Multi-Cycle 작업을 위한 Y/T 최단거리 및 예상 이동거리 계산

박태진* · 김한수* · 김치용**

*부경대학교· **동의대학교

Computation of the Shortest distance and Forecasting movement distance for Y/T Multi-Cycle System

Tae-jin Park* · Han-soo Kim* · Kim Cheeyong**

*Pukyong National University · **Dong-Eui University

E-mail : csptj2@naver.com, hskim3363@hanmail.net, kimchee@deu.ac.kr

요 약

본 연구 논문에서는 RTLS(Real Time Location System)를 기반으로 하는 Y/T(Yard Tractor) Multi-Cycle System (RYMS)의 핵심기능으로써 컨테이너터미널 양・적하작업 등 저효율적인 하역 분야 개선에 목적을 둔다. 이와 같은 목적을 달성하는데 있어서는 최적 경로의 탐색과 최단 이동거리를 찾아낼 수 있도록 최적 알고리즘을 적용, Y/T를 선택하고 확정하는 기법이 요구된다. 본 연구논문에서 제안된 방식의 실험결과를 살펴보면, 평균치대비 절감 운전거리율은 12%가 되었고 최대치대비 절감 운전거리율은 23%가 되었음을 확인할 수 있었다.

.

ABSTRACT

In this paper, we introduce a Y/T(Yard Tractor) Multi-Cycle System on the basis of RTLS that improves a low efficient loading and unloading. In the proposed approach, we apply the best suited algorithm looking for seeking of the optimum path and the shortest movement distance. In this paper, Our experiment results show that rate of a driving distance is reduced more than 12% compared to the average value, and that is reduced more than 23% compared to the maximum value.

키워드

RYMS, 최단거리 계산, 예상이동거리 계산, 최적경로 탐색, Dijkstra 알고리즘

Ⅰ. 서 론

최근, 동북아 인접 경쟁국가들의 전략적인 항만 개발로 인한 국내 항만 경쟁력의 정체, 초대형 컨테이너 선박 기항의 보편화로 인한 컨테이너터미널의 생산성 향상경쟁이 치열하게 전개되고 있다. 따라서 첨단 u-IT기술을 이용한 해운 항만 물류 분야의 경쟁력 강화라는 정부의 정책목표에 대응하고, 기존의 RTLS[1]를 기반으로 하는 컨테이너터미널 양・적하 작업 등 저효율적인 하역 분야 개선에 목적을 둔다.

컨테이너터미널의 생산성은 취급하는 작업시간과 관련되어 있다는 점에서 작업시간의 최소화할 수 있는 기술과 운영방법이 중요하며, 작업소요 시간은 곧 비용이라는 점에서 운영 및 작업 방식의 개선과 자동화 등의 연구가 필요하다[2].

컨테이너터미널의 급격한 물동량 증가를 고려했을 때, 본 연구논문에서 제안하는 방법은 컨테이너터미널의 효율적 운영과 개선에 큰 역할을 할 것이다. 본 연구논문에서는 하역분야 개선에 반드시 필요한 최적경로의 탐색과 Y/T 작업 최단거리에 대한 계산을 수행함으로써 효율적 이동거리를 찾아내는데 있다.

II. 최단경로 설정을 위한 알고리즘과 하역시스템에서의 운영절차와 구성

2.1 최단경로 Dijkstra 알고리즘

그림 1(a)와 같은 방향그래프(Directed Graph)의 간선이 양수의 가중치를 가질 때 임의의 출발 정점에서 도착 정점까지의 경로 중 경로의 길이가 최소인 경로를 최단경로라고 정의한다. 최단경로 설정을 위한 단계별 알고리즘은 표 1과 같다.

<Dijkstra's Algorithm>

[*Step 1*] 인접행렬 상태로 각 간선의 가중치가 존재하도록 한다. 정점 i에서 i까지의 가중치는 0이고 정점 i에서 j까지의 간선 E(i, j)가 존재치 않으면 가중치는 무한대(∞)가 된다.

[*Step 2*] 집합 S와 T를 정하는데 S는 출발 정점을 초기 값으로 하고 집합 T는 출발 정점을 제외한 모든 정점을 포함하도록 초기화한다.

[*Step 3*] 출발 정점을 제외한 모든 정점으로부터의 거리의 초기값(dist[i], 2<=i<=n)을 [1]의 인접행렬에서 취한다.

[*Step 4*] 집합 T의 원소 중 출발점으로부터의 거리가 최소인 정점 v를 택하여 T에서 제거하고 집합 S에 추가한다.

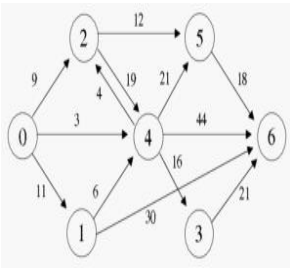
[*Step 5*] T집합 내의 모든 정점 w에 대해 출발점으로부터의 거리(dist[w])와 간선 E(v, w)의 길이에 v정점의 거리(dist[v])를 합한 값 중 적은 것을 선택하여 정점 w의 거리(dist[w])로 한다.

[*Step 6*] Step 4와 Step 5의 과정을 T가 공집합이 될 때까지 반복한다.

표 1. 최단경로 설정을 위한 Dijkstra's algorithm

```
shortest(int adj_matrix[n][n]) {
 T={2, 3, ..., n};
    S={1};
 for(i=2; i<=n; i++)
    dist[i]=adj_matrix[1][i];  //초기 경로 길이
    for(i=2; i<=n; i++) {
       v=T내의 정점 중 dist[i]가 최소인 정점;
       T에서 v를 삭제;
       S에서 v를 삽입;
       while(T내의 모든 정점 w에 대해)
             dist[w]=MIN(dist[w],
                   dist[v]+adj_matrix[v][w])
    } return(dist);
 }
```

adj_matrix[ ][ ]는 그림 1(b)와 같은 간선의 인접행렬을 나타낸다.



	0	1	2	3	4	5	6
0	0	11	9	∞	3	∞	∞
1	∞	0	∞	∞	6	∞	30
2	∞	∞	0	∞	19	12	∞
3	∞	∞	∞	0	∞	∞	21
4	∞	∞	4	16	0	21	44
5	∞	∞	∞	∞	∞	0	18
6	∞	∞	∞	∞	∞	∞	0

(a) 방향그래프 (b) 인접행렬과 가중치

그림 1. 최단경로 설정과 인접행렬

2.2 컨테이너터미널 하역시스템과 Y/T 배정 및 운영 절차

(1) 하역시스템 구성요소 및 인터페이스 특성

컨테이너터미널 내 하역시스템은 선박, 본선하역장비(C/C; Container Crane), 내부차량(Y/T; Yard Tractor), 야드 하역장비(T/C; Transfer Crane, R/S; Reach Stacker), 야드, GATE, 외부차량으로 구성 되어 있으며, 하역 시스템별 구성 요소 및 인터페이스 특성은 그림 2와 같다. 화살표는 구성 요소별 인터페이스에 대한 요구사항을 나타내고 있으며, 요구사항에 대한 최적의 계획, 실행, 모니터링, 자원배치 등을 제공 했을 때에는 터미널의 운영 효율성이 극대화 된다.

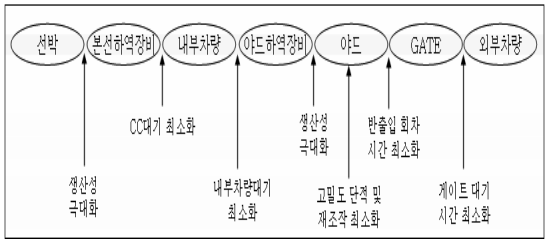


그림 2. 하역시스템 구성요소 및 인터페이스 특성

(2) Y/T(Yard Tractor) 배정 및 운영 절차

Y/T는 양.적하 컨테이너 화물을 본선과 야드간에 운반하는데 이용되는 운송 장비로써 배정 및 운영절차는 다음과 같다. 첫째, 선박이 입항하게 되면 선박별 본선하역장비(C/C) 투입현황을 확인하고 C/C별 Y/T를 배정(C/C당 Y/T: 4대), 배정된 Y/T는 해당 C/C로 이동한다. 둘째, 특정 C/C별로 배정된 Y/T에 대하여 Under Clerk(U/C)은 배정된 Y/T를 확인하고 입력관리를 수행한다.

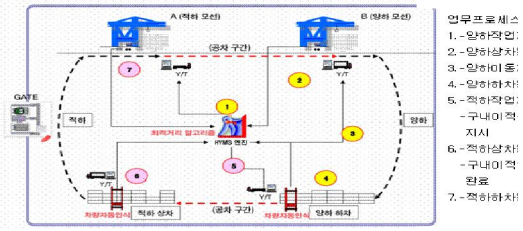


그림 3. 하역시스템에서의 업무프로세스

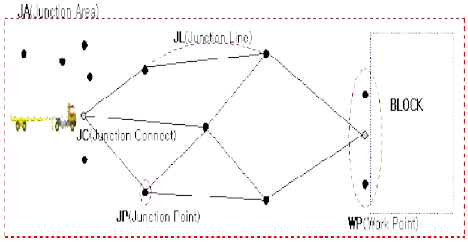


그림 4. 접점경로의 구성요소

위 그림 3과 같은 업무프로세스에서의 Y/T 운영형태는 Y/T 풀링(Pooling) 시스템에 의해서 그 목적을 달성할 수 있으며, Y/T 풀링 시스템은 Y/T 배정 및 배차, 교대, 그리고 선택, 작업지시 등을 Y/T 풀링 엔진에 의해서 수행되도록 한다.

III. 시스템 설계와 구현

본 연구논문에서는 데이터베이스에 구축된 야드 맵(Yard Map) 상에서 Y/T 작업 최단거리 자동 탐색 및 예상 이동거리를 계산한다. 이를 위해서는 터미널 환경에 맞도록 알고리즘이 적용되어야 하며, Y/T 효율적 이동선 재현 그리고 최단거리 재설정 검증을 위한 시뮬레이션 기능을 포함한다. 여기서 야드 맵에 입력된 JP의 좌표위치값 자동 조회와 터미널 환경에 따른 최단거리 설정을 위해서 Dijkstra 알고리즘[3][4]을 적용하였다.

3.1 야드맵에서의 이동경로 설정과 위치정의

야드 맵에서의 이동경로 설정과 위치 정의를 위해서는 야드 맵에서의 단위 접점을 나타내는 JP(Junction Point)와 현재 차량의 위치에 가장 가까이 있는 JP할당에 필요한 JA(Junction Area), 그리고 JA에 소속된 JP임을 나타내는 JC(Junction Connect), 작업지점을 나타내는 WP(Work Point), 접선, 두 지점 간의 거리를 나타내는 JL(Junction Line), YT가 임의의 JP에서 목적지 JP까지의 경로를 복수적으로 요청할 수 있는 HOOP 등의 연산을 필요로 하며, JP, JA, JC, JL, HOOP에 대한 합연산을 수행한다.

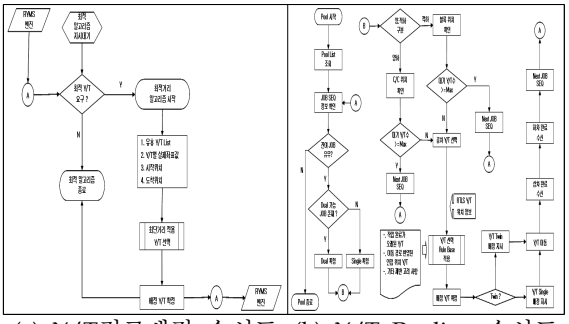
(1) 접점 경로(Junction Path)의 구성

JP할당에 필요한 접점영역은 그림 4와 같이 나타낼 수 있으며, 터미널 환경에 따른 최단거리 알고리즘이 적용되도록 하였다.

특히, 아래 그림 4에서처럼 접선, 두 지점간의 거리를 정의하기 위한 JL은 JP와 다음 JP의 위치를 알 수 있도록 이동거리를 계산하였으며, 이는 JP에서 다음 JP까지의 속성을 의미하는 것으로 접점의 차를 나타낸다. 여기서 작업지점 WP는 JP라는 개념의 연장선에서 표현되며 인접 블록과 연결되어 있는 것으로 한다.

3.2 Y/T 최적거리 계산을 위한 알고리즘

Y/T 경로배정을 위해서는 그림 5(a)에서처럼 알고리즘의 수행과 동시에 유휴 Y/T를 찾아내어 목록을 작성하고 Y/T별 실제좌표값을 탐색한다. 이와 같이 탐색되고 계산된 최단경로는 야드 맵에서 적용이 가능하여 최종 목적지로 이동할 수 있는 Y/T를 선택하고 확정한다. Y/T 풀링 시스템은 그림 5(b)에서 보인 것처럼 Y/T 배정 및 배차, 교대, 그리고 선택, 작업지시 등을 Y/T 풀링 엔진에 의해서 수행될 수 있도록 구현된다.



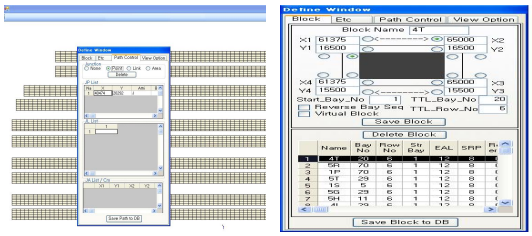
(a) Y/T경로배정 순서도 (b) Y/T Pooling 순서도

그림 5. Y/T 최적거리 계산 알고리즘

IV. Y/T를 위한 모니터링 및 시뮬레이션

RTLS 기반의 위치추적 기술을 이용해서 컨테이너의 위치를 파악함과 동시에 생성된 접점 경로(Junction path)정보를 바탕으로 최단거리알고리즘 연산을 수행하고 야드 맵 상의 블록표시와 등록된 JP 표시를 통해서 시뮬레이션 기능을 제공할 수 있다. 여기서 Y/T에 대한 최단, 최장 운행거리 조회가 가능하다. 실시간 위치 추적기술은 기존 RTLS기반의 거리측정방식으로써 RSSI 기법을 적용하되 삼각측량법에 의존하는 것으로써 동일 직선상에 있지 않은 3지점으로부터의 거리를 계산하여 위치를 측정하는 방법이다. 결과적으로는 Y/T실시간 위치정보와 야드 컨테이너 정보 등을 활용해서 컨테이너터미널 내 하역장치장의 상태를 모니터링 할 수 있게 된다. 아래 그림 6(a)는 최단거리알고리즘이 적용되는 야드 맵과 경로제어 상태를 나타내고 있으며 Y/T 이동

경로 정보를 데이터베이스에 저장할 수 있도록 한다.



(a) 컨테이너 위치와 경로제어 (b) 블록 위치 설정·

그림 6. Y/T 경로제어 및 블록 위치 설정

위 그림 6(b)는 하역장치장에서 배치된 컨테이너의 위치를 파악할 수 있도록 하고 블록의 위치를 데이터베이스에 저장할 수 있도록 한다. 이로써 Y/T가 이동할 수 있는 최단경로를 탐색하고 예상 이동거리를 자동 계산할 수 있게 된다.

아래 그림 7에서와 같이 이미 데이터베이스에 저장되고 구축된 야드 맵을 출력하고, 출력된 좌표의 위치를 탐색할 수 있어서, 이와 같이 탐색된 야드 맵에서 JP간 최단거리 자동 설계알고리즘이 적용되도록 한다. 그런 다음 야드 내 특정지점을 선택한 후, 목적지까지의 시뮬레이션을 수행하게 된다. 따라서 Y/T 위치 이력조회가 가능하고 Y/T 최단운행과 최장 운행거리, 최적 Y/T 투입수를 확인할 수 있다. 결과적으로는 조건을 만족하는 컨테이너 개수 및 현재 야드 내 장치된 컨테이너 개수도 출력되며, 그리드에는 조회된 컨테이너 상세정보가 나타남을 알 수 있다.

그림 7. 최단거리 탐색 및 시뮬레이션

위의 그림 6,7에서와 같은 Y/T 모니터링 및 시뮬레이션 결과를 통해서 하역장치장에서의 작업 현황을 파악하고 작업진행을 원활히 진행될 수 있음을 알 수 있다. 본 연구논문에서 제안된 시뮬레이션을 통해서 아래 표 2와 같은 결과를 얻을 수 있었다. 절감운전거리 비율을 살펴보면, 평균치대비 12%가 되었고 최대치대비 23%가 되었음을 확인할 수 있었다.

표 2. 최단.최장총거리에 대한 절감운전비율 분석

순번	선택된 Y/T		최단		최장		Diff
			No	Distance (m)	No	Distance (m)	
1	319		319	30898	319	30898	0
2	415		415	19692	415	19692	0
3	415		415	12508	415	12508	0
4	416		416	45343	416	45343	0
.....	.....		.....	.....	.....	.....	.....
126	416		416	44204	416	44204	0
127	327		327	45328	327	45328	0
128	319		319	34972	319	40430	5458
129	413		413	45343	413	47669	2326
.....	.....		.....	.....	.....	.....	.....
201	413		413	31535	413	31535	0
202	378		378	16936	378	53315	36379
203	321		321	27125	321	84587	57462
204	321		321	61536	321	134000	72464
.....	.....		.....	.....	.....	.....	.....
260	364		364	10777	364	104403	93626
261	415		415	10935	415	111478	100543
262	379		379	21523	379	130292	108769
263	415		415	21838	415	149809	127971
최단총거리(A)			**10,272,384**		**최장총거리(B)**		**12,664,669**
B-A (C)			**2,392,285**				
평균거리: D=(B-A)/2+A			**11,468,526**				
절감운전거리율 (평균치대비) (D-A) / A			**12%**		**절감운전거리율 (최대치대비) (B-A) / A**		**23%**

Ⅴ. 결 론

본 연구 논문에서는 최단거리 및 예상거리 계산을 수행할 수 있는 알고리즘이 포함되며, RYMS 엔진이 사용된다. Y/T(Yard Tractor) 작업 시, 최단경로의 탐색과 예상 이동거리를 계산함으로써 하역장치장에서의 작업 내용을 원활히 할 수 있으며, Y/T 작업에 대한 업무 효율성을 높일 수 있게 된다.

본 연구논문에서 제안된 방식의 실험결과를 살펴보면, 평균치대비 절감 운전거리율은 12%가 되었고 최대치대비 절감 운전거리율은 23%가 되었음을 확인할 수 있었다. 결과적으로는 Y/T 작업에 대한 최단경로 탐색과 예상 이동거리의 계산으로 컨테이너터미널에서의 생산성을 최대화 할 수 있음을 알 수 있다.

참고문헌

[1] 권순량, 정광주, 박상훈, 김정훈, “유비쿼터스 항만 운영 효율화를 위한 RTLS 기술 적용”, 한국정보과학회 논문지, VOL. 13, NO. 06

[2] 홍동의, 정태충, "자동화항만의 야드 운영시스템 레이아웃 설계", 한국정보처리학회 논문지, VOL. 10-D, NO. 01, PP. 0101-0108, 2003. 02.

[3] B.-Shearing, "Tribute to Edsger W. Dijkstra," *Annals of the History of Computing*, VOL. 25, NO. 01, PP. 0067-0069, Jan. 2003.

[4] P.P.-Chen, "From goto-less to structured programming: the legacy of Edsger w.Dijkstra," *IEEE Software*, VOL. 19, NO. 05