

컨테이너 터미널의 불확실한 환경 하에서의 ALV 주행 계획 수립방안

김정민* · 이동균** · † 류광렬

*부산대학교 대학원, ** (주)사이버로지텍 Core Asset 팀, † 부산대학교 정보컴퓨터공학부

Routing of ALVs under Uncertainty in Automated Container Terminals

Jeongmin Kim* · Donggyun Lee** · † Kwang Ryel Ryu

* Graduate School of Pusan National University, Busan 609-735, South Korea

** Core Asset Team, Cyberlogitec Co., Ltd., Busan 600-755, South Korea

† School of Computer Science and Engineering, Pusan National University, Busan 609-735, South Korea

요 약 : 무인 자가 운반 하역차량(Automated Lifting Vehicle, ALV)은 자동화 컨테이너 터미널에서 컨테이너를 수송하는 무인 차량의 하나로 자가 하역 및 수송 능력을 가지고 있다. 여러 대의 ALV를 이용해 컨테이너를 효율적으로 수송하기 위해서는 ALV가 컨테이너의 이송 작업을 시작할 때마다 최소 시간에 주행이 가능한 경로를 실시간으로 찾을 수 있어야 한다. 또한 차량 간의 충돌 및 교착 상태 발생 시 스스로 해결이 불가능한 무인 차량의 특성 상 이러한 충돌 및 교착을 막을 수 있도록 차량이 목적지까지 가기 위해 점유해야 하는 점유 영역과 그 점유 시간을 결정하여 이를 겹치지 않도록 주행 계획을 수립하여야 한다. 하지만 주행 계획 수립을 위한 ALV의 점유 영역에서의 점유 시간 계산은 교통 상황에 따른 주행 시간의 변화나 주행 경로 상에 작업을 수행하는 크레인의 작업 상황의 불확실성 때문에 정확한 추정이 어렵다. 본 논문에서는 개미 집단 최적화 기법을 기반으로 이러한 ALV 도착 시간의 불확실성을 고려한 ALV 주행 계획 수립방안을 제안한다. 시뮬레이션 실험을 통해 제안 방안이 불확실한 환경에서 효율적으로 좋은 경로를 찾아냄을 확인하였다.

핵심용어 : 무인 자가 운반 하역차량, 주행계획 수립, 불확실성, 개미 집단 최적화, 시뮬레이션

Abstract : An automated lifting vehicle(ALV) used in an automated container terminal is a type of unmanned vehicle that can self-lift a container as well as self-transport it to a destination. To operate a fleet of ALVs efficiently, one needs to be able to determine a minimum-time route to a given destination whenever an ALV is to start its transport job. To find a route free from any collision or deadlock, the occupation time of the ALV on each segment of the route should be carefully scheduled to avoid any such hazard. However, it is not easy because not only the travel times of ALVs are uncertain due to traffic condition but also the operation times of cranes en route are not predicted precisely. In this paper, we propose a routing method based on an ant colony optimization algorithm that takes into account these uncertainties. The result of simulation experiment shows that the proposed method can effectively find good routes under uncertainty.

Key words : Automated lifting vehicle, Routing, Uncertainty, Ant colony optimization, Simulation

1. 서론

자동화 컨테이너 터미널에서 무인 자가 운반 하역차량(Automated Lifting Vehicle, ALV)은 안벽 크레인(Quay Crane, QC)과 장치장의 자동화 크레인(Automated Stacking Crane, ASC) 간의 컨테이너 운반을 담당한다. 적하 작업 시 ALV는 ASC가 장치장에서 집어낸 컨테이너를 지정된 QC까지 운반하고, 양하 작업 시에는 QC가 선박에서 내린 컨테이너를 지정된 장치장의 ASC까지 운반하여 전달한다. ALV 운영의 일차적 목표는 컨테이너를 전달해 주거나 받는 과정에서 QC나 ASC가 ALV를 기다리는 일이 가능한 없도록 함으로써

터미널의 안벽 생산성을 최대화하는 것이다. 이를 위한 중요한 필요조건 중의 하나는 매 ALV가 컨테이너를 운반할 때마다 그 소요시간을 최소화하는 주행 계획을 실시간에 수립할 수 있어야 한다는 것이다. 운반시간을 최소화하려면 출발지에서 목적지까지 다른 ALV나 크레인들과의 충돌 혹은 교착 상태를 피하는 과정에서 발생하는 대기 시간까지 포함하여 총 주행 소요 시간이 최소가 되는 경로를 찾아야 한다. 이를 위해서는 주행경로 전체에 걸쳐 ALV가 점유하며 지나가는 영역을 잘게 나눈 뒤 그 각 단위 영역을 어떤 시점에 예약하거나 점유하며 통과할지를 사전에 면밀히 계획해야 한다. (Bae et al., 2011).

* 대표저자: 정희원, jeongminkim.islab@gmail.com 051)510-3531

** 정희원: finht@naver.com 051)440-1849

† Corresponding author : 종신회원, krryu@pusan.ac.kr 051)510-2453

ALV는 AGV(Automated Guided Vehicle)와 달리 스스로 컨테이너를 집거나 내릴 수 있을 뿐 아니라 컨테이너를 적재한 상태에서 지상에 1단으로 놓인 다른 컨테이너 위로 지나갈 수도 있다. 크레인의 입장에서는 차량과 컨테이너를 주고받을 때 크레인 아래 교환 지점(Transfer Point)에 차량이 도착해 있어야만 작업이 가능한 것이 아니어서 대기 시간이 줄어드는 장점이 있다. 그러나 이러한 운영상의 유연성 때문에 ALV의 주행계획 수립은 오히려 더 어려워지는 측면이 있다. ALV는 AGV보다 그 운영상의 자유도가 높아서 다른 차량 및 크레인과의 충돌이나 교착 가능성 또한 더 높아지기 때문이다. 앞서 언급한 주행경로 상의 단위 영역들에 대한 예약이나 점유의 적정 시점도 예상치 않은 교통 상황 변화로 주행 시간이 달라지면 따라서 변경되어야 하고, 주행 경로 상의 전방에서 컨테이너를 내리거나 올리고 있는 크레인이 있을 경우 그 작업 소요 시간을 정확하게 예측할 수 없기 때문에 주행 계획 시 단위 영역의 적정 예약 및 점유 시점 결정이 매우 어렵게 된다.

자가 운행 차량의 주행 계획 수립에 관한 기존 연구는 크게 차량이 목적지에 도착하기 까지 경유해야 하는 지점을 결정하는 연구와 장비 사이의 충돌 및 교착 상태를 방지하는 연구로 나눌 수 있다. 경유 지점 결정에 관한 연구로는 휴리스틱을 이용한 방안(Imai et al., 2007; Jin et al., 2005, Ko and Egbelu, 2003; Krishnamurthy et al., 1993)과 Dijkstra, A*와 같은 그래프 탐색 알고리즘을 이용한 방안(Choe et al., 2013; Gawrilow, et al., 2008; Möhring et al., 2005)이 있다. 휴리스틱 기반의 주행 경로 결정 방안은 최적 주행 경로를 보장하기 어렵다는 문제가 있고, 그래프 탐색 알고리즘을 이용한 방안들은 주행 환경의 그래프가 복잡해질수록 최적 경로 도출에 많은 시간이 걸리는 문제가 있다. 장비 사이의 충돌 및 교착 상태 방지에 관한 연구에는 Moorty et al.(2003)이 AGV의 주행공간을 존(Zone)으로 나누고 AGV의 이동을 존 단위로 파악하여 AGV의 교착상태를 해소하였으며, Taubin et al.(1998)은 Petri Net을 이용하여 AGV의 교착상태를 감지 및 해소하였고, Möhring et al.(2005)은 경유하는 간선에 대해 점유 시간을 설정하여 충돌 및 교착을 회피하였다. Bae et al.(2011)은 차량의 예정 주행 궤적을 작은 단위 영역들로 나누고 각 단위 영역별 예약 및 점유 시간을 계획함으로써 충돌과 교착을 피했다. 그러나 이들 연구에서는 교통 상황이나 크레인의 작업 지연 등으로 인한 불확실성을 전혀 고려하지 않았다. Choe et al.(2013)은 이미 결정된 주행 경로 상의 예약 및 점유 계획이 예상 외의 상황 변화로 인해 부적절해질 경우 주행 중인 모든 ALV의 경로 계획을 다시 수립함으로써 충돌 및 교착 문제를 해결하였다. 그러나 이렇게 단순한 방안은 불필요하게 많은 계산 시간을 요구한다.

본 논문에서는 제한된 시간 내에 비교적 최적에 가까운 경로를 탐색할 수 있는 개미 집단 최적화(Ant Colony Optimization, ACO) 기반의 주행경로 탐색 방안(Chen et al., 2009; Montemanni et al., 2005)을 이용하여 주행 경로를 탐색

한다. 불확실성에 대처하는 방안으로는 주행 경로 상의 단위 영역이나 TP를 예약하는 시간에 적절한 여유를 두는 방안을 제안한다. 여유시간이 충분치 못하면 경로계획을 전반적으로 다시 수립해야 하는 경우가 자주 발생하여 계산 부담이 가중되는 반면 여유시간이 너무 길어지면 재계획을 거의 하지 않게 되는 대신 다른 차량들의 주행을 지나치게 막음으로써 전체 수송 효율이 떨어지게 된다. 본 연구에서는 적절한 여유시간을 실험적으로 결정하였다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 자동화 컨테이너 터미널 및 ALV의 주행계획 수립 문제를 상세히 설명한다. 3장에서는 작업 환경의 불확실성을 고려하여 주행경로 계획을 수립하는 방안을 제안한다. 4장에서는 제안 방안을 검증하기 위한 실험을 소개하고 그 결과를 분석한다. 마지막으로 5장에서는 결론을 내리고 향후 연구에 대해 기술한다.

2. 자동화 컨테이너 터미널과 ALV의 작업

본 논문에서는 Fig. 1과 같은 수직 자동화 컨테이너 터미널에 대해서 다루고 있다. 수직 자동화 컨테이너 터미널에서 선적작업을 위한 영역으로 선박이 접안하는 안벽(Quay), 수출입 컨테이너를 임시로 쌓아두는 장치장(Stacking Yard) 그리고 안벽과 장치장 사이의 에이프런(Apron) 영역으로 구성된다. 에이프런 영역은 QC레인(QC lane), 블록레인(Block lane) 그리고 QC레인과 블록레인을 연결하는 수직레인(Vertial lane)으로 구성된다. QC레인은 QC 아래를 수직으로 주행 가능한 레인으로 각각의 QC레인에는 QC와 ALV가 컨테이너 교환이 발생하는 버퍼 영역인 컨테이너 TP가 있다. 수직레인에는 대기 지점이 있다. 대기지점은 작업을 할당 받지 못하거나 목적 TP를 확보하지 못한 ALV가 잠시 대기하는 작은 버퍼 영역이다. 본 논문에서는 TP와 대기지점을 같은 의미로 정차지점이라고 부른다.

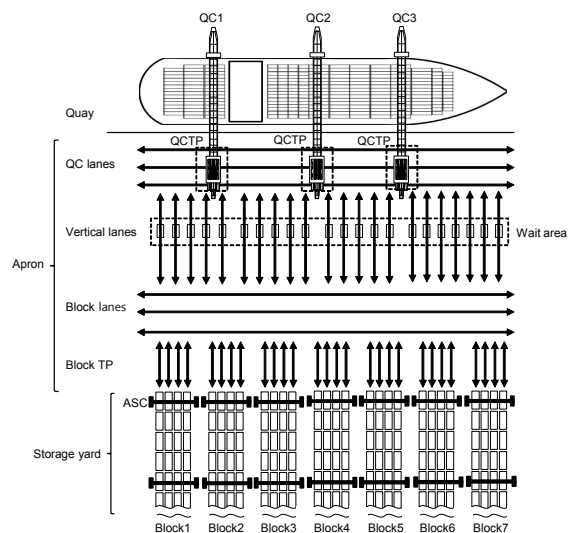


Fig. 1 The layout of an automated container terminal

자동화 컨테이너 터미널에서 QC와 ASC는 각각 안벽과 장치장에서 컨테이너를 운반하는 역할을 담당한다. 그리고 ALV는 에이프런 영역 안에서 QC와 ASC를 오가며 컨테이너를 운반하는 역할을 담당하며 Fig. 2와 같이 컨테이너의 자가 하역이 가능하고 컨테이너가 적재된 TP 위를 지나갈 수 있는 특징이 있다.

ALV의 작업은 선박으로부터 양하된 수입 컨테이너를 안벽에서 장치장으로 운반하는 양하작업과 수출 컨테이너를 선박에 적하하기 위한 적하작업으로 나누어진다. 각 작업의 흐름은 다음과 같이 진행된다. 먼저 양하작업은 ALV가 수입 컨테이너를 받기 위해서 수직레인 상에 존재하는 대기영역에서 기다리다가 QC가 하역작업을 완료하면, ALV는 컨테이너가 놓여진 TP를 목적 TP로 결정하여 주행한 후, 스스로 컨테이너를 차체에 싣는다. 그런 다음, ALV는 컨테이너를 적재할 장치장 블록의 해측 끝에 위치한 TP 중 하나를 목적 TP로 결정하여 주행한 후, 해당 TP에 컨테이너를 내리면 양하작업이 끝난다. 적하작업은 양하작업과 반대로 ASC가 하역작업을 완료하면, 컨테이너가 놓여진 TP를 목적 TP로 결정하여 주행한 후, 스스로 컨테이너를 차체에 싣는다. 그런 다음, QC 아래에 위치한 TP를 목적 TP로 결정하여 주행한 후, 해당 TP에 컨테이너를 내리면 적하작업이 완료된다.

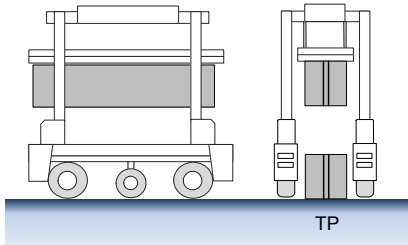


Fig. 2 Front and side view of the ALV

3. ALV의 주행계획 수립 방안

3.1 그래프 형태의 주행 네트워크 표현

본 논문에서는 자동화 컨테이너 터미널에서 ALV가 안벽과 장치장 사이를 오가며 컨테이너를 운반하는 에이프런 영역을 노드와 에지를 가지는 그래프 형태로 일반화 하여 표현한다.

그래프의 노드는 Fig. 3과 같이 에이프런 영역에 존재하는 주행레인과 정차지점(TP와 대기지점)으로 구성되는데, QC레인과 수직레인의 경우 각 레인 상에 존재하는 정차지점(TP와 대기지점)을 기준으로 나누어 각각 다른 레인으로 간주하고 블록레인의 경우 QC레인과 같은 길이로 나누어 각각 다른 레인으로 간주한다.

그래프의 에지는 두 노드 간의 연결 관계를 나타낸다. 예를 들어, Fig. 4에서 노드 1은 노드 2와 노드 3과 연결되어 있기 때문에 그래프의 에지가 있다고 가정하고 노드 1과 노드 4와는 직접 연결되어 있지 않기 때문에 에지가 없다고 가정한다.

본 논문에서는 이러한 방식으로 ALV가 주행하는 에이프런 영역을 그래프 형태로 표현하고 이를 주행 네트워크라 한다.

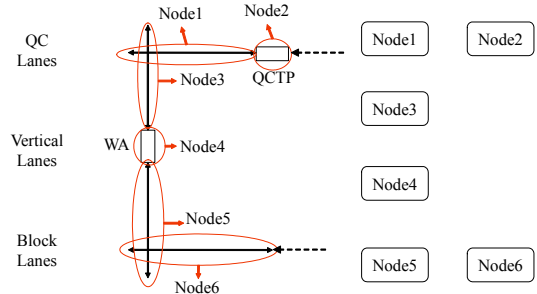


Fig. 3 Graph representation of apron(Node)

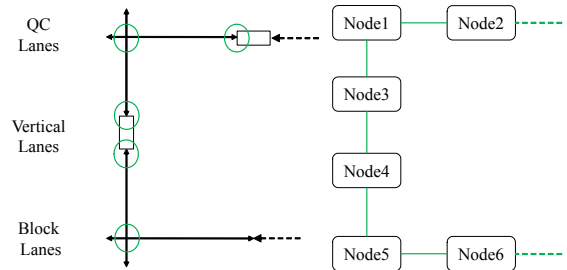


Fig. 4 Graph representation of apron(Edge)

3.2 ACO 알고리즘을 이용한 주행경로 결정 방안

본 절에서는 주행네트워크 상에서 ACO(Ant Colony Optimization) 알고리즘과 교착상태 회피 개념을 적용하여 교착상태를 회피하면서, 최소시간으로 목적 TP에 이르는 주행 경로를 결정하는 방법에 대해서 살펴본다.

ACO는 실제 개미들이 집에서 먹이까지 가장 짧은 경로를 찾는 방식을 모방하여 그래프 상에서 최적 경로를 찾는 알고리즘이다. 개미들은 먹이를 향해 가는 동안 자신이 지나간 각 경로에 페로몬을 분비하고 나중에 지나가는 개미들은 경로에 남아 있는 페로몬 양을 가지고 먹이까지의 이동경로를 결정한다. ACO 알고리즘은 페로몬 갱신 방법에 따라 AS(Ant System)와 ACS(Ant Colony System) 알고리즘으로 나눌 수 있다(Chen et al., 2009). AS는 개미가 이동경로를 결정할 때 다른 개미들의 이동경로에 대한 고려 없이 단순히 자신의 입장에서 비용이 적게 소요되는 이동경로만을 결정하기 때문에 지역 최적화에 빠질 확률이 높다. 반면에, ACS는 개미가 이동경로를 결정하면 지역적으로 방문한 경로에 대해서 페로몬의 양을 변경하여 나중에 지나가는 개미가 다른 이동경로로 방문할 수 있게 하므로 다양한 경로를 탐색할 수 있어 AS에 비해 지역 최적화에 빠질 확률이 낮다. 본 논문에서는 ACS 알고리즘을 적용하여 ALV의 주행경로를 결정한다.

ACS 알고리즘은 정해진 시간 동안에만 주행경로를 탐색

한다. 주행경로 탐색과정은 후보 해를 생성 하는 작업과 페로몬을 갱신하는 작업으로 나눌 수 있다. 먼저 해를 생성하는 작업은 현재 노드와 이웃 노드들 중에서 이동 가능한 노드들을 선별하는 작업과 선별된 노드들 중에 페로몬과 휴리스틱 정보를 이용해서 확률적으로 다음에 이동 할 노드를 선택하는 작업으로 나누어진다. 먼저 이동 가능한 노드를 선별하기 위해서 본 논문에서는 Table 1과 같은 조건을 가지고 이동 불가능한 노드를 제외시킨다.

Table 1 Conditions for nonexpandable nodes

1. Visited node
2. Destination node of other ALV
3. Physically immovable node from current node

위와 같은 조건으로 이동 가능한 이웃 노드들을 선별 한 후, 본 논문에서는 식 (1)을 적용하여 다음에 이동 할 노드를 확률적으로 선택한다.

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \{ [\tau_l]^\alpha [\eta_l]^\beta \} & \text{if } q \leq q_0; \\ P & \text{otherwise;} \end{cases} \quad (1)$$

$$P = \frac{[\tau_i]^\alpha [\eta_i]^\beta}{\sum_{l \in N_i^k} [\tau_l]^\alpha [\eta_l]^\beta}, \quad \text{if } j \in N_i^k$$

식 (1)에서 N_i^k 는 현재 노드 i 와 이웃한 노드 리스트이고 τ_i 은 현재 노드 i 와 이웃한 노드 l 에 대한 페로몬의 양을 나타낸다. q 는 $[0,1]$ 사이의 무작위 값을 가지는 파라미터이고, $q_0 (0 \leq q_0 \leq 1)$ 는 Greedy하게 τ_l 와 η_l 값이 높은 노드를 선택할 것인지 아니면 τ_l 와 η_l 를 이용하여 확률적으로 노드를 선택할 것인지 결정하는 파라미터이다. η_l 은 현재 노드 i 와 이웃한 노드 l 에 대한 예상 주행 시간에 반비례하는 값으로 식 (2)와 같이 계산한다.

$$\eta_l = \frac{1}{g(l) + h(l)} \quad (2)$$

$$h(l) = \frac{|x_i - x_{\text{destination point}}| + |y_i - y_{\text{destination point}}|}{v}$$

$g(l)$ 은 시작노드부터 노드 l 까지의 주행 시간이고 $h(l)$ 은 노드 l 에서 목적지까지의 주행시간을 추정 한 값이다. 노드의 주행시간은 노드 형태에 따라 다르게 계산한다. 먼저 노드의 형태가 주행라인 일 경우에는 주행시간 추정 알고리즘을 적용하여 주행라인의 끝점에 도착하는 시간을 계산한다. 노드의 형태가 정차지점 일 경우에는 정차지점을 떠나는 시간을 기준으로 삼는다. 차량이 정차지점에서 떠나는 시간은 교착상태

회피 개념을 적용하여 계산된다. 즉, 해당 정차지점의 예약기간을 추정한 후, 우선적으로 해당 정차지점을 예약 할 수 있는 다른 ALV의 예약기간과 겹치는지를 파악한다. 예약기간이 겹치는 경우에는 다른 ALV가 정차지점을 해지하는 시간을 기준으로 해당 정차지점의 예약기간을 다시 계산하고, 겹치지 않는 경우에는 원래의 예약기간을 그대로 사용한다. 정차지점의 예약기간은 이전 정차지점을 떠나는 시간부터 현재 정차지점에 도착하는 시간으로 가정한다.

페로몬 τ_l 을 갱신하는 작업은 크게 지역 페로몬을 갱신하는 작업과 전역 페로몬을 갱신하는 작업으로 나누어진다. 지역 페로몬은 하나의 개미가 해를 생성 할 때마다 자신이 방문한 노드들에 페로몬을 감소시켜서 다른 개미가 자신이 방문하지 않은 노드로 해를 생성 할 수 있게 만드는 역할을 한다. 지역 페로몬의 갱신은 식 (3)과 같이 이루어진다.

$$\tau_l = (1 - \rho)\tau_l + \frac{1}{\hat{t}} \quad (3)$$

ρ 는 페로몬 증발 비율이고, \hat{t} 는 해에 대한 추정된 주행시간을 나타낸다. 전역 페로몬은 시간이 지날수록 주행시간이 짧은 주행경로에 페로몬을 누적시키는 역할을 한다. 전역 페로몬의 갱신은 한 개미의 집단이 모두 해를 찾으면 수행되는데, 그 중 최소시간으로 목적지로 주행 할 수 있는 가장 좋은 해에 대하여 다음의 식을 이용하여 갱신한다.

$$\tau_l = (1 - \rho)\tau_l + \frac{1}{\hat{t}^{best}}, l \in S^{best} \quad (4)$$

$$\tau_l = (1 - \rho)\tau_l, l \in A, l \notin S^{best}$$

식에서 S^{best} 는 개미 집단이 생성한 해들 중에서 가장 좋은 해를 나타내고, \hat{t}^{best} 은 해당 해에서 추정된 주행시간을 나타낸다. 그리고 A 는 그래프 상에 존재하는 모든 노드의 집합을 의미한다.

위와 같이 해를 생성하고, 페로몬을 갱신하는 과정을 제한 시간 동안 반복적으로 수행하면서 획득한 해들 중 목적 TP까지의 주행시간이 가장 짧은 해를 최종 주행경로로 결정한다.

3.3 교착상태 및 충돌 방지 방안

차량의 교착상태 및 충돌을 방지하기 위한 방안으로 그래프의 노드 형태에 따라 서로 다른 충돌 및 교착상태 방지 알고리즘을 적용한다. 그래프를 구성하는 노드 중 주행라인에 해당하는 노드에 대해서는 Bae et al.(2011)의 방법을 적용하였다. 점유 영역 기반의 이 방법에서는 차량의 궤적을 바탕으로 점유 영역을 생성한 후 점유 영역 별 점유 시간을 관리한다. 만일 같은 점유 영역이나 해당 점유 영역과 겹치는 부분이 있는 점유 영역에 대해서는 동 시간대에 점유할 수 없게 함으

로써 충돌 및 교착 상태를 방지한다.

그래프를 구성하는 노드 중 정차지점에 해당하는 노드에는 Choe et al.(2013)이 제안한 방법을 적용한다. 이 방법은 교착 상태를 사전에 회피하는 방안과 회피가 불가능한 교착 상태의 경우 교착 상태를 감지하여 이를 해소하는 방안으로 구성되어 있다. 교착상태를 회피하는 방안은 차량이 정차지점을 예약한 순서에 따라 정차지점의 예약기간을 설정하면서 먼저 예약한 차량의 예약 기간과 겹치지 않는 경우에만 예약을 허용하는 것이다. 교착 상태 해소 방안으로는 만일 작업의 불확실성으로 인해 예약 스케줄이 어긋나면서 차량 사이에 교착 상태에 빠진 경우에는 차량이 이동하고자 하는 목적지에 따른 의존 관계를 계산하여 사이클이 발생하는 경우 사이클에 들어 있는 모든 차량의 주행 경로를 다시 결정한다.

3.4 불확실성을 고려한 정차지점 예약기간 추정

앞선 3.3 절에서 추정된 정차지점의 예약스케줄이 주행 도중 어긋나면서 교착상태가 발생 할 수 있고, 이를 해결하기 위해서 정차지점의 예약 스케줄과 어긋나게 정차지점이 예약되어 있는 경우 주행 중인 모든 ALV의 경로를 다시 설정하는 방법을 소개하였다. 하지만 이러한 방안은 주행 중인 모든 ALV의 주행경로를 다시 변경하는데 많은 계산비용이 소요된다. 또한, 뒤에 주행한 ALV에 의해서 앞서 주행한 ALV가 불필요하게 우회해서 목적 TP로 주행하는 경우가 발생 할 수 있다. 따라서 본 논문에서는 Fig. 5와 같이 정차지점을 예약하는 시간의 상한 값과 정차지점을 떠나는 시간의 하한 값을 추정하여 보다 길게 정차지점의 예약기간을 구함으로써 예약 스케줄이 조금 어긋나더라도 교착 상태에 빠지지 않게 만든다.

i 번째 정차지점을 예약하는 시간의 시작 시점 B_{p_i} 는 이전 정차지점에서 가장 빨리 떠날 수 있는 시점인 $D_{p_{i-1}}$ 값으로 설정하고, 예약이 끝나는 시간의 상한과 하한 값은 이전 정차지점을 떠나는 시점의 상한과 하한 값 $\overline{D_{p_{i-1}}}$, $\underline{D_{p_{i-1}}}$ 값을 기준으로 현재 정차지점까지 도착하는 시간을 추정한 값 \hat{t}_{p_i} 에 일정한 매개변수 α 와 β 를 곱하여 구한다. Fig. 6은 정차지점 예약기간을 추정하는 방안을 나타낸다.

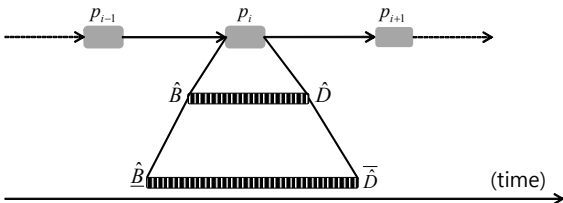


Fig. 5 Concept of stopping point reservation under uncertain environment

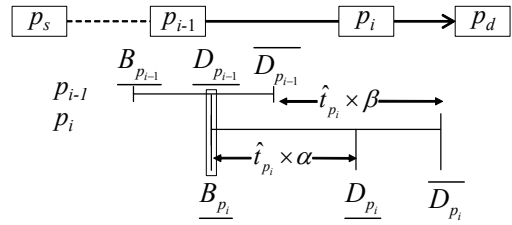


Fig. 6 Estimating reservation period of stopping point

위와 같은 방법으로 정차지점의 예약기간을 추정하고 해당 정차지점을 앞서 예약 가능한 다른 ALV의 예약기간과 겹치는지를 파악한다. 만약 다른 ALV의 예약기간과 겹치는 경우 다른 ALV가 해당 정차지점을 떠날 때까지 진입이 불가능하므로 이전 정차지점을 p_{i-1} 을 떠나는 시간의 하한 값 $\underline{D_{p_{i-1}}}$ 을

해당 정차지점을 앞서 예약 할 수 있는 다른 ALV가 예약을 해지하는 시간으로 수정한다. 그 후, 바뀐 $\underline{D_{p_{i-1}}}$ 값을 기준으로 해당 정차지점을 떠나는 시간의 하한 값을 새로 계산한다.

이 때, 정차지점을 떠나는 시간의 하한 값은 해당 정차지점을 앞서 예약 할 수 있는 ALV가 해당 정차지점에 수행하는 작업에 따라 다르게 계산된다. 먼저 ALV가 해당 정차지점을 통과하는 경우에는 Fig. 6과 같은 방안으로 계산한다. 하지만 ALV의 작업이 해당 정차지점에 컨테이너를 내리는 작업 일 경우에는 식 (5)를 적용하여 크레인이 해당 컨테이너를 집어가는 작업시간과 이전 정차지점에서 현재 정차지점까지 도착하는 시간을 비교하여 계산한다.

$$\overline{D^i} = \text{Max}(\overline{D^{i-1}} + \overline{T_{i-1}^i}, \overline{D_{-n}^i} + \overline{CPT} + \overline{T_{i-1}^i}) \quad (5)$$

$\overline{D^{i-1}}$ 은 이전 정차지점을 떠나는 시간의 하한 값을 나타내고, $\overline{T_{i-1}^i}$ 은 이전 정차지점에서 현재 정차지점까지 도착하는 시간의 하한 값을 나타낸다. $\overline{D_{-n}^i}$ 은 해당 정차지점을 앞서 예약 할 수 있는 다른 ALV가 정차지점을 떠나는 시간을 나타낸다. \overline{CPT} 는 크레인의 작업시간을 나타내고 식 (6)을 적용하여 계산한다.

$$\overline{CPT} = \text{Crane Processing Time} + \text{Avg Delay} + 2\sigma \quad (6)$$

4. 실험 결과

본 장에서는 3장에서 제안한 ALV의 주행방안의 성능을 시뮬레이션 기반의 비교실험을 통해 확인하고자 한다.

4.1 실험환경

본 절에서는 제안한 주행방안의 성능을 확인하기 위해서 Table 2와 같은 가상의 터미널 환경을 설정하였다. 그리고 Table 3는 실험에 사용된 ALV의 사양을 나타낸다.

Table 2 Terminal configuration

Parameter	Value
Number of berth	3
Number of QCs per berth	3
Number of TPs per QC	3
Number of blocks per berth	7
Number of TPs per block	4
Number of vertical lanes per berth	42
Total number of nodes	641

Table 3 ALV specification

Straight speed	Rotational speed	Acceleration	Deceleration
6 m/s	2 m/s	0.64 m/s ²	1.55 m/s ²

본 논문에서는 제안한 주행방안의 성능을 확인하기 위해서 시뮬레이션 실험을 수행하였다. ALV의 수는 12대에서 30대까지 6대씩 변화시켜 가며 각각 실험하였다. 그리고 매 실험마다 3000번의 랜덤 주행을 실시한 후, 10회씩 반복한 결과의 평균을 구하였다.

4.2 불확실성을 고려한 정차지점 예약기간 추정의 효과

첫 번째 실험은 불확실성을 고려하여 정차지점의 예약기간을 추정한 방안의 유용성을 검증하기 위한 실험이다. 본 실험에서는 A*알고리즘과 휴리스틱 함수로 식 (2)의 $H()$ 을 적용해 최소시간 주행경로를 결정하였고 주행경로 탐색에 소요되는 시간은 없다고 가정한다. 주행시간 추정의 상하한 값을 계산하기 위해서 사용되는 파라미터 α 와 β 는 α 의 값이 작고 β 의 값이 크면 TP 점유시간이 늘어나면서 평균 TP 대기시간이 점차 길어지고, α 와 β 의 값이 비슷할수록 대기시간이 짧아지지만 리라우팅 빈도가 높아지는 경향이 있어 실험을 통해서 각각 1.5와 3.0으로 정하였다. 비교방안으로는 정차지점의 해지시간을 모르는 경우(Conservative reservation)와 정차지점의 해지시간을 아는 경우(Exact reservation)가 있다.

Table 4는 ALV의 수를 12대에서 30대까지 6대씩 증가시키 나가면서 평균주행시간을 측정된 결과를 나타낸다. 제안방안인 Strict reservation이 비교방안들 보다 주행 시간 단축에 유리한 것을 확인할 있다. Table 5는 차량이 TP에서 대기한 평균 시간을 나타내고 Table 6은 평균 주행거리를 나타낸다. 평균 예약대기시간의 경우 비교방안들에 비해 10초에서 30초까지 대기 시간이 단축되었음을 볼 수 있고 이것이 주행 시간의

단축으로 이어졌음을 알 수 있다. 또한 평균주행거리의 결과를 살펴보면 제안방안이 비교방안들의 차이가 평균 거리의 1~2% 정도로 주행 시간을 단축하기 위해 많은 거리를 돌아서 가는 것이 아님을 알 수 있다. Exact reservation의 경우 Conservative reservation보다 평균 주행 시간 측면에서 나은 성능을 보였지만 주행 시간 추정 시 QC 작업으로 인해 차량이 통과하지 못하는 경우의 시간을 고려하지 못해 QC와의 간섭이 발생할 경우 차량이 TP에서 대기하면서 제안 방안에 비해서는 평균 주행 시간이 오래 걸림을 알 수 있다.

Table 4 Average travel time between different TP reservation methods

Unit: seconds			
Number of ALVs	Strict reservation	Conservative reservation	Exact reservation
12	112.90	124.76	116.62
18	123.63	139.86	128.28
24	138.49	158.36	146.56
30	155.66	177.98	164.86

Table 5 Average waiting time on TP between different TP reservation methods

Unit: seconds			
Number of ALVs	Strict reservation	Conservative reservation	Exact reservation
12	0.05	10.30	2.68
18	0.22	17.82	4.86
24	0.73	27.96	10.04
30	1.40	38.40	16.93

Table 6 Average travel distance between different TP reservation methods

Unit: meters			
Number of ALVs	Strict reservation	Conservative reservation	Exact reservation
12	383.32	384.82	384.85
18	383.10	381.60	381.81
24	386.36	384.21	386.09
30	390.42	384.94	389.85

4.3 ACO 알고리즘의 적용 효과

두 번째 실험은 ACO 알고리즘과 최소시간 주행경로를 찾는 A*알고리즘의 성능을 비교 분석한다. 두 알고리즘의 성능 비교는 탐색 시간을 주행시간에 포함하지 않은 경우와 포함한 경우로 나누어 실험하였다. 그리고 두 알고리즘 모두 불확실성을 고려하여 정차지점의 예약기간을 계산하였다. A*알고리즘의 경우 휴리스틱 함수로 식 (2)의 $H()$ 을 적용하였고, ACO의 경우 Table 7과 같이 설정 값을 적용하였다. 실험에 사용

한 설정 값 중 α , β 는 ACO 알고리즘을 이용하여 차량의 라우팅 결정 문제를 연구한 Chen et al. (2009)의 연구 결과를 참고 하였으며, ρ 의 경우 값이 너무 커지면 주행 시간이 빠른 경로에 개미가 집중되어 전역 최적 경로를 찾기가 어려워지고 값이 너무 작으면 주어진 시간 안에 최적해로의 수렴이 어려워지는데 본 논문의 탐색 대상 그래프는 복잡도가 주어진 시간에 비해 낮은 편이라 개미의 수와 더불어 실험적으로 결정한 결과 각각 0.05, 7로 설정되었다.

Table 7 parameter values of ACO algorithm

Parameter	Value
α (exponential value of pheromone)	1
β (exponential value of expected travel time)	1
ρ (evaporation ratio of pheromone)	0.05
Number of ants	7
Time limit of the routing	0.5/1/2/3/4/5

ACO 알고리즘은 제한시간 동안만 주행경로를 탐색한다. 하지만 이때, 어느 시간이 최적의 설정 값인지 알 수 없으므로 본 논문에서는 ALV가 30대 운행하는 환경에서 0.5초부터 5초까지 주행경로 탐색시간을 변경해가며 최적의 탐색시간을 찾았다. Table 8은 탐색시간 변경에 따른 평균 주행시간을 나타낸다. 1초까지는 탐색한 경로의 개선으로 인한 시간 단축이 탐색에 걸린 시간을 상회하여 평균 주행 시간이 감소하였으나 2초 이후로는 탐색에 걸리는 시간이 경로 개선으로 인한 시간 단축보다 길어져 평균 주행 시간이 증가하는 추세를 보인다. 탐색 시간을 주행 시간에 포함한 경우 주어진 탐색 시간보다 조금 더 주행시간이 증가하는 것을 볼 수 있는데 이것은 동시에 여러 주행 경로를 계산하는 경우 해당 요청 수만큼 대기하는 시간이 길어지기 때문이다.

Table 9는 A*알고리즘과 주행경로 탐색에 소요된 시간을 주행 시간에 반영하지 않고 ACO 알고리즘의 성능을 비교 분석한 결과를 나타낸다. ACO의 경우 30대에서 가장 좋은 성능을 보인 2초의 탐색시간을 기준으로 주행 경로를 구하였다. 차량 대수가 많아질수록 최적 탐색 결과를 보장하는 A*알고리즘이 ACO 알고리즘에 비해 주행 시간을 단축하는 것을 볼 수 있는데 이는 차량 수가 많을수록 최적화의 여지가 더 많음을 나타낸다. 하지만 이 결과에는 탐색에 소요된 시간이 반영되지 않아 실제 주행 시간과는 괴리가 있다.

Table 8 Average travel time of ACO algorithm between different time for search

Unit: seconds		
Time limit	Travel time(exclude routing time)	Travel time(include routing time)
0.5	161.63	162.07
1	159.44	160.14
2	158.45	160.74
3	158.72	161.85
4	160.16	164.89
5	162.71	169.13

Table 9 Average travel time between different graph search algorithms(exclude routing time from travel time)

Unit: seconds		
Number of ALVs	ACO	A*
12	113.05	112.90
18	125.56	125.33
24	141.28	140.10
30	158.45	154.57

Table 10은 탐색시간이 주행시간에 포함되는 환경에서의 A* 알고리즘과 ACO 알고리즘의 평균 주행시간을 비교한 결과이다. ACO의 탐색시간은 Table 8에서 가장 좋은 성능을 보인 1초로 설정 하였다. 실험 결과, 12대에서 18대까지는 두 알고리즘의 평균주행시간이 비슷하나 24대부터는 평균주행시간이 A* 알고리즘에 비해 짧아지면서 30대에서는 A* 알고리즘에 비해 약 5% 정도 평균 주행시간의 단축을 확인하였다.

이러한 주행 시간의 차이는 Table 11의 주행경로 탐색을 위한 대기시간을 보고 분석이 가능하다. ACO 알고리즘은 주행경로 탐색시간을 1초로 정하면 탐색지연 시간이 일정한 반면, A* 알고리즘은 탐색시간의 제한이 없어 최적 주행경로 탐색에 많은 시간이 소요되는 경우가 있다. 실험을 통해 확인 한 바로는 주행경로 탐색에 최고 561초까지 소요되는 것을 확인 하였다. 결론적으로 주행경로 탐색시간도 주행시간에 포함되는 실제 환경에서는 본 논문에서 ACO를 이용하여 주행경로를 탐색하는 것이 효과적임을 확인 할 수 있다.

마지막으로 본 논문에서는 위의 모든 실험을 수행하면서 단 한 번의 교착상태도 발생하지 않았음을 확인하였다.

Table 10 Average travel time between different graph search algorithms(include routing time into travel time)

Unit: seconds		
Number of ALVs	ACO	A*
12	113.77	113.42
18	127.16	126.31
24	142.22	145.35
30	160.14	168.87

Table 11 Average waiting time for routing between different graph search algorithms

Unit: seconds		
Number of ALVs	ACO	A*
12	1.02	0.52
18	1.05	0.98
24	1.08	5.25
30	1.12	13.42

5. 결론

본 논문에서는 자동화 컨테이너 터미널에서 ALV가 주행계획 수립을 위해 Choe et al.(2013)의 충돌 및 교착 방지를 위한 주행 계획 수립 방안을 기반으로 컨테이너 터미널의 불확실한 요소를 감안하여 정차지점의 예약기간을 추정하는 방안을 제안하였다. 그리고 ACO 알고리즘을 적용하여 최소시간 주행경로를 결정하는 방안에 대해서 제안하였다.

시뮬레이션을 이용한 비교실험 결과 불확실성을 고려한 정차지점의 예약을 통해 정차지점을 보다 정확하게 예약하고 보다 최소시간으로 목적지에 이르는 주행경로를 설정한다는 것을 확인 하였다. 그리고 ALV가 정차지점을 예약하는 과정에서 어떠한 교착상태도 발생하지 않음을 확인하였다.

향후 연구로는, 앞서 본 논문에서는 불확실한 요소를 고려하여 정차지점의 예약기간을 추정하기 위해서 두 가지 설정변수를 사용하는데, 컨테이너 터미널의 상황 변화를 감지하여 유동적으로 이 설정변수 값을 재 변경하는 방안에 대한 연구가 필요하다. 그리고 본 논문에서는 주행경로를 결정하기 위해서 미시적 관점으로 자신의 주행시간이 최소가 되는 경로를 탐색한다. 하지만 이는 거시적인 관점에서 보면 오히려 성능에 안 좋은 영향을 미칠 수도 있다. 따라서 거시적 관점에서 전체 ALV의 주행시간을 단축시키기 위해서 주행라인과 정차지점을 동적으로 재조정하는 방안에 대한 연구가 필요하다.

후 기

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

본 논문은 BK21플러스, IT기반 융합산업 창의인력사업단에 의하여 지원되었음.

References

- [1] Bae, H. Y., Choe, R., Park, T. J. and Ryu, K. R.(2011), "Comparison of Operations of AGVs and ALVs in an Automated Container Terminal", Journal of Intelligent Manufacturing, Vol. 22, No. 12, pp. 413-426.
- [2] Chen, P, Huang. H and Dong, X.(2009). "An Ant Colony System Based Heuristic Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pickup", Computers and Operation Research, Vol. 36, No. 12, pp. 3215-3223.
- [3] Choe, R., Lee, D. G. and Ryu, K. R.(2013), "Routing method for automated lifting vehicles considering the reservation schedule of transfer points", The International Conference on Tourism, Transport, and Logistics 2013
- [4] Gawrilow, E., Köhler, E., Möhring, R., and Stenzel, B.(2008), Mathematics - Key Technology for the Future, Springer Berlin Heidelberg Press, pp. 165-177.
- [5] Imai, A., Nishimura, E. and Current, J.(2007), "A Lagrangian relaxation-based heuristic for vehicle routing with full container load", European Journal of Operational Research, Vol. 176, pp. 87-105.
- [6] Jin, G. G., So, M. O. and Lee, H. S.(2005), "Job Deployment and Dynamic Routing for Container AGVs", Journal of the Korean Society of Marine Engineering, Vol. 29, No. 3, pp. 369-376.
- [7] Ko, K. C. and Egbelu, P. J.(2003), "Unidirectional AGV Guidepath Network Design: A heuristic Algorithm", International Journal of Production Research, Vol. 41, No. 10, pp. 2325-2343.
- [8] Krishnamurthy, N., Batta, R. and Karwan, M.(1993), "Developing Conflict-Free Routes for Automated Guided Vehicles", Operations Research, Vol. 41, pp. 1077-1090.
- [9] Möhring, R., Köhler, E., Gawrilow, E., and Stenzel, B.(2005), Conflict-free Real-time AGV Routing. Operations Research Proceedings 2004 SE - 3, Springer, pp. 18 - 24.

- [10] Montemanni, R., Gambardella, L. M., Rizzoli, A. E. and Donati, A. V.(2005), "An ant colony system for Dynamic vehicle routing problem", Journal of Combinatorial Optimization, Vol. 10, No. 4, pp. 327-343
- [11] Moorthy, R.L., Wee, H. G. and Ng, W. C.(2003), "Cyclic deadlock prediction and avoidance for zone-controlled AGV system", International Journal of Production Economics, Vol. 83, March, pp. 309-324.
- [12] Taubin, A., Kondratyev, A., and Kishinevsky, M.(1998), "Deadlock Prevention using Petri Nets and Their Unfoldings", International Journal of Advance Manufacturing Technology, Vol. 14, No. 10, pp. 750-759.

원고접수일 : 2014년 06월 16일
심사완료일 : 2014년 10월 28일
원고채택일 : 2014년 10월 30일