한국시뮬레이션학회 논문지
Vol. 17, No. 3, pp. 85-94 (2008. 9)

컨테이너 터미널에서의 AGV의 교착방지를 위한 주행영역 예약 표 작성방법

전수민[1] · 김갑환[1†]

Constructing a Reservation Table for Automated Guided Vehicles in Port Container Terminal

Su Min Jeon · Kap Hwan Kim

ABSTRACT

Automated Guided Vehicles (AGVs) are important equipment in automated container terminals. To utilize AGVs for transporting containers from a position to another, the deadlock is a serious problem that must be solved before they are deployed to real operations. This study assumes that traveling area for AGVs are divided into a large number of grid-blocks and, as a method of traffic control, grid-blocks are reserved in advance during the travel of AGVs. The purposes of the reservation are to make the room between AGVs and to prevent deadlocks. Because the size of an AGV is much larger than the size of a grid-block on guide paths, this study assumes an AGV may occupy more than one grid-block at the same time. This study suggests a method for constructing a table of reservation schedules by using a simulation. A sensitivity analysis was conducted to evaluate the performance of the reservation method in this study.

Key words : Container terminal, AGV, Deadlock

요 약

본 연구는 컨테이너 터미널에서 사용될 수 있는 교착방지를 위한 알고리즘 개발을 목표로 하였다. 기존의 교착방지 알고리즘은 많은 계산시간을 소요한다는 단점을 극복하기 위하여 본 연구에서는 하나의 경로에 대해서 예약스케줄을 작성하여 두고 반복적으로 사용하는 예약 스케줄 표 방식을 제안하였다. 시뮬레이션 실험 결과 예약 스케줄 표 방식은 그때그때의 주행경로에 대한 예약스케줄링을 위하여 계산시간이 필요 없기 때문에 AGV의 실시간 운영을 위하여 유용한 방식임을 알 수 있었다.

주요어 : AGV, 컨테이너 터미널, 교착방지

1. 서 론

자동화 컨테이너 터미널의 핵심 기술은 컨테이너의 하역, 운송, 보관, 분류 등의 터미널 내의 물류 기능을 무인 자동화 하는 것이다. 본 연구에서는 터미널 내 운송기능을 담당하는 무인 운반차(AGV: Automated Guided Vehicle) 시스템 에서의 AGV간의 교착방지를 위한 예약방법을 제안하고자 한다. 컨테이너 터미널에서 사용되는 AGV는 그 크기가 대형이고 많은 수의 AGV가 동시에 주행을 하기 때문에 차량간의 혼잡을 야기시킨다. 이를 피하기 위하여 보다 효율적으로 주행 영역을 사용하는 것이 무엇보다 중요하다.

AGV 주행을 제어하는 교통통제 시스템으로 존 제어 방식[1-3, 5-12]과 그리드 제어 방식[4]이 있다. 존 제어방식은 주어진 AGV 주행경로를 따라 일정구간의 존을 만들어서 AGV를 통제하는 방식이며 하나의 존에 한 대의차량만이 들어 갈 수 있다. 그리드 제어방식은 주행영역을 바둑판처럼 나누고 한 대의 차량은 복수개의 그리드를 동시에 점유할 수 있다. 본 연구의 AGV 교통통제 방식은 그리드

* 본 연구는 2008년 정부(교육과학기술부)로부터 지원받아 수행되었음(지역거점연구단육성사업/차세대물류IT기술연구사업단).
2008년 7월 23일 접수, 2008년 9월 16일 채택
[1] 부산대학교 산업공학과
주 저 자 : 전수민
교신저자 : 김갑환
E-mail; kapkim@pusan.ac.kr

방식이며 주행 시 필요한 그리드를 사전에 예약하면서 주행한다.

차량의 교통통제 시 가장 중요한 문제는 차량들 간의 교착상태 해결이다. 따라서 차량주행 시 발생하는 교착상태를 예측하고 방지하는 방법에 대한 기존연구를 살펴보면 다음과 같은 연구들이 있다. Lee and Lin[5]은 단일 방향 패스 네트 워크에서 차량이 한 존에서 다음 존으로 진행할 때 페트리 넷의 reachability를 조사 하여 교착가능성을 예측하고 방지하는 알고리즘을 제안하였다. Yeh and Yeh[10]는 존 컨트롤 하에서의 단일 방향 패스 네트워크에서 그래프 모델을 사용하여 교착을 예측하고 회피하는 알고리즘을 제안하였다. Rajeeva et al.[7]는 수평 배치 야드와 선석을 오가는 AGV 주행에서 생기는 Cyclic deadlock을 예측하고 회피하기 위한 알고리즘을 제안하고 시뮬레이션을 통해 제시한 알고리즘을 검증하였으며, 성능에 대한 실험을 수행하여 그 결과를 제시하였다. Reveliotis[8]는 존 컨트롤을 사용하는 양 방향 패스 네트워크에서 교착이 없는 안전한 주행경로를 구하는 방법을 제시하였다. Banker's algorithm을 AGV시스템에 맞게 변형하여 경로의 안전성을 조사하는 방법으로 사용하였다. 이재용과 서윤호[1]는 제조 생산환경에서 실시간 교착상태 해결 알고리즘이 포함된 AGV 시뮬레이터를 소개하고 실험을 통하여 검증하였다. Samia and Castaga[9]는 차량간의 충돌을 해결하기 위하여 주행문제가 발생될 수 있는 후보 AGV들에 노드 점유권을 우선으로 부여해 주는 RVRAA 알고리즘을 제안하였다. Zeng and Hsu[12]의 연구에서는 그리드 형태와 유사하게 그물망 형태(mesh-linked topology)로 AGV주행영역을 나누었다는 측면에서는 그리드 제어방식과 유사하지만 그리드의 의 크기가 차량의 크기보다 크게 설계되었다는 점에서 존 제어방식에 속한다고 할 수 있다. Evers and Kopper[2]는 semaphore라는 차량의 수용능력을 가진 신호 장치를 이용하여 효율적인 교통제어 방안을 제시하였다. Duinkerken et al.[11]는 교통제어 컨트롤러 TRACES를 개발 하여 시뮬레이션을 통한 성능 검증을 하였다. Matthias et al.[6]은 AGV 주행에서 발생되는 교착을 실시간 탐지하는 방법을 제시하고 작업처리 순서를 바꾸거나 작업 할당을 수정하는 방식으로 동시에 자원을 선점하려는 것을 회피하여 교착을 해결하고자 하였다. Grunow et al.[3]은 AGV multi-load dispatching 문제에서 발생하는 교착을 해결하기 위한 알고리즘을 제시하고 시뮬레이션을 통한 검증을 수행하였다. Kim et al.[4]은 AGV의 교착방지를 위한 알고리즘을 제시하였는데, 운행경로를 만들 때 실시간 상황을 고려하여 교착을 방지할 수 있는 동적인 그리드 예약스케줄을 작성하는 방식을 제안하였다.

본 연구에서는 Kim et al.[4]의 알고리즘을 이용하되, 예약스케줄을 작성하기 위하여 소요되는 계산시간을 줄이기 위하여 교착으로 방지할 수 있는 예약 스케줄을 사전에 작성하여 테이블에 저장하여 두고 주행 시에는 단순히 예약스케줄을 읽어 오는 방식으로 차량을 운행하는 방법을 제안하고자 한다. 이를 위하여서는 시뮬레이션을 통하여 교착에 관한 많은 라우팅 자료를 수집한 후 이를 바탕으로 예약 스케줄 테이블을 작성하는 방안에 관한 연구가 필요하다.

자동화 컨테이너 터미널에서 사용되는 장비 2장에서는 연구의 대상이 되는 컨테이너 터미널에서의 AGV주행 시 발생하는 교착 현상에 대해서 설명한다. 3장에서는 시뮬레이션을 이용한 교착을 방지할 수 있는 예약 스케줄 표도출방식을 제안하고 4장에서는 시뮬레이션의 실험을 통한 검증을 통하여 결론을 맺는다.

2. 컨테이너 터미널에서의 AGV 교착현상과 예약 스케줄을 이용한 주행 표현

자동화 컨테이너 터미널에서 사용되는 장비는 대표적으로 세 종류로 구분한다. 선석에서 컨테이너의 양하와 적하 작업을 담당하는 안벽 크레인(QC: Quay Crane)과, 장치장에서의 컨테이너의 반입과 반출작업을 담당하는 자동화 트랜스퍼 크레인(ATC: Automated Transfer Crane) 그리고 선석과 장치장 사이를 오가면서 컨테이너를 이송하는 무인 반송차(AGV: Automated Guided vehicle)이다. 터미널의 생산성을 향상시키기 위해서는 이들 장비간의 연계적인 작업이 중요하며, 특히 선석과 장치장의 작업을 연결하는 AGV의 효율적인 주행은 터미널 운영의 생산성을 향상시킬 수 있는 주요 요소로 볼 수 있다.

그림 1은 컨테이너 터미널에서의 AGV의 주행경로를 나타낸다. 대부분의 자동화 터미널에서 사용하는 AGV는 무궤도 주행(Free-Ranging) AGV로서 기계적인 한계 내에서는 어떠한 경로로도 운행이 가능하다. 일반적으로 AGV운행을 위하여 가상의 AGV운행경로를 결정해 두고 운반명령이 있는 경우, 운반명령의 출발지와 도착지에 해당되는 경로를 따라 AGV가 주행하게 된다. 본 연구에서는 AGV의 주행영역을 모듈단위영역으로 나누었으며 각 모듈은 여러 개의 그리드들로 구성되어 있다. 모듈을 구성하고 있는 그리드의 크기는 차량의 크기보다 작기 때문에 차량 주행 시 여러 개의 그리드를 점유하면서 주행한다고 가정하였다.

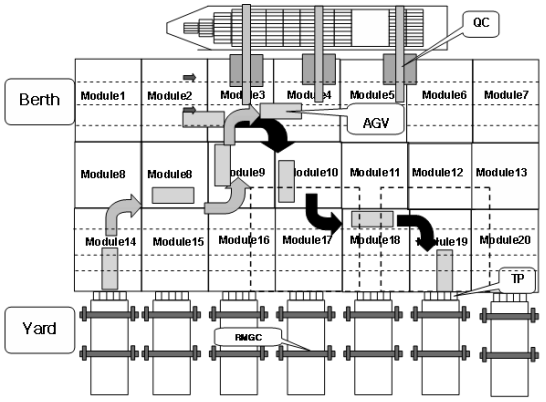

그림 1. 컨테이너 터미널에서의 AGV 주행 레이아웃

2.1 교착현상

차량은 정해진 궤도가 아닌 차량의 출발지와 도착지를 고려해 자유롭게 주행을 할 수 있도록 하였다. 따라서 차량의 주행의 효율성은 높일 수 있지만 여러 대의 차량이 동시에 주행할 때, 복잡하고 다양한 형태의 경로로 주행할 수 있어 차량 간 교착 발생 가능성이 많다. 그림 1의 모듈3에서의 차량주행을 예로 들어 보면 각 차량의 주행경로가 인접하여 차량 간에 동일한 영역을 공유하고 있음을 알 수 있다. 차량 간 같은 점유영역이 있다는 것은 차량 간 교착이 발생될 수 있음을 의미한다. 그림 2는 그림 1의 모듈3에서 발생 되는 교착의 형태를 예시하고 있다.

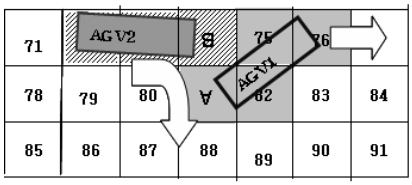

그림 2. 교착발생

그림 2의 상태에서 각 차량이 다음 경로로 이동하기 위해 필요한 그리드를 예약을 하려 할 때 AGV1은 B를 예약 하려 하고 동시에 AGV2는 A를 예약하려 한다면 교착이 발생하게 된다. 교착 현상의 원인은 주행 시 필요한 예약 영역을 확보하지 못하여 생기는 경우이다. 주행 중 예약 영역을 확보 하지 못한 차량은 대기되고 이때 주행 중인 차량이 대기한 차량의 영역을 다음 경로로 이동을 위해 예약하려 한다면 교착이 발생하는 것이다. 이와 같은 교착 상태는 해당 주행경로의 예약 스케줄을 조정하여 교착을 방지해야 한다.

2.2 예약 스케줄을 이용한 주행 표현

예약 스케줄은 차량의 실제 주행 경로를 반영하여 만들어 진다. 그림 3의 차량의 순차적인 주행경로를 도식화해서 나타내 보면 표 1과 같다.

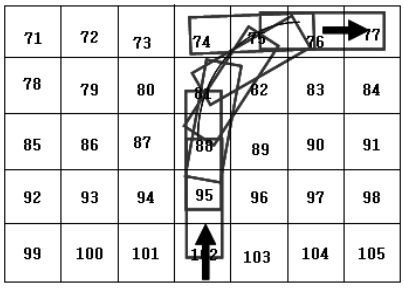

그림 3. 차량의 주행 경로의 예

표 1. 그림 3의 경로스케줄

모듈번호	주행단계	점유 그리드
3	1	102,95,88
3	2	95,88,81
3	3	85,88,81,82
3	4	88,82,81,75
3	5	88,82,81,76,75
3	6	82,81,76,75
3	7	76,75,74
3	8	77,75,76

표 1은 주행단계 별로 AGV가 점유하는 그리드들을 나타낸 것이다. 표 2는 표 1의 경로 스케줄을 기초로 작성한 예약 스케줄을 나타낸다.

표 2. 표1의 예약 스케줄

모듈번호	주행단계	예약 스케줄		
		예약 그리드	예약 해제 그리드	예약 예정 그리드
3	1	102,95,88		81
3	2	95,88,81	102	82
3	3	95,88,81,82		75
3	4	88,82,81,75	95	76
3	5	88,82,81,76,75		88
3	6	82,81,76,75	88	74
3	7	76,75,74	82,81	77
3	8	77,75,76	74	

예약 스케줄은 다음 단계로 주행 시 필요한 그리드만 예약 한다고 가정하였다. 예약 스케줄은 진입하려는 모듈

의 우선순위 테이블의 그리드 간 예약 우선순위 관계에 위배 되지 않도록 생성해야 한다. 어떤 차량의 라우트(Z1 → Z2)는 예약스케줄에 따라 우선순위 테이블에(Z1, Z2)의 값을 1을 넣는다. 이때 다른 차량의 라우트(Z2 → Z1)는 예약 스케줄에 따라 우선순위 테이블에 넣어야 할 값이(Z2, Z1) = 1 이지만, 이미 우선순위 테이블엔 Z1이 Z2보다 우선순위를 높게 두었기에 예약 스케줄과 일치하지 않게 되어 교착이 발생 한다. 이러한 경우 우선순위테이블과 일치하게 예약 스케줄을 재 계획한다면 교착을 해결할 수 있다. 표 3은 표 2를 반영한 3번 모듈의 우선순위 테이블을 나타낸 것이다. 우선순위테이블에서 (i, j)칸의 값이 1인 경우에는 그리드 i가 먼저 예약된 상태에서 그리드 j보다 우선순위가 높다는 것을 의미한다.

표 3. 3번 모듈의 우선순위 테이블

	73	74	75	76	77	78	79	80	81	82	88	89	95	96	102	103
73																
74																
75		1		1	1						1					
76		1			1						1					
77																
78																
79																
80																
81		1	1	1	1					1	1					
82		1	1	1	1						1					
88		1	1	1	1				1	1	1					
89																
95		1	1	1	1				1	1	1					
96																
102		1	1	1	1				1	1	1					
103																

3. 시뮬레이션을 이용한 예약 스케줄 도출 방식

AGV의 주행경로는 컨테이너의 양하 및 적하 작업을 수행하는 안벽크레인의 위치와 해당되는 컨테이너가 장치될 위치나 인출될 야드 내의 위치에 따라 주행의 시작위치와 종점이 결정된다. AGV가 하나의 운반작업을 완료하면 새로운 운반작업이 할당되고 이 운반작업의 출발시점과 도착지점에 대해서 미리 정해져 있는 경로와 예약스케줄이 AGV에게 전달된다. 이를 위하여 모든 경로 각각에 대해서 예약스케줄이 AGV 통제컴퓨터에 기억되어 있어야 한다. 본 연구에서는 어떤 상황에서도 교착이 발생하지 않는 예약 스케줄 표를 사전에 작성하여 두고 이를 통제컴퓨터에 기억시킨 후 운행경로가 차량에게 전달될 때, 예약 스케줄을 읽어서 전달하는 방식을 제시하고자 한다.

본 연구의 목적은 하나의 경로에 대해서 상황에 따라서 변하지 않는 일정한 예약 스케줄을 제공하고자 하는 것이므로 어떤 우선순위테이블을 가정하여야 하느냐는 문제가 생긴다. 본 연구에서는 어떤 운행경로를 가진 차량들이 미리 예약을 한 상태에서도 교착이 발생하지 않을 운행경로에 따라 상황에 따라 변하지 않는 예약 스케줄을 작성하고자 한다. 이를 위하여 시뮬레이션을 통하여 오랜 시간 다양한 상황이 발생하도록 한 다음, 결과적으로 얻어지는 우선순위테이블을 이용하여 특정한 운행경로의 예약스케줄을 작성함으로써 모든 상황에서도 교착상태를 방지할 수 있는 예약 스케줄을 결정한다.

3.1 시뮬레이션을 이용한 예약스케줄 표 도출

그림 4는 예약 스케줄 표를 작성하기 위하여 시뮬레이션을 이용한 도출 절차를 나타낸 것이다. 예약 스케줄 도출 절차를 예를 들어 설명하면 다음과 같다.

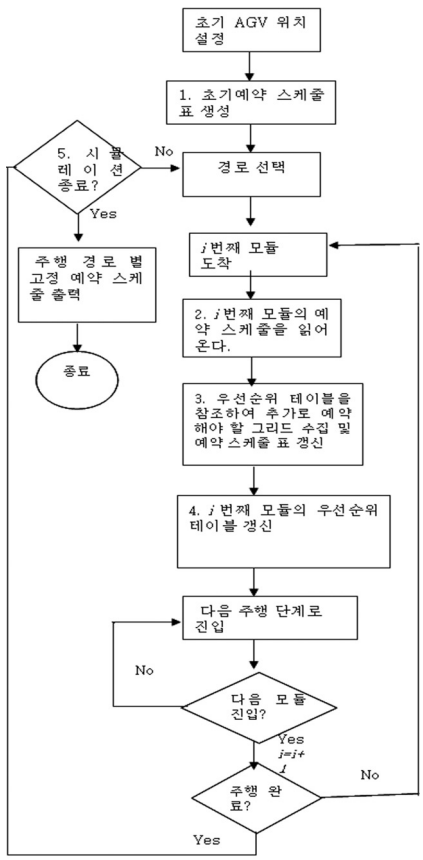

그림 4. 시뮬레이션을 이용한 예약 스케줄 표 작성절차

[단계 1] 초기예약 스케줄 표 생성.

모든 가능한 출발지와 도착지 쌍에 대해서 각 모듈별 예약 스케줄을 작성하여 예약 스케줄 표에 초기 값으로 넣어 둔다. 참고로 하나의 출발지와 도착지의 표현은 표 4와 같다. 운행 경로상에 있는 각 모듈 별로 주행상에 거치게 되는 예약, 해제, 예약예정 정보를 이용해서 표 2와 같은 정보를 생성하여 예약 스케줄표의 초기 값으로 한다.

표 4. AGV에 할당된 운반요구 및 주행경로 예시

출발지		도착지	
모듈번호	그리드 번호	모듈번호	그리드 번호
2	31,30,29	19	1058

주행모듈	
모듈 방문 순서	모듈 번호
1	2
2	3
3	10
4	17
5	18
6	19

[단계 2] 진입 예정 모듈(i번째 모듈)에 대해서 해당 경로의 예약 스케줄을 읽어온다.

진입예정 모듈이 3번 모듈이라고 가정하고 모듈 3에서의 주행 경로는 그림 5와 같다고 하자. 이때 해당 경로에 해당되는 읽어 온 예약 스케줄이 표 5와 같이 나타낼 수 있다.

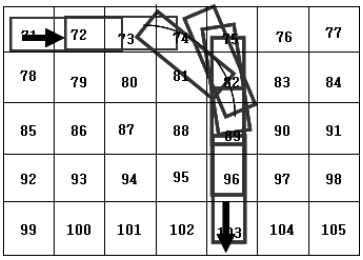

그림 5. 3번 모듈에서의 새 차량의 주행경로

표 5. 새롭게 진입하려는 차량의 예약 스케줄

모듈 번호	주행 단계	예약 스케줄		
		예약 그리드	예약 해제 그리드	예약 예정 그리드
3	1	71,72,73		74
3	2	72,73,74	71	81,82
3	3	82,81,73,74	72	75,89
3	4	89,82,75,74	73,81	
3	5	89,82,75	74	96
3	6	96,89,82	75	103
3	7	103,96,89	82	

[단계 3] 진입 예정 모듈 (i번째 모듈)의 최근 우선 순위 테이블을 읽어온다.

이때, 해당 모듈의 우선순위 테이블을 참조하여 추가로 예약되는 그리드를 수집하게 되는데 세부단계는 아래와 같다.

초기설정 j=1

단계 3-1 j단계의 예약예정 그리드와 j 이후 단계에서의 예약예정 그리드의 쌍을 선택한다.

단계 3-2 해당 쌍의 그리드의 우선순위 관계가 우선순위테이블의 그리드 우선순위관계가 위배되는지 확인한다.
위배되지 않으면 단계 3-3으로 간다. 위배되는 쌍이 있다면 해당 예약예정 그리드를 j단계의 예약 예정 그리드에 추가한다.

단계 3-3 예약 스케줄의 모든 예약예정 그리드의 우선순위 관계를 확인 하였으면 종료 하고 그렇지 않으면 $j=j+1$로 하고 단계 3-1로 간다.

표 2와 같이 경로의 예약 스케줄이 확정된 상황에서 새로운 경로의 예약 스케줄을 추가로 작성하는 경우에 대해서 예시 해 보겠다. 표 2는 다른 차량의 운행이 없다는 가정하에서의 예약 스케줄이다. 그러나 기존 경로가 먼저 계획되어 집행되고 있는 상황에서 새로운 차량의 주행경로 위한 예약 스케줄알고리즘에 의해서 작성하면 표 6과 같이 되는데 여기에는 교착방지를 위하여 그리드들이 추가로 예약되고 있음을 알 수 있다. 예를 들어, 표 6의 AGV의 예약 스케줄에서 예약 예정 그리드 중 74가 75보다 우선순위가 높게 되어있지만 참조한 우선순위테이블인 표 3에서 보면 이미 그리드 75가 74보다 우선순위가 높게 스케줄 되어있으므로 우선순위 규칙에 위배되는 상황이 발생하게 된다. 이러한 경우에는 그리드 75를 그리드 74와 동시에 예약을 함으로써 직전 우선순위 테이블상에 명시된 우선순위를 만족시킬 수 있다. 이렇게 교착을 방지하기 위하여 각 단계에서 동시에 예약되어야 그리드의 집합을 수집하게 된다. 동시에 예약되어야 할 그리드들을 추가하여 기존의 예약스케줄 표를 갱신한다. 참고로 표 7은 표 6의 예약스케줄을 고려한 후 갱신된 우선순위테이블 예를 보여주고 있다.

표 6. 교착을 방지하기 위해서 조정된 예약 스케줄

모듈 번호	주행 단계	예약 스케줄		
		예약 그리드	예약 해제 그리드	예약 예정 그리드
3	1	71,72,73		74,75
3	2	72,73,74,75	71	81,82
3	3	82,81,73,74,75	79,80	89
3	4	89,82,75,74	73,81	
3	5	89,82,75	74	96
3	6	96,89,82	75	103
3	7	103,96,89	82	

표 7. 조정된 예약 스케줄을 반영한 모듈 3의 우선순위 테이블

	73	74	75	76	77	78	79	80	81	82	88	89	95	96	102	103
73										1		1		1		1
74										1		1		1		1
75		1		1	1					1	1	1		1		1
76		1			1						1					
77																
78	1	1	1						1	1		1		1		1
79	1	1	1						1	1		1		1		1
80	1	1	1						1	1		1		1		1
81	1	1	1	1	1					1	1	1		1		1
82		1	1	1	1						1			1		1
88		1	1	1	1				1	1	1					
89														1		1
95		1	1	1	1				1	1	1					
96														1		1
102		1	1	1	1				1	1	1					
103														1		

[단계 4] 새롭게 작성된 예약스케줄 상에 있는 예약된 그리드와 예약할 그리드들의 관계를 이용하여 우선순위테이블을 갱신한다. 우선순위 테이블 갱신 세부절차는 다음과 같다.

초기설정 k=1

단계 4-1 조정된 예약 스케줄에서 k단계의 예약 그리드와 예약 예정 그리드 리스트를 선택한다.

단계 4-2 예약 그리드와 예약 예정 그리드 쌍을 차례대로 선택한다.
우선순위테이블의 X축에서 해당되는 예약 그리드 번호를 찾고, Y축에서 해당 되는 예약 예정 그리드를 찾아서 교차되는 칸에 1을 넣는다.
현재 단계의 예약 그리드와 예약 예정 그리드의 관계 값 입력이 다 되었으면 단계 4.3으로 간다. $k = k + 1$로 한다.

단계 4-3 더 이상 갱신 할 주행단계가 없다면 4.4로 가고 그렇지 않으면 단계 4.1로 간다.

단계 4-4 조정된 예약 스케줄에서 예약 그리드와 예약할 그리드들 사이에 transitive 관계가 있는 그리드 쌍을 찾는다.
우선순위 테이블에 해당 그리드 쌍의 위치를 찾아서 교차되는 칸에 1을 넣는다.
예약 그리드와 예약예정 그리드에 대한 transitive 과정이 끝났다면 종료한다.

즉, 예약 스케줄에서 먼저 예약된 그리드는 예약할 그리드보다 높은 우선순위를 둔다. Z1을 먼저 예약된 그리드 집합의 원소이고 Z2는 그 시점의 예약 되어야 할 그리드 집합의 원소로 가정하자. 이 경우 Z1이 Z2보다 우선순위가 높고, Z1 → Z2로 표현된다. 그리고 Z1 → Z2 이고 Z2 → Z3이라면 Z1 → Z3로 되는 transitive 특성을 만족하며 이 관계도 우선순위테이블에 추가되어야 한다. 새로운 예약 스케줄을 반영하여 3번 모듈의 우선순위 테이블은 표 6과 같이 갱신된다.

[단계 5] 더 이상의 추가로 예약되는 그리드가 생기지 않는다면 시뮬레이션을 종료하고 주행경로 별 고정예약 스케줄을 출력한다.

이와 같은 단계를 통하여 마지막으로 얻게 되는 예약 스케줄 표가 우리가 찾는 고정 예약 스케줄이 된다.

3.2 알고리즘 복잡도 분석

본 연구에서 제시한 알고리즘의 시간 복잡도를 결정하는 부분은 단계 3과 단계 4의 계산부분이다. 알고리즘 복잡도를 나타내기 위하여 사용된 기호는 다음과 같다.

n = 주행단계의 수
r = 주행단계별 예약 할 그리드의 개수
m = 한 모듈을 구성하는 그리드의 개수
c = 주행 단계별 예약된 그리드 개수

단계 3의 예약 스케줄 표 갱신은 3개의 세부 단계로 구성되는데 3-1단계의 반복탐색 내부에 단계 3-2와 3-3의 조건탐색이 있는 구조이다. 단계 3의 복잡도를 나타내어 보면 (r(2+m)-(m+1))(4r-2+mr-m+n(r-1)2(n)+(r-1)3(n))이 된다. 이 식의 최고차항을 나타내 보면 $r^4(2+m)(n)$이므로 단계 3의 계산시간의 복잡도는 $O(r^4mn)$이 된다. 단계 4의 우선순위 테이블 갱신단계에서는 크게 두 개의 독립 탐색 영역으로 이루어지는데 첫 번째 탐색범위는 4-1에서 4-3

단계까지 이고 나머지는 4-4 단계가 된다. 단계 4의 첫 번째 탐색 범위에서의 복잡도는 $2nc^2(2+m)+rmmc(m+5)+3rnc$ 이고, 두 번째 탐색의 범위의 복잡도는 $n(2r+mr-m-1)(r^2+r+rm-m-2)$이다. 알고리즘 수행시간에 문제가 되는 것은 데이터 개수가 많을 때 이므로 최고 차수를 가진 단계 3의 복잡도를 알고리즘의 계산시간의 복잡도로 나타낼 수 있다. 따라서 본 연구에서 제시한 알고리즘의 복잡도는 O(r4 mn)이다. 즉 $O(m^6)$을 나타낸다.

알고리즘의 복잡도 $O(r^4mn)$에 사용된 변수들의 관계를 살펴보면 $r < m$ 이고 $n < m$ 이다. 따라서 여러 대의 차량이 특정모듈에서 주행할 경우 예약할 그리드 개수와 주행단계의 수는 모듈 내 그리드의 개수 보다는 작거나 같게 되므로 복잡도는 $O(m^6)$이 된다.

4. 예약스케줄 방법에 대한 평가 및 민감도 분석

이 장에서는 앞장에서 도출한 예약 스케줄방법을 시뮬레이션을 통하여 성능을 평가해 보았다. 시뮬레이션을 통하여 출발지와 도착지로 표현되는 모든 운반요구에 대해서 예약 스케줄을 도출하였다. 이 과정에서 앞장에서 설명한 방식으로 하나의 모듈에 대해서 한 차량의 예약 스케줄을 수정하는데 소요되는 시간은 차량의 대수가 16대일 때 약 1.2초가 소요되었다.

모델링 대상 컨테이너 터미널은 하나의 선석을 가지고 있었고 3대의 안벽크레인(QC:QUAY CRANE)과 7개의 야드 블록으로 구성되며 그림 1과 같다. AGV주행레인의 방향은 미리 정해져 있으며 직진 주행 시 속도는 6m/s이고 회전주행 시 속도는 2m/s이다. 시뮬레이션 모델은 EM-PLANT 7.6버전을 사용하였으며 PENTIUM 4, 3.0 GHZ, 800MB PC에서 실험을 수행하였다.

Warm up period는 7일로 하였으며 10회씩 반복실험을 하였다. 차량의 주행 속도, 차량 간의 blocking 시간, 예약면적에 대한 실험을 수행하여 도출된 예약 스케줄의 성능을 검증하였다. AGV의 주행 효율을 알아보고자, 차량의 대수를 늘려가면서 주행 속도와 blocking시간을 측정해 보았다. blocking 시간은 차량주행 시 다른 차량에 의해 지체되는 시간을 의미한다. 차량의 대수는 8대부터 2대씩 증가하게 하여 최대 16대까지 하였다. blocking시간은 각 경로의 blocking되는 시간의 합으로 계산하였다. AGV의 라우팅은 선석과 야드 TP(transfer point)사이를 주행하는 것으로 묘사한다. 그림 6은 예약 스케줄 표 방식으로 예약을 하는 경우, 전체 점유하게 되는 그리드 개수에 비해서 교착을 방지하기 위하여 추가로 예약하게 되는 그리드 개수의 비율을 나타낸 것이다. 5% 이하의 그리드를 추가로 예약하여 교착을 방지하게 되었다는 의미이다.

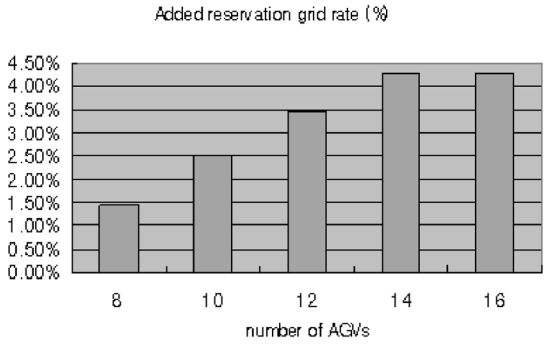

그림 6. 추가로 예약되는 그리드 비율

예약 스케줄 표 방식에서의 차량의 평균주행속도는 아래 그림 7과 같은데, 차량의 수가 많아질수록 차량의 속도가 떨어짐을 알 수 있다. 그림 8은 하나의 운행경로에 대해서 하나의 차량 당 평균 blocking 시간을 보여준다.

그림 9는 예약 스케줄 표 에 의한 주행 시 요구되는 예약 면적을 나타낸 것이다. 단위 시간당 요구되는 예약면적 계산은 차량들이 주행 중에 수집한(예약 면적*예약시간)의 합을 차량들의 주행시간의 합으로 나누었다.

Kim et al.[4]에서 제시된 방법을 이용하여 그때 그때의 예약된 상황을 감안하여 새로운 예약스케줄을 작성할 수도 있겠으나 실시간으로 적성되어야 한다는 점을 고려할 때, 계산 시간상의 문제가 있다는 것을 다음과 같이 확인

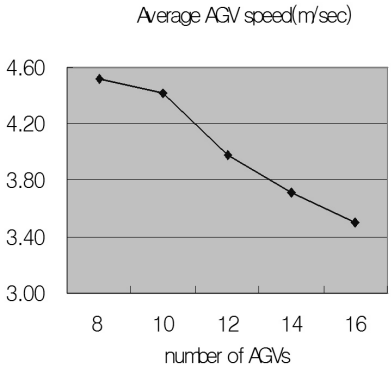

그림 7. 평균 주행속도

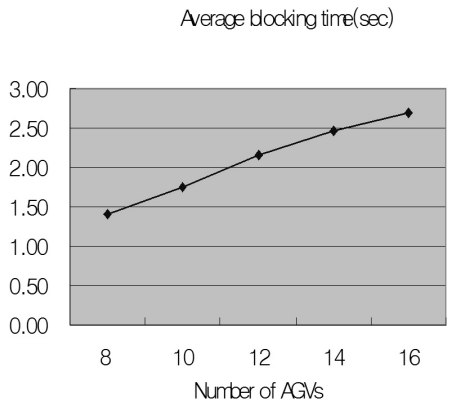

그림 8. 평균 blocking 시간

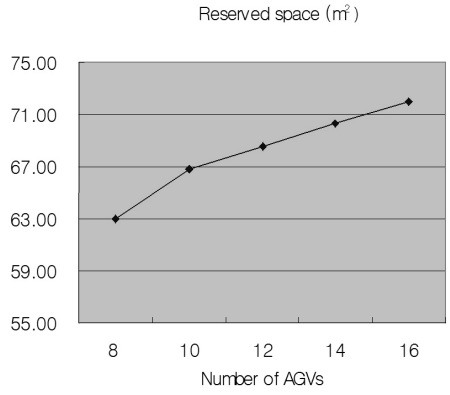

그림 9. 주행 시 필요한 평균 예약면적

할 수 있었다. 실험에 의하면 한번 예약스케줄을 계산하는데 소요되는 시간은 1.2초가 소요되었다. 차량이 모듈에 진입할 때마다 예약스케줄을 작성하는 경우, 여러 대의 차량이 하나의 차량 통제 컴퓨터에게 계산을 요청하게 되므로 계산시간이 길 경우 차량이 대기하는 시간이 발생될 수 있다.

계산을 요청하는 차량과 계산을 수행하는 컴퓨터와의 관계를 대기행렬 모형(M/M/1)으로 나타낼 수 있다. 실험에서 가장 많은 차량대수를 투입한 16대를 가지고 대기시간을 계산해 보자. 한 라우트를 주행하는 시간은 평균 93초가 걸렸고, 3개의 모듈 진입마다 알고리즘계산을 요청하므로 차량 당 하나의 경로가 할당되는 경우 전체 차량의 총 계산 요청 횟수는 평균 16*3=48회가 된다. 이는 약 2초마다 1번의 요청이 발생한다는 것을 의미하므로 평균 도착 률 λ= 0.5가 된다. 16대 차량이 동시에 투입된 경우 하나의 예약 스케줄에 대한 평균계산시간이 1.2초 이므로 μ=1/1.2 = 0.83이다. M/M/1 대기행렬 모형에서 평균대기 시간 $W_q = \frac{\rho}{\mu - \lambda}$이고 $\rho = \frac{\lambda}{\mu}$이다. 이 식을 이용하여 대기행렬 모형에서 알고리즘 계산을 위한 평균 대기시간을 구해보면 W_q=1.825초가 된다. 계산시간으로 1.2초가 걸리므로 한 모듈 진입 시 차량 당 평균 대기시간과 계산시간의 합은 3.025(=1.825+1.2)초가 된다.

표 8은 동적 예약 스케줄링 방식과 본 논문에서 제시한 고정예약 스케줄링 방식을 비교한 것을 나타낸다.

표 8. 주행 시 주행 당 평균 소요시간의 비교

	동적 예약 스케줄링 방식	고정 예약 스케줄링 방식
1회주행 평균거리(m)	344m	344m
평균 주행속도 (m/sec)	3.7m/sec	3.5m/sec
1회 주행당 평균 주행시간(sec)	93초	98.2초
모듈 당 예약 스케줄링 계산시간과 대기시간 합(sec)	3.025초	0초
1회 주행당 평균 계산시간과 대기시간의 합(sec)	9.075초	0초
1회 주행당 평균 소요시간(sec)	102.75초	98.2초

표 8에서 보듯이 1회 주행 당 순수 주행시간은 동적 예약 스케줄링 방식이 유리하지만, 동적 예약 스케줄링 방식의 경우 모듈에 진입할 때 마다 예약 스케줄링을 위한 계산시간이 소요되므로 1회 주행 당 평균 소요 시간을 순수 주행시간과 예약 스케줄링 계산시간의 합으로 나타내 보면 고정예약 스케줄링 방식이 더 나음을 알 수 있다. 뿐만 아니라 현재 실험은 차량 16대를 대상으로 한 것이지만 통상 자동화 터미널에서는 차량의 대수가 선석당 16대 이상이 되고 3선석을 가진 터미널을 생각할 때, 차량의 대수는 50대 가까이 되리라 예상되는 바, 이 경우 동적 예약방식의 계산시간은 기하급수적으로 증가할 것으로 예상된다.

5. 결 론

자동화 컨테이너 터미널의 효율적인 운영을 위해서는 QC의 지연 작업시간 최소화가 요구된다. 따라서 QC와 장치장을 오가는 운송수단인 AGV의 효율적인 주행이 필

요하다. 그리고 AGV의 효율적인 운영을 위해서는 교착상태를 방지하기 위한 운영논리의 개발이 필수적이다. 본 연구는 컨테이너 터미널에서 사용될 수 있는 교착방지를 위한 알고리즘 개발을 목표로 하였다.

기존의 교착방지 알고리즘은 차량이 교착구간에 도착할 때마다 그 때의 상황을 고려하여 교착방지를 위한 예약스케줄을 작성하는 방식이었으나 이는 지나치게 많은 계산시간을 소요한다는 단점이 있었다. 이를 극복하기 위하여 본 연구에서는 하나의 경로에 대해서 예약스케줄을 작성하여 두고 반복적으로 사용하는 예약 스케줄 표 방식을 제안하였다. 실험 결과 예약 스케줄 표 방식은 그때그때의 주행경로에 대한 예약스케줄링을 위하여 계산시간이 필요 없기 때문에 AGV의 실시간 운영을 위하여 유용한 방식임을 알 수 있었다.

본 연구는 선석이 하나인 단순한 경우에 대해서 예약스케줄 방식을 제시하였다. 그러나 다양한 터미널의 배치형태에 따라서 경로결정 방식도 다양해 질 것이고 이에 따라 예약방식도 다양해 질 필요가 있을 것이다. 본 연구의 교착방지를 위한 알고리즘을 바탕으로 운행 효율이 극대화 될 수 있는 주행로 설계와 야드 배치에 관한 후속 연구가 필요하다고 생각한다.

감사의 글

"본 논문은 2008년 정부(교육과학기술부)로부터 지원받아 수행된 연구임"(지역거점연구단육성사업/차세대물류IT기술연구사업단).

참 고 문 헌

1. 이재용, 서윤호 (2008), "자재 취급 시스템을 위한 다중 에이전트 기반의 교착상태에 자유로운 AGV 시뮬레이터 개발", 한국시뮬레이션학회 논문지, 제17권, 제2호, pp. 91-103.
2. Evers, J. M. and Koppers, S. A. (1996), "Automated guided vehicle traffic control at a container terminal", Transportation Research Part A, Vol. 30, No. 1, pp. 21-34.
3. Grunow, M., Günther, H.O. and Lehmann, M. (2004), "Dispatching multi-load AGVs in highly automated seaport container terminals", OR Spectrum, Vol. 26, No. 2, pp. 211-236.
4. Kim, K. H., Jeon, S. M. and Ryu, K. R (2006), "Deadlock Prevention for automated guided vehicles in automated container terminal", OR Spectrum, Vol. 28, No. 4, pp. 659-679.
5. Lee, C. C. and Lin, J. T. (1995), "Deadlock prediction and avoidance based on Petri nets for zone control Automated Guided Vehicle Systems", International journal of production research, Vol. 33, No. 12, pp. 3239-3265.
6. Matthias, L., Grunow, M. and Günther, H.O. (2006), "Deadlock handling for real-time control of AGVs at automated seaport container terminals", ORSpectrum, Vol. 28, No. 4, pp. 631-659.
7. Rajeeva, L. M., Wee, H. G., Ng, W. C., Teo, C. P. and Yang, N. S. (2003), "Cyclic dead-lock prediction and avoidance for zone-control AGV system", International Journal of Production Economics, Vol. 83, pp. 309-324.
8. Reveliotis, S. A. (2000), "Conflict resolution in AGV system", IIE Transactions Vol. 32, pp. 647-659.
9. Samia, M. and Castagna, P.(2005), "A performance-based structural policy for conflict free routing of bi-directional automated guided vehicles" Computers in industry, Vol. 56, No. 7, pp. 791-733.
10. Yeh, M. S. and Yeh, W. C. (1998), "Deadlock prediction and avoidance for zone-control AGVs", International Journal of Production Research, Vol. 36, No. 10, pp. 2879-2889.
11. Duinkerken, M. B., Evers, J. M. and Ottjes, J. A. (1999), "TRACES : Traffic control engineering system", Proceedings of the 31st Summer Computer Simulation Conference, pp. 461-465.
12. Zeng, J. Y. and Hsu, W. J. (2003), "Conflict-free routing of AGV on the mesh topology based on a discrete-time model," Robotics and automation proceeding ICRA 03 IEEE International conference, Vol. 3, pp. 3510-351.

전 수 민 (1006sumin@pusan.ac.kr)

2002 한국해양대학교 물류시스템공학과 학사
2004 부산대학교 산업공학과 석사
2004～현재 부산대학교 산업공학과 박사과정

관심분야 : 물류관리, 시뮬레이션

김 갑 환 (kapkim@pusan.ac.kr)

1977 서울대학교 산업공학과 학사
1979 한국과학기술원 산업공학과 석사
1987 한국과학기술원 산업공학과 박사
1984～현재 부산대학교 산업공학과 교수

관심분야 : 물류시스템, 생산관리