo, X., Hsu, W. J., & Lim, W. L. (2012). Embedding simulation in yard crane dispatching to minimize job tardiness in container terminals. *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1-11. <https://doi.org/10.1109/WSC.2012.6465137>
- Huang, S. Y., Ya Li, & Guo, X. (2014). Yard crane dispatching to minimize vessel turnaround times in container terminals. *Proceedings of the Winter Simulation Conference 2014*, 1747-1758. <https://doi.org/10.1109/WSC.2014.7020024>
- Korea Maritime Institute. (2020). KMI international logistics weekly, volume 530. https://www.withlogis.co.kr/bbs/board.php?bo_table=research_weekly&wr_id=544
- Korea Ministry of Oceans and Fisheries. (2021a). Container cargo volume recovers to pre-COVID levels in 2021. Korea Ministry of Oceans and Fisheries.
- Korea Ministry of Oceans and Fisheries. (2021b). Smart port maintenance and management technology

- developmnet. Korea Ministry of Oceans and Fisheries. www.kimst.re.kr
- Korea Ministry of Oceans and Fisheries. (2022). Initiation of technology development for the introduction of domestic automated ports. Korea Ministry of Oceans and Fisheries.
- Kusakana, K. (2021). Optimal energy management of a retrofitted Rubber Tyred Gantry Crane with energy recovery capabilities. *Journal of Energy Storage*, 42. <https://doi.org/10.1016/j.est.2021.103050>
- Lin, S., Zhen, L., & Wang, W. (2022). Planning low carbon oriented retrofit of diesel-driven cranes to electric-driven cranes in container yards. *Computers and Industrial Engineering*, 173. <https://doi.org/10.1016/j.cie.2022.108681>
- Liu, W., Zhu, X., Wang, L., Yan, B., & Zhang, X. (2021). Optimization approach for yard crane scheduling problem with uncertain parameters in container terminals. *Journal of Advanced Transportation*, 2021. <https://doi.org/10.1155/2021/5537114>
- Liu, Y. F., & Lee, C. B. (2019). A study on the structural changes of import & export containers between South Korea and China ports. *Korean Logistics Research Association*, 29(2), 1-12. <https://doi.org/10.17825/klr.2019.29.2.1>
- Navis. (2023). Navis N4 terminal operating system. Navis. <https://dev-navis.azurewebsites.net/en/products/terminal-operations/n4-terminal-operating-system#>
- Navis Collaboration Center. (2019). Container terminal retrofits, what's holding you back? Navis Collaboration Center. <https://collaboration.navis.com/blogDetails?id=9061M000000Qg5UQAS>
- Ng, W. C., & Mak, K. L. (2005). Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29(3), 263-276. <https://doi.org/10.1016/j.apm.2004.09.009>
- Petering, M. E. H., Wu, Y., Li, W., Goh, M., & de Souza, R. (2009). Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals. *OR Spectrum*, 31(4), 801-835. <https://doi.org/10.1007/s00291-008-0142-7>
- Port Technology International. (2020). Automation in 2020 and beyond. Port Technology International. <https://www.porttechnology.org/news/automation-in-2020-and-beyond/>
- Sha, M., Zhang, T., Lan, Y., Zhou, X., Qin, T., Yu, D., & Chen, K. (2017). Scheduling optimization of yard cranes with minimal energy consumption at container terminals. *Computers and Industrial Engineering*, 113, 704-713. <https://doi.org/10.1016/j.cie.2016.03.022>
- Yu, D., Li, D., Sha, M., & Zhang, D. (2019). Carbon-efficient deployment of electric rubber-tyred gantry cranes in container terminals with workload uncertainty. *European Journal of Operational Research*, 275(2), 552-569. <https://doi.org/10.1016/j.ejor.2018.12.003>
- Yu, X., Tang, G., Guo, Z., Song, X., & Yu, J. (2018). Performance Comparison of Real-Time Yard Crane Dispatching Strategies at Nontransshipment Container Terminals. *Mathematical Problems in Engineering*, 2018, 1-15. <https://doi.org/10.1155/2018/5401710>

요약문

항만 자동화에 대한 필요성이 증대되고 있는데 항만의 타이어형 야드 크레인(rubber tired gantry crane, RTGC)의 유연성을 유지하면서 생산성을 올릴 수 있도록 개조하는(retrofit) 방법을 통해 자동화를 하는 전략이 주목받고 있다. RTGC는 특정 컨테이너 블록에 고정되지 않고 인접 블록으로의 이동이 가능하여 RMGC(Rail Mounted Gantry Crain)보다 그 유연성이 높다. 따라서 효율적인 운영을 위해서는 크레인 대수 변화와 같은 상황에 대처할 수 있는 작업배정 방법론이 요구된다. 본 연구에서는 항만 야드 크레인을 효율적으로 실시간 배정하기 위한 점수기반 작업배정(score-based dispatching) 로직을 제시하였다. 해당 로직에서 점수는 다양한 변수에 적절한 가중치를 곱한 합산으로 계산되며, 가장 우수한 점수를 갖는 작업을 우선적으로 실행된다. AutoMod 시뮬레이션을 통해 타 작업배정 규칙과의 성능지표를 비교한 결과, 점수기반 작업배정 로직의 우수성을 확인할 수 있었다. 또한, RTGC의 블록 간 이동이 턴타임과 같은 항만 야드의 주요 성능지표를 개선할 수 있다는 사실을 확인하였다.

주제어: 컨테이너 터미널 자동화, 야드 크레인 작업배정 규칙, 고무 타이어 갠트리 크레인, 다중 야드 블록, 다중 야드 크레인