

교차로 구간 회전 및 감속을 고려한 트랜스포터 최소 공주행 운영계획

문중헌¹·유원선^{2,†}·조두연³
충남대학교 선박해양공학과 대학원¹
충남대학교 선박해양공학과^{2,†}
목포대학교 조선해양공학과³

An Operation Scheduling of Transporters Considering Turns and Passing Delay at the Intersection Roads on the Shipyard

Jong-Heon Moon¹·Won-Sun Ruy^{2,†}·Doo-Yeoun Cho³
Graduate School, Dept. of Naval Architecture & Ocean engineering, Chungnam National University¹
Dept. of Naval Architecture & Ocean Engineering, Chungnam National University^{2,†}
Dept. of Naval Architecture & Ocean Engineering, Mokpo National University³

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

The operation planning of transports used to move blocks is the one of key factors. Furthermore, reducing the running time through the effective plan contributes to pulling forward the whole logistic process of the shipyard and substantially saving the fuel consumption of itself as well. The past researches of the transporter focused on finding only the shortest distances, so called, Manhattan distance. However, these searching approaches cannot help having the significant difference in the real operational time and distance with the minimum cost approach which considers the speed retardation for turns or safety at the intersection. This study suggests the noble transporter's operational model which could take account of the consuming operational time around the crossroads on the shipyard. Concretely, the proposed method guarantees the minimization of transporters' turns and passage number which are huge burdensome to the operation time and the whole planning of transports with the given period. Resultantly, this paper is willing to explain the appropriateness of our approach, compared with the previous ones.

Keywords : Dijkstra algorithm(다익스트라 알고리즘), Transporter(트랜스포터), Block transportation(블록 운송), Shortest path(최단경로), Simulated annealing(SA, 시뮬레이티드 어닐링), Intersection(교차로)

1. 서론

조선소에서 건조되는 선박 생산 과정은 선박 건조 기본 단위인 블록을 강판에서 절단, 성형, 소조립, 중조립, 대조립 등의 순차적인 단계를 거쳐 제작하며 그 후 선행 의장, 도장, PE를 거쳐 도크에서 탑재될 때까지 블록의 이동과 적치의 반복이다. 이 과정 중에 생산 공정 및 적치에 필요한 블록운반과정을 담당하는 것이 트랜스포터(transporter)이다. 공정 간의 블록 이동이 비효율적으로 계획된다면 전체적인 물류비용이 증가하고, 증가한 물류비용은 전반적인 생산 비용을 증가시킨다. 또한 적치장의 운용

효율을 떨어뜨리게 되고, 효율이 떨어진 만큼 물류 흐름이 원활히 이루어지지 않아 전체적인 생산 공정의 지연을 초래하여 막대한 손실을 야기할 수 있기 때문에 트랜스포터의 물류 운영 계획은 대단히 중요하다. 이에 관한 기존 연구를 정리해 보면 다음과 같다.

2. 관련 연구 현황

Koo, et al. (2004)은 화물 컨테이너 운반시스템에서 최소의 차량으로 일정한 시간 내의 주어진 컨테이너를 운반하는 차량 경로계획을 설정하는 절차를 제시하였다. 제시된 절차는 우선 공차

이동 시간을 최소로 하는 최적화 모형을 이용하여 차량의 소요 대수 하한 값을 결정하고, Tabu Search 기반의 차량 경로 계획을 수립하였다.

Jo, et al. (2005)은 트랜스포터의 종류별 블록 운송 계획을 세우기 위해 비슷한 중량의 블록들을 그룹화하고 최소의 트랜스포터 수를 구하기 위해 한 종류의 트랜스포터를 이용하도록 가정하여 Maxwell and Muckstadt (1982)가 제시한 최적화모형을 통해 휴리스틱 시간(공차 운행 시간, 지연 시간 등)을 최소화하는 것을 목표로 트랜스포터의 경로 및 일정 계획을 제시하였다.

Yu (2005)는 트랜스포터의 블록 운반 시 운반 거리 및 경로에 따른 운반 시간이 최소화되는 최적 경로를 Backtracking 알고리즘을 이용하여 결정하였다. 하지만 블록의 최적 운반 순서 결정은 다루지 않았다.

Yim, et al. (2008)과 Roh and Cha (2011)은 개미 알고리즘과 유전자 알고리즘을 이용하여 트랜스포터의 공주행 거리를 최소화하는 블록 운반 계획 문제의 최적 해를 구하였다. 그러나 운송 경로 손상 반영이 어렵고, 트랜스포터 교차 주행 여부를 고려하지 않았다.

Cha, et al. (2012)는 트랜스포터의 교차 주행 여부를 고려하였고 A* 알고리즘을 이용하여 계산 시간을 단축시키는 알고리즘을 구현하였다. 그러나 운송 경로 손상을 고려하지 않았다.

Heo, et al. (2013)은 트랜스포터의 최적 블록 운송 계획을 위해 Yim, et al. (2008)과 Cha, et al. (2012)에서 제시한 최적화 모형을 이용하여 트랜스포터가 계획된 운행을 하는 중 돌발 상황(트랜스포터의 고장, 사고, 경로 중 일부 PE장 활용)에 대처할 수 있는 방안을 제시하고 있다. 돌발 상황 발생 시 해당 도로를 제외한 다른 최단 경로를 A* 알고리즘을 통해 재계산하여 새로운 경로를 재탐색하도록 하는 연구를 수행하였다.

기존의 연구(Yim, et al., 2008; Cha, et al., 2012; Heo, et al., 2013)에서 경로를 탐색하는 데 있어 교차로 구간에서의 회전 및 통과 지연을 고려하지 않고 경로를 탐색해왔다. 실제 현장에서의 트랜스포터 운영을 살펴보면 교차로를 통과할 때 교차로의 좌, 우 도로에서의 시야 확보가 쉽지 않기 때문에 안전을 위해 트랜스포터는 감속하게 된다. 또한 좌회전과 우회전 할 경우에 따른 트랜스포터의 속도 변화를 경로 탐색에 반영할 경우 이를 반영하지 않았을 때와는 다른 결과를 줄 것이라는 가정으로 본 연구는 시작되었다. 본 연구에서는 경로를 탐색하는 과정에서 트랜스포터가 교차로 구간을 통과하는 상황(좌/우회전, 직진)을 고려하여 경로가 탐색 되도록 하려한다. 최단 경로 탐색에 사용된 알고리즘으로는 Dijkstra 알고리즘으로 하나의 출발지와 출발지를 제외한 다른 모든 절점 간의 최단 경로를 구하는 일대다 탐색 방법의 알고리즘이다(Dijkstra, 1959). 조선 야드에서 운영하는 전체 트랜스포터의 운영계획 차원에서 블록 운송 계획을 수립하기 위해 미리 모든 지번 간의 최단 경로가 필요하게 되는데 Moon, et al. (2016)에서 모든 지번 간의 최단 경로를 탐색하는데 있어 Dijkstra 알고리즘이 가장 적합한 알고리즘인 것을 확인하였다. 이를 바탕으로 트랜스포터의 운영계획에 있어서 최소 공주행 시간을 가질 수 있도록 주어진 트랜스포터들에 블록 할당

및 운반 순서를 결정을 하는데 전역 최적화 방법 중 하나인 Simulated Annealing(이하 SA) 기법을 사용하였다.

3. 교차로에서의 회전 및 통과 지연을 고려하기 위한 도로망 구축

H사의 조선소 야드 정보를 바탕으로 Ruy, et al. (2015)에서 개발된 조선 전용 GIS(Geographic Information System)(Fig. 1)를 이용하여 트랜스포터가 운행 가능한 주 도로를 따라서 주변에 가까운 위치에 있는 지번(적치장, PE장, 각종 조립장 등)과 연결하기 위한 절점(node point)을 정의하였다(Fig. 2참조). 이 절점들 간의 최단경로를 구한다면 해당 절점과 연결되어있는 지번까지의 최단경로를 확보할 수 있다고 가정하였다.

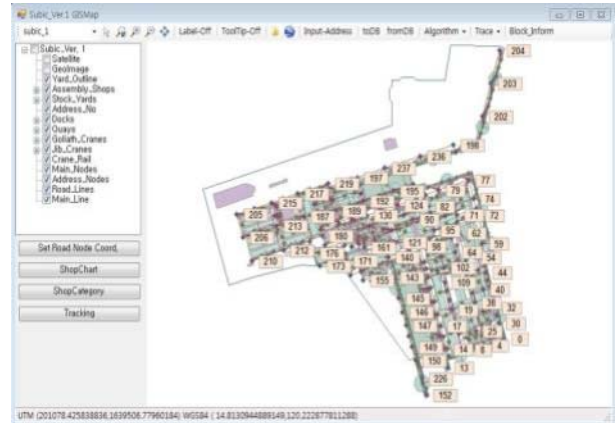


Fig. 1 GIS numerical map of a ship yard (Ruy, et al., 2015)

교차로에서 회전방향의 구분이 없던 기존의 교차로 도로망 구성을 보면(Fig. 2) 각 도로마다 중심에 하나의 간선만이 존재하고 각 간선 위에는 주변의 지번들과 연결하기 위한 절점들이 정의되어 있다. 트랜스포터는 이 간선들을 따라서 양방향으로 움직이는 것으로 가정하였다. 교차로에서는 이러한 간선들이 하나의 절점에 연결되어있는 것을 볼 수가 있다(Fig. 2의 사각점선 원안)

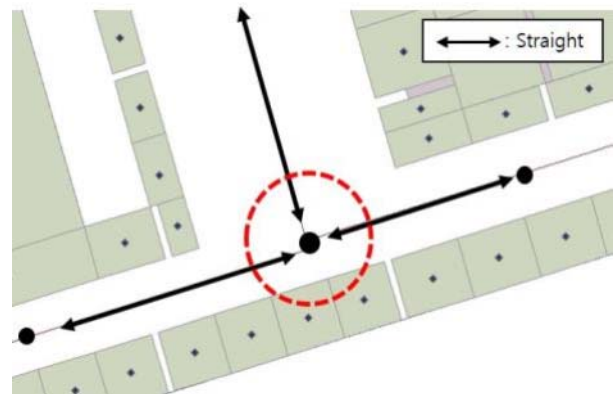


Fig. 2 Single road having the characteristics of Bi-directional road

Fig. 2에서와 같이 모델링된 도로망에 Dijkstra 알고리즘을 이용하여 각 절점 간의 최단 거리 정보를 계산하면 Table 1과 같은 결과를 얻을 수 있다. Table 1에서 절점 간의 최단 거리가 대칭 행렬을 이루고 있는 것을 볼 수가 있는데, 이는 교차로에서의 좌/우회전에 따른 경로의 차이가 고려되지 않았기 때문에 출발지와 도착지가 서로 바뀌어도 항상 같은 경로를 보여주기 때문이다.

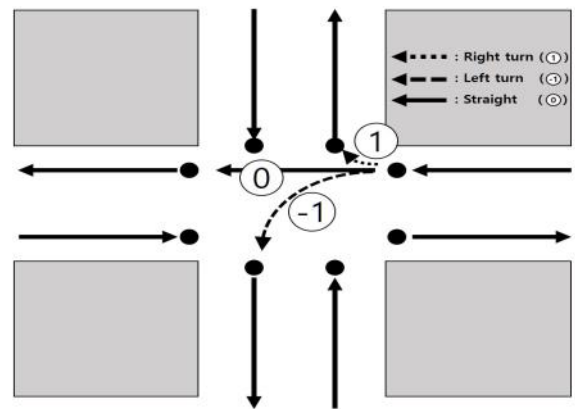
Table 1 Direction of the bi-directional main road

	1	2	3	4	...	238
1	0	17.75	86.19	103.94	...	2415.7
2	17.75	0	103.94	86.19	...	2398.0
3	86.19	103.94	0	169.94	...	2336.5
4	103.9	86.19	169.94	0	...	2336.4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
238	2415.7	2398.0	2336.5	2336.4	...	0

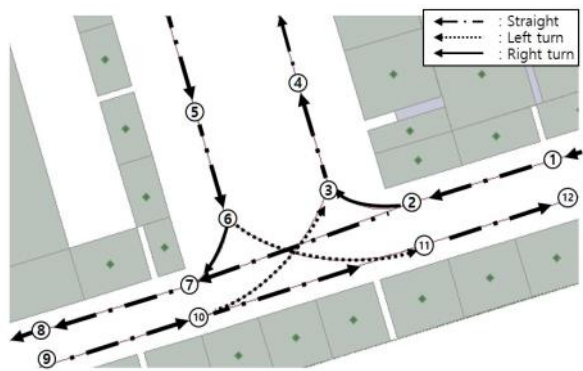
반면, 본 논문에서는 교차로에서의 회전에 따른 경로의 차이를 고려하기 위해 Fig. 3(b)처럼 각 도로에 두 개의 간선(사각 점선 화살표, 점선 화살표)을 생성하여 트랜스포터가 이동하는 방향에 따라서 경로를 분리하였으며, 교차로에서는 Park (1997)이 제안한 것처럼 Fig. 3의 (a)와 같이 직진, 좌회전, 우회전 경로를 구별하여 구성하였다. 이렇게 추가된 교차로에서의 직진, 좌회전, 우회전 경로에는 기본적으로 해당 경로의 이동 거리를 속성으로 갖지만, 교차로 구간에서의 회전 및 통과 시, 감속으로 인한 운송 지연 또는 통과 난이도 등을 고려하여 페널티를 도로 간선의 거리에 부여할 수도 있다(예를 들면, 교차로에서 회전시 속도를 절반으로 줄이는 경우 회전거리를 2배로 늘림. 상세한 내용은 4.1절을 참조)

이러한 방법으로 교차로 구간 도로망을 구성하여 Dijkstra 알고리즘을 수행하였을 때 Table 2와 같은 결과를 얻었는데 Table 1과는 다르게 비대칭 행렬인 것을 확인할 수가 있다. 이는 교차로 구간에서의 좌회전, 우회전의 간선 길이가 다르기 때문에(Fig. 3(a)의 교차로 구간 도로망) 절점 간의 최단거리가 비대칭 행렬의 모습을 보이는 것이다(참고로 Fig. 3(b)의 노드번호와 Table 2의 노드번호와는 같은 번호가 아니다. Fig. 3(b)의 노드번호는 Table 3을 설명하기 위한 노드번호이다.).

Table 3은 본 연구에서 사용하는 운송경로를 데이터베이스에 저장하는 방법을 보여주고 있다. 교차로 구간에서 좌회전, 우회전을 구별하기 위해 좌회전 경로에는 -1, 우회전은 1, 직진(교차로 통과)은 0의 값을 Intersection 속성 Column에 추가하였다. 간선에 방향성을 부여하는 방법의 예로 Table 3을 보면 First_Node와 Sec_Node Column에 간선의 시작 절점번호와 끝나는 절점 번호가 입력되어있는데 이것은 Fig. 3(b)에서 보면 ① (First_Node) → ②(Sec_Node)로 방향을 가지는 간선을 말해준다. 그리고 ②에서는 ③, ⑦로 이동할 수가 있다. ①과 연결되었을 때의 ②는 Sec_Node 이었지만, 다음 연결되어있는 절점으로 이동할 때의 ②는 First_Node가 된다(②(First_Node) → ③(Sec_Node), ②(First_Node) → ⑦(Sec_Node)).



(a) Road type of the intersection junction (Park,1997)



(b) Double road having the characteristics of uni-directional road

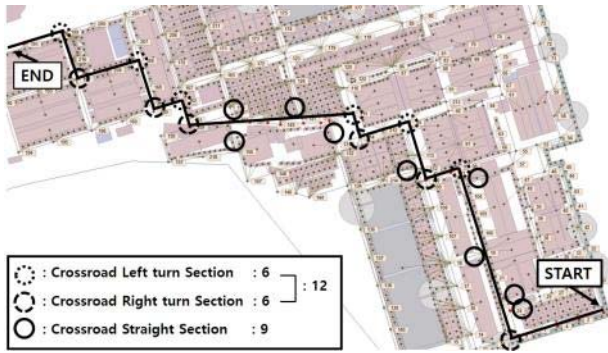
Fig. 3 Roads configuration considering turns and passing round the intersection

Table 2 Direction of the uni-directional main road

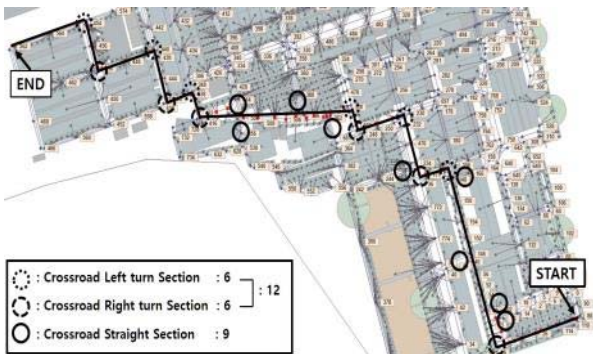
	1	2	3	4	...	788
1	0	776.2	565.9	380.3	...	1392.4
2	588.3	0	445.2	143.1	...	1358.9
3	143.1	355.1	0	498.2	...	1260.1
4	445.2	657.2	302.1	0	...	1221.2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
788	1257.5	1252.4	1114.3	1121.9	...	0

Table 3 Configuration of the main road on DB table

First_Node	Sec_Node	Distance	IsActive	...	Intersection
1	2	31.0	0	...	0
2	3	15.5	0	...	1
2	7	46.2	0	...	0
3	4	31.0	0	...	0
5	6	31.0	0	...	0
6	7	15.5	0	...	1
6	11	43.2	0	...	-1
7	8	31.0	0	...	0
9	10	30.0	0	...	0
10	3	43.2	0	...	-1
10	11	46.2	0	...	0
11	12	30.0	0	...	0



(a) Bi-directional path when turning is not considered at the intersection



(b) Uni-directional path when turning is considered at the intersection (Not a penalty applies; Left(L): 0%, Right(R): 0%, Straight(S): 0%)

Fig. 4 Shortest distance/path between the same set of origin and destination (Dijkstra algorithm)

Fig. 4는 도로 망 구성이 서로 다를 경우(양방향(bi-direction) vs 단방향(uni-direction))에 경로 탐색 결과를 비교한 것으로 동일한 경로를 보여주고 있다. 도로망 구성에서는 단방향 도로망 (Fig. 4(b))이 교차로에서의 회전 경로의 차이를 고려한 것이지만 회전에 특별한 페널티를 부여하지 않았기 때문에 양방향 도로망 (Fig. 4(a))과 동일한 것이다. 이 결과로부터 교차로를 고려하여 최단 경로를 탐색하는데 있어 도로망 구성에 따른 차이는 없으며, 다음 장에서부터 페널티를 적용할 수 있는 단방향 도로망을 기준으로 경로를 비교하려한다. 참고로 Fig. 4(b)에서의 “n%”의 의미는 속도의 감속 비율이며 다음 장에 자세히 상술된다.

4. 회전 구간 페널티 적용 경로 비교

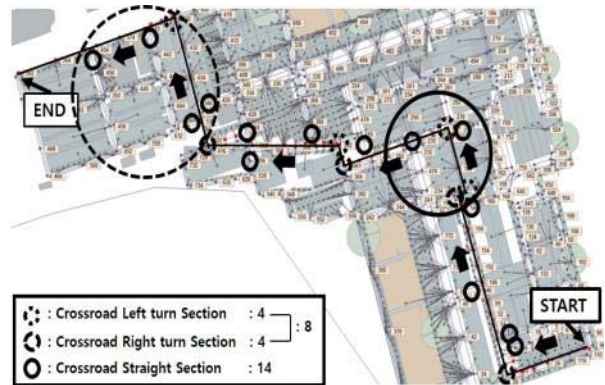
4.1 페널티 정의

최단 경로를 탐색하는데 있어 Dijkstra 알고리즘의 탐색 기준은 절점 간의 최단 거리이다. 절점 간의 최단 거리를 기준으로 경로를 탐색하기 때문에 경로 상의 회전구간 횟수는 중요하지가 않다 (회전 포함 여부와 관계없이 Manhattan distance는 같다). 하지만 실제 현장에서의 트랜스포터 이동 상황을 보았을 때 이동 경로 상의 교차로 구간에서 회전(좌/우회전)과 교차로 통과(직진)

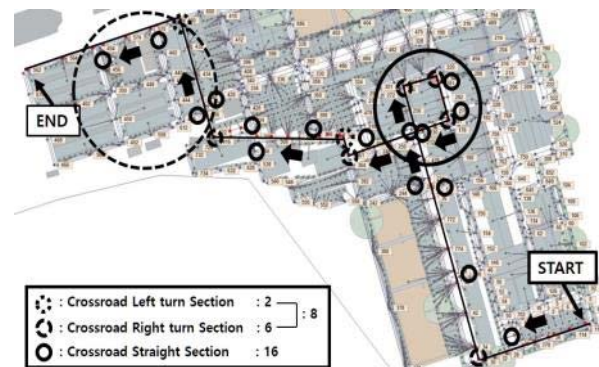
시 안전과 블록의 변형을 방지하기 위해 트랜스포터가 감속을 하게 된다. 이처럼 교차로 구간을 통과하는데 소요되는 시간은 트랜스포터의 전체 운행 소요시간 관점에서 중요한 부분이다. 이는 교차로 구간을 지나는 횟수에 따라 트랜스포터 전체 이동시간이 달라질 수 있음을 보여준다. 따라서 본 연구에서는 교차로에서의 회전 및 통과 실제 소요시간을 경로 탐색에 적용하는 방법으로 좌/우회전, 직진을 할 경우 각각의 간선 길이에 트랜스포터 정상 대비 이동속도의 비율(%)을 나누는 형식으로 페널티를 부과하였다. 예를 들어 좌회전 구간을 통과 할 경우 좌회전 간선의 길이가 10 m이고, 좌회전에 부과한 페널티가 트랜스포터의 교차로가 없는 직진 구간에서 평균 속도가 20 km/h일 때의 20%이면, 4 km/h로 움직이는 것이므로 결과적으로 좌회전 구간을 통과하는 거리는 10 m에서 50 m로 변하게 된다. 이렇게 회전 구간을 통과 할 때 회전 구간 간선의 길이를 증가시키면 절점 간의 최단 거리를 탐색하려고하는 Dijkstra 알고리즘으로 인해 교차로 구간에서의 회전 횟수가 최소가 되는 경로를 탐색할 것으로 예상하였다.

4.2 좌회전 구간 극한 페널티 적용

앞서 제안한 도로망 데이터베이스를 구축한 후, 동일한 조건 (출발지, 도착지가 같음)에서 좌회전 구간의 간선 길이에 각각 20%, 10%, 6.6%, 5%의 속도 감속 페널티를 주어 탐색된 경로를 비교하였다.



(a) L: 20%(same result with 10%), R: 0%, Straight: 0%



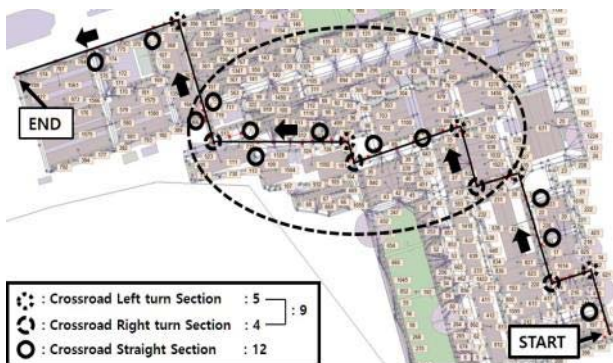
(b) L: 6.6%(same result with 5%), R: 0%, Straight: 0%

Fig. 5 Application of penalty for left turns

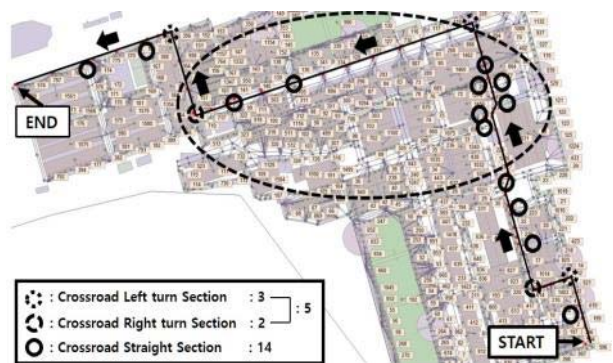
좌회전 구간에만 속도 감속 페널티를 부여하였을 때 경로는 Fig. 5와 같다. 페널티를 부여하지 않은 Fig. 4(b)와 Fig. 5의 경로를 비교해보면 Fig. 5에서 좌회전 구간에 해당하는 간선 길이에 각각 20%, 10%, 6.6%, 5%의 페널티를 주었기 때문에 좌회전 구간을 지나는 총 횡수(6회→4회, 2회)가 줄어드는 것을 확인할 수 있다. Fig. 5의 (a),(b)에서 큰 점선 원을 보면 동일한 위치에 Fig. 4(b)와 경로가 다른 것을 볼 수가 있다. 이는 좌회전 구간에 부여된 페널티로 인한 이동거리 증가로 좌회전 구간 통과를 최소화 하려고 하는 것이다. Fig. 5에서 좌회전 구간에만 페널티를 20%, 10% 적용한 경로(Fig. 5(a))와 6.6%, 5% 적용한 경로(Fig. 5(b))를 비교해 보면 Fig. 5(b)에서 큰 실선 원 안처럼 어색한 경로를 보여주는 것을 볼 수가 있는데 이는 좌회전에 부가한 페널티가 극단적으로 크기 때문에 좌회전 구간을 통과하는 횡수를 줄이려고 하다 보니 비교적 트랜스포터 감속이 적은 우회전의 횡수가 증가한 경우이다.

4.3 우회전 구간 극한 페널티 적용

우회전 구간에도 좌회전 구간에 속도 감속 페널티를 부여하였을 때와 동일한 조건(출발지, 도착지가 동일)으로 20%, 10%, 6.6%, 5%의 페널티를 각각 개별로 부여했을 때 경로를 비교하였다.



(a) L: 0%, R: 20%, Straight: 0%



(b) L:0%, R:10%(same result with 6.6% & 5%), Straight: 0%

Fig. 6 Application of penalty right turns

좌회전 구간에만 속도 감속 페널티를 적용한 Fig. 5와는 다른

경로를 보여주고 있다. Fig. 6에서 (a), (b)를 비교해보면 큰 점선 원 안에서 보여주는 경로가 다른 것을 볼 수가 있는데, 우회전 구간에 적용한 페널티 값이 증가함에 따라 교차로 구간에서 우회전을 하는 총 횡수가 감소(6회→4회, 2회)하는 것을 알 수가 있다. 따라서 교차로 구간에 페널티를 적용하였을 때, 경로를 탐색하는데 있어 페널티 값에 따라 좌우 교차 통과 횡수가 감소되는 것과 경로의 변화가 부분적으로 발생하는 것을 Fig. 5와 Fig. 6을 통해 확인할 수가 있다.

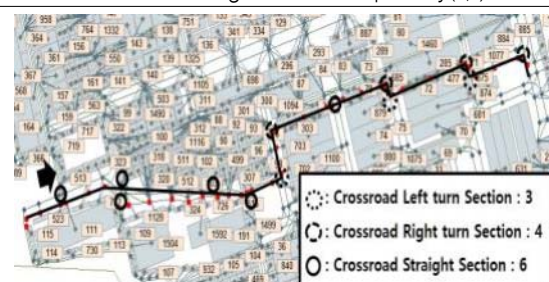
5. 트랜스포터 운영 계획 최적화

5.1 교차로에서의 트랜스포터 감속을 반영한 최적(최소비용) 탐색 결과

앞서 4.2, 4.3절에서 좌회전, 우회전 구간에 각각 속도 감속 페널티를 주었을 때 페널티를 부여한 회전 구간의 횡수가 감소하는 것을 확인할 수 있었다. 이번에는 좌/우회전, 직진 통과 구간에 각각 국내 조선소 현황 데이터를 반영한 페널티를 주어 경로를 비교하였다.

Table 4 Minimum cost distance vs manhattan distance

Departure	Arrival	Penalty	Minimum Cost Distance[m]	Manhattan Distance [m]
115	939	Left: 20% Right: 20% Straight:40%	2616.11(①)	1659.82(③)
		Left: 0% Right: 0% Straight: 0%	3184.28(②)	1463.59(④)
Difference [m]			(①-②)-568.2 (⑤) -17.8%	(③-④)196.2 (⑥) +13.4%
Not considering retardation penalty(↓)				



Considering retardation penalty(↓)

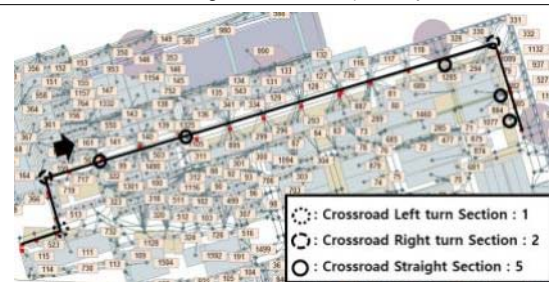


Table 4는 교차구간 지연을 고려한 전체 트랜스포터의 운행 계획 중, 한 대의 트랜스포터 이동 경로를 교차로 고려 여부에 따른 순수이동거리(Manhattan distance)와 최소비용거리(Minimum Cost Distance)를 비교한 결과이다. Table 4(①)는 교차로에서 회전으로 인한 지연 시간을 고려하여 도출한 최적경로의 비용거리이고, Table 4(③)는 해당 경로의 순수이동거리이다. Table 4(②)는 교차로에서의 지연을 고려하지 않고 도출한 최적경로의 비용거리이고, Table 4(④)는 이 경로의 순수이동거리이다. 여기서 교차로 지연을 고려한 경로와 고려하지 않은 경로를 순수 이동거리로만 비교하면 고려하지 않은 경로가 최단거리를 가지지만(⑥), 교차로 구간을 고려하지 않은 경로에서 교차로 구간 통과(좌/우회전, 직진통과)시 감속으로 인한 페널티를 동일하게 적용하여 비용 관점에서의 총 이동거리를 계산하여 비교해보면 -568.2[m]의(⑤) 차이가 발생하는데 이는 17.8% 감소한 것이다.

Table 5 Comparison of number of turns and passing cross roads between no penalty and penalty case (L:20%, R:20%, Straight:40%)

TP No	Intersection	No penalty case (①)	Penalty case (②)	Difference (① vs ②)
1	Left turn	58	58	0
	Right turn	57	56	-1
	Straight	81	123	+42
2	Left turn	54	29	-25
	Right turn	47	20	-27
	Straight	71	60	-11
⋮	⋮	⋮	⋮	⋮
7	Left turn	25	44	+19
	Right turn	23	36	+13
	Straight	39	85	+46
8	Left turn	53	42	-11
	Right turn	55	38	-17
	Straight	94	68	-26

이전까지 한 대의 트랜스포터 운행 결과를 비교하였는데 Table 5는 운행 가능한 전체 트랜스포터의 운영계획을 교차로 구간 고려 유무에 따라 좌/우회전, 직진 통과 횟수를 비교한 것이다. 트랜스포터 2번, 8번의 경우 좌/우회전/직진의 횟수가 감소

하는 것을 볼 수가 있다. 흥미로운 점은 횟수가 감소하는 트랜스포터가 있는 반면, 7번 트랜스포터처럼 횟수가 증가하는 경우도 있었다. 2번, 8번 트랜스포터처럼 횟수가 감소한 것은 페널티 적용으로 좌/우회전/직진 구간의 거리가 증가하였기 때문에 최소거리를 탐색하려는 Dijkstra 알고리즘의 특성상 최단경로를 탐색하는데 있어 교차로에서의 좌/우회전/직진 구간을 통과하는 횟수가 줄어든 것이다. 7번 트랜스포터의 경우 다른 트랜스포터와의 교차주행을 할 수 없는 목적함수 때문에 최단경로를 탐색하는데 있어 좌/우회전 횟수가 증가한 것으로 추정된다. 일부 회전 구간을 통과하는 횟수가 증가한 트랜스포터가 있지만 전체 트랜스포터의 회전 통과구간 횟수를 비교해보면 전체적으로 감소한 것을 알 수가 있다(Table 6).

Table 6 Comparison of total turns and passing in case of all transporters (L:20%, R:20%, Straight:40%)

	No penalty case	Penalty case	Ratio
Left turn	379	274	about 27.7% decrease
Right turn	404	366	about 9.4% decrease
Straight	701	712	about 1.5% increase

5.2 운영 계획 최적화

지금까지 블록 운송 계획을 세우기 전 각각 블록의 출발점과 도착지점간의 최단거리와 교차로에서의 회전, 직진의 페널티를 고려한 최소 비용 경로를 주로 언급하였다. 다음은 실제 블록 운송 정보와 트랜스포터 운영 정보를 입력하여 실용적 트랜스포터의 최적 운영계획을 수립하였다.

Table 7 Block transportation data of a shipyard

Ship ID.	Block ID.	Time		Departure	Arrival
		Min departure	Max arrival		
NCP 0111	1101	13:00	14:30	Z00-002	PNA-350
	⋮	⋮	⋮	⋮	⋮
	2104	10:30	10:50	ASB-25A	PSD-C17
NCP 0120	0C01	8:00	8:30	BP-64	PNA-340
	⋮	⋮	⋮	⋮	⋮
	1406	10:30	13:40	OBA-N02	PNA-S04
⋮	⋮	⋮	⋮	⋮	⋮
NLP0 101	1306	13:18	13:50	BSB-C14	BP-64
POP 0099	2401	8:00	20:00	PSF-C04	HUA-13A

Table 8 Transporter data of a shipyard

TP ID.	Max ability	Velocity		Initial position	Is active	Date
		Unload	Load			
TP01	200	14	5	BPA-N10	0	8:00
TP02	250	14	5	BPA-N12	0	8:00
⋮	⋮	⋮	⋮	⋮	⋮	⋮
TP11	700	11	4.5	BPA-N10	1	8:00
TP12	700	11	4.5	BPA-N10	1	8:00

Table 7, Table 8은 본 논문에서 적용한 트랜스포터의 운행 정보와 운송 블록 정보이다. 트랜스포터 정보에는 트랜스포터가 운반할 수 있는 최대 중량과 운행속도, 운행가능여부, 운행시작 시간 등이 있다. 블록 정보에는 출발지점, 도착지점, 최소 출발시간, 최대 도착시간 등이 나열되어있다. 이 정보를 가지고 트랜스포터가 어떤 블록을 할당 받고 어떤 블록 순서로 운영하느냐에 따라 트랜스포터의 최적 운행 계획이 수립되게 될 것이다. 트랜스포터의 최적 블록 운송 계획을 세우는 목적은 트랜스포터의 공주행시간을 최소화하는 것이다. 최적 경로만으로 운송된다는 가정 하에서 블록 운반 시간은 출발지와 도착지가 고정되어있기 때문에 고정된 시간으로 볼 수가 있다. 하지만 블록을 운반한 후 공차상태(상차되지 않은 트랜스포터)에서 다음 운송 대상 블록이 있는 지번까지 이동하는 공차이동시간은 다음 블록의 선택에 따라 크게 달라질 수 있다. 결국 트랜스포터의 총 공차이동시간을 줄이는 것이 최적 블록 운송 계획을 수립하는 것이다.

트랜스포터에 블록을 할당하고 할당된 블록들의 운반 순서를 결정하는 문제를 최적화 문제에 적용하였다. 목적함수는 트랜스포터의 공차 이동 시간을 최소화하도록 하면서 교차로 구간에서 좌/우회전/직진의 횟수를 최소화 하는 것이며(식 (1)), 제약조건으로는 운송 블록의 무게가 트랜스포터의 적재 능력보다 작아야 하며(식 (2)), 모든 블록은 운송 완료 제한 시간 내에 완료(식 (4))된다고 가정하였다. 또한, 모든 블록은 운반 시작 가능 시간 이후에 운송을 시작해야 하고(식 (3)), 우선순위에 따라 블록을 운송(식 (5))해야 한다. 우선순위는 해당 블록 출발 요청시간으로 결정하였다. 마지막으로 식(6)에서는 한 도로에 존재하는 트랜스포터의 개수를 2이하로 제한하여 교차 주행을 방지하였다. 이 제약조건들을 목적함수에 적용하기 위해 페널티 방법을 사용하였다. 블록의 중량과 트랜스포터의 적재능력 간의 부등식, 제한 시간과 실제 운송 시작/완료 시간 간의 부등식을 비교하여 위배하는 경우 페널티를 부여하는 방식을 사용하였다.

Minimize

$$F = \alpha \sum_{h=1}^{n_B} \sum_{k=1}^{n_T} \sum_{i=1}^{n_T} x_{hi}^k \left(\frac{e_{hi}^k}{V^k} \right) \quad (1)$$

Subject to

$$w_i \leq c_k \quad (2)$$

$$r_i \leq p_i^k \quad (3)$$

$$d_i^k \leq s_i \quad (4)$$

$$p_j \leq p_i \quad (5)$$

$$N_{mn}^t < 2 \quad (6)$$

$$i, j = 1, \dots, n_B$$

$$k = 1, \dots, n_T$$

$$h = 1, 2, \dots, n_B \text{ (단, } h \neq i \text{)}$$

where,

e_{hi}^k : 트랜스포터 k 가 블록 h 를 운반 후 블록 i 위치로 이동할 때 교차로 구간 통과를 고려한 최소 비용 거리.

V^k : 트랜스포터 k 의 속력

$x_{hi}^k = 1$: 트랜스포터 k 가 블록 h 를 운반 후 블록 i 를 운반 할 경우

$x_{hi}^k = 0$: 트랜스포터 k 가 블록 h 를 운반 후 블록 i 를 운반하지 않을 경우

N_{mn}^t : 절점 m, n 사이에 시간 t 에서 통과하는 TP 수

w_i : 블록 i 의 중량

c_k : 트랜스포터 k 가 운반 할 수 있는 적재중량

r_i : 블록 i 의 계획된 운반가능 시간

p_i^k : 트랜스포터 k 가 블록 i 를 운반하는 시작 시간

d_i^k : 트랜스포터 k 가 블록 i 를 운반 완료한 시간

s_i : 블록 i 의 계획된 운반 완료 시간

p_i : 우선순위가 높은 블록 i 의 운반 시작 시간

p_j : 우선순위가 낮은 블록 j 의 운반 시작 시간

n_B : 운반해야 할 블록의 총 수

n_T : 사용가능한 트랜스포터의 총 수

α : 가중치 (weight factor)

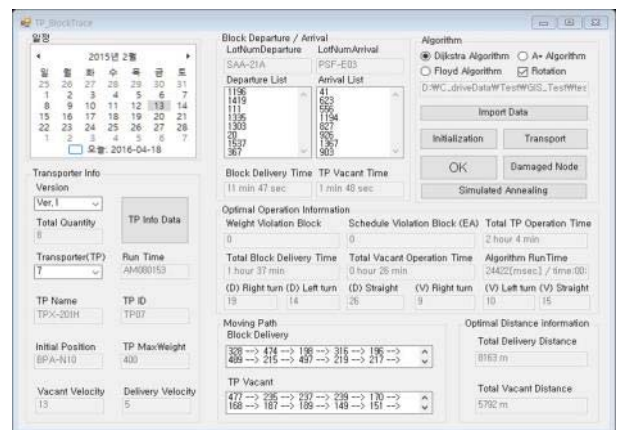


Fig. 7 Optimum block transportation planning using SA.

Fig. 7은 Table 7, Table 8의 트랜스포터 운행 정보와 블록 정보를 입력 받아 SA를 이용한 최적화를 수행한 후의 관심 트랜스포터의 블록 운반 최단경로와 공차운행 최단경로를 보여주고 있다.

Table 9 The result of transporters' operational plan (L: 20%, R: 20%, Straight: 40%)

	Delivery transportation		Vacant transportation	
	No penalty case	Penalty case	No penalty case	Penalty case
Left turn	171	127	208	147
Right turn	204	188	200	178
Straight	311	353	390	359

Table 10 Comparison between delivery and vacant distance in case of Table 9 (L: 20%, R: 20%, Straight: 40%)

	Delivery transportation [m]		Vacant transportation [m]	
	No Penalty	Penalty	No Penalty	Penalty
Manhattan Distance[m]	84,127	90,112	89,893	97,949
Min. Cost Distance[m]	147,906	132,962	157,604	131,395

Table 9는 교차로 구간에서 속도 감속 페널티를 적용하지 않을 때와 적용하였을 때를 각각 최적화한 것으로 블록 운반 운행과 공주행 일 때 좌/우회전, 직진의 횟수를 비교한 것이다. 블록 운반 운행 시 교차로 구간에 페널티를 적용하였을 때(좌회전: 20%, 우회전: 20%, 직진: 40%) 좌회전 구간은 약 26%, 우회전 구간은 약 8% 감소하였고, 직진 구간은 약 14% 증가하였다. 공차운행은 좌회전 구간 29%, 우회전 구간 11%, 직진 약 8% 감소하였다. 회전 구간의 적용한 페널티로 인해 회전구간을 통과하는 횟수를 최소화하려는 것을 알 수가 있다. 이들을 종합하여 Table 10에서는 Table 9와 동일한 설정으로 페널티 적용 여부에 따라 각각의 상하주행과 공주행거리를 비교했다. Manhattan 거리 차원에서는 증가(공주행: 108%, 상차주행: 107%)는 예상된 바였지만 운송비용 차원에서 확인한 개선 경향(공주행: 83%, 상차주행: 90%)을 확인할 수 있었다.

6. 결론

본 연구는 야드 현장의 실질적인 다양한 조건(공주행거리 기

준 최적화, 손상 구간 적용, 교차주행 금지, 교차로에서의 회전 및 직진 통과 시에 트랜스포터의 운행 속도 감속)을 고려한 트랜스포터 운행 시스템을 구축하려는 일환에서 교차로에서의 회전 및 직진 통과 시에 발생하는 시간 지연에 관한 문제를 적용하여 탐색된 트랜스포터의 최적경로가 최소비용거리를 가지는 결과를 검증하였다. 본 연구를 통해서 조선소 현실을 반영한 TP의 운행 거리 반영의 필요성을 검증하였으며, 단순한 이동 거리 차원의 접근이 아닌 최소비용거리 관점에서의 최적 경로를 획득할 수 있는 방법론을 정리할 수 있었다. 이어 이를 반영한 TP 운영 계획 시스템을 개선할 수 있었다.

후기

이 연구는 2017년도 산업통상자원부 및 산업기술평가관리원 (KEIT) 연구비 지원(10067423)과 미래창조과학부 ICT융합 Industry4.0S(조선해양) 기술개발 사업 “해양조선 생산 코스트 최적화 생산 전략 및 실행 시뮬레이션 기술 개발” 과제 (S1106-16-1020)의 지원을 받아 수행하였습니다.

References

- Cha, J.H. Cho, D.Y. Song, H.C. & Roh, M.I., 2012. Development and application of optimal block transportation simulation system of transporters in shipyard. *Proceedings of the Society of CAD/CAM engineers conference*, Pyeongchang, Korea, 1-3 February 2012.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1). pp.269-271.
- Heo, Y.J. Cha, J.H. Cho, D.Y. & Song, H.C., 2013. Optimal block transportation path planning of transporters considering the damaged path. *Journal of the Society of Naval Architects of Korea*, 50(5), pp.298-306.
- Joo, C.M. Lee, W.S. & Lee, K.B., 2005. Transporter scheduling for block transportation in the shipyard. *Conference of Korean Operations Research and Management Science*, Cheongju, Republic of Korea, 13-14 May 2005, pp.348-352.
- Maxwell, W.L. & Muckstadt, J.A., 1982. Design of automatic guided vehicle system. *IIE Transactions*, 14(2), pp.114-124.
- Moon, J.H. Ruy, W.S. & Cha, J.H., 2016. Comparison of Optimal Path Algorithms and Implementation of Block Transporter Planning System. *Journal of the Society of Naval Architects*

- of Korea*, 53(2), pp.115–126.
- Park, S.C., 1997. *An evaluation method for one-way systems using shortest path algorithms*. Master's Thesis. University of Mokwon.
- Roh, M.I. & Cha, J.H., 2011. A block transportation scheduling system considering a minimisation of travel distance without loading of and interference between multiple transporters. *International Journal of Production Research*, 49(11), pp.3231–3250.
- Ruy, W.S., 2015. An optimization of process planning around quays based on the yard customized GIS and the simulator. *Journal of the Society of Naval Architects of Korea*, 52(2), pp.97–103.
- Yim, S.B. Roh, M.I. Cha, J.H. & Lee, K.Y., 2008. Optimal block transportation scheduling considering the minimization of the travel distance without overload of a transporter. *Journal of the Society of Naval Architects of Korea*, 45(6), pp.646–655.

- Yu, H.K., 2005. *A study on the transportation routing optimization of shipbuilding blocks*. Master's Thesis. University of Ulsan.

