Management Science Design

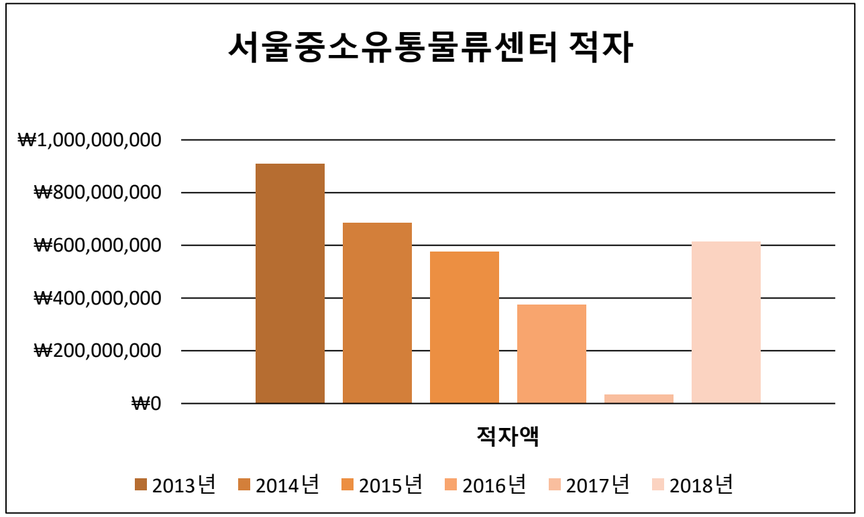
Improved A* Algorithm을 이용한 중소물류센터에서의 AGV 경로 최적화

2016147016 신우현, 2016147026 김민수, 2017199098 전영목, 2020147051 장건, 2022849427 손한나

CONTENTS

1) Problem Statement

서울 중소유통 물류센터



서울 중소유통 물류센터란?

- 중소 슈퍼마켓 및 골목가게에 저렴한 가격으로 상품을 공급하는 물류센터
- 3,372㎡의 크기로 서초구 양재대로 12길 36(양재동 양곡도매시장 내)에 위치
- 슈퍼마켓 및 골목가게로부터 낱개의 상품들을 주문 받아 배송

서울 중소유통물류센터의 문제점

- 2013년 설립이후 수년동안 적자를 면치 못하고 있음
- 물류시스템 및 정보 인프라 구축 미흡이 매출부재의 원인

1) Problem Statement

주제: 서울 중소유통 물류센터의 AGV 도입을 위한 AGV Path Planning

주제 선정 이유: 유통업의 AGV 활용을 통한 경쟁력 강화

- AGV 도입을 통한 물류시스템 개선(+인건비 절감)
- 유통대기업과 달리, 유통중소기업에게는 AGV 운영비용 최적화 문제로 인하여 진입장벽이 높음
- 따라서, 중소물류센터에서의 AGV 운영비용 최소화를 위해서도 AGV 경로 최적화를 구현하고자 함

서울 중소유통 물류센터 물류시스템 개선을 위해

AGV 도입 시 필요한 경로 최적화를 구현하는 것이 목적

2) Background

What's A* Algorithm?

- 주어진 출발 노드에서부터 목표 노드까지 가는 최단경로를 찾아내는 그래프 탐색 알고리즘
- 목표 노드 없이 출발 노드에 가까운 노드부터 순서대로 결정해 최단경로를 구하는 Dijkstra's Algorithm을 발전시킨 알고리즘

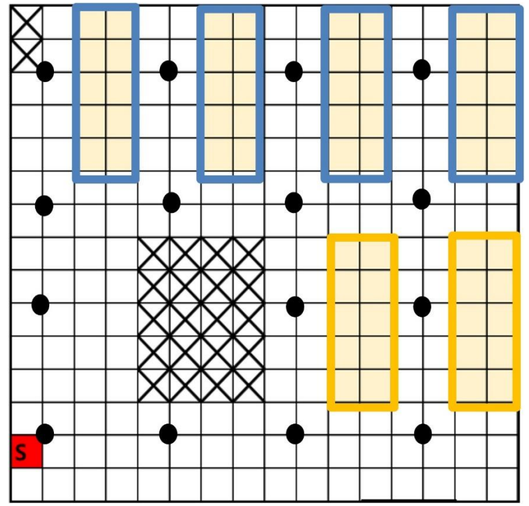
Why A* Algorithm?

- 최단 경로 문제를 풀기위해 대표적으로 사용되는 두가지 알고리즘: Dijkstra's Algorithm과 A* Algorithm
- Dijkstra's Algorithm이 아닌 A* Algorithm을 improve시켜 사용하기로 결정한 이유:
  - Dijkstra's Algorithm은 "목표 노드"가 존재하지 않아, "목표 노드"뿐만 아니라 모든 노드를 고려하여 최단경로를 찾아야하기 때문에 효율성이 떨어질 것이라 판단
  - Multi AGV path planning시, Dijkstra's Algorithm을 통해 찾은 최적 경로가 A* Algorithm을 이용한 최적 경로의 약 두배의 시간이 걸림을 입증하는 논문을 참고하여 결정*(Yuan, Z et al)*

3) Setting

I. 중소물류센터 환경 설정

- 실제 물류센터 방문 후 도면화 → 노드화 시킨 노드맵 사용



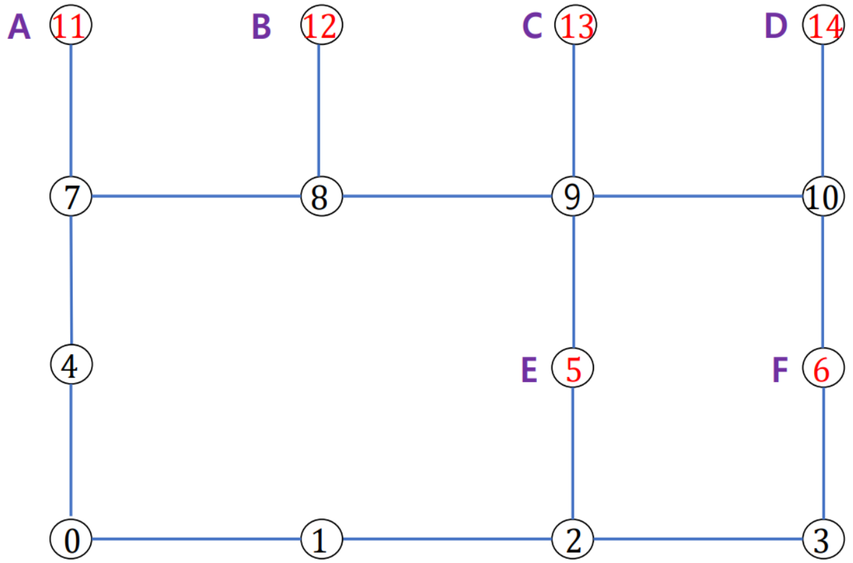
0 = 입구/출구 노드

5, 6, 11, 12 ,13, 14 = AGV가

물품을 픽업하는 노드

2, 3, 7, 8, 9, 10 = 충돌/교차가

일어날 수 있는 노드



3) Setting

I. 중소물류센터 환경 설정

- 서울 중소유통 물류센터와 동일하게, 6개의 Inventory(**5, 6, 11, 12 ,13, 14**)에는 각각 다른 브랜드의 쌀이 재고됨

현장방문 후 직접 찍은 서울 중소유통 물류센터의 사진들

"슈퍼마켓 및 골목가게로부터
낱개의 상품들을 주문 받아 배송"

AGV에게 매일 다른
Random한 오더가 주어짐

3) Setting

I. 중소물류센터 환경 설정

- 물류센터의 내부자료를 활용하여 6분당(1시간에 오더 10개) 처리해야하는 물류량 계산

[표 2-6] 서울시 중소유통물류센터 취급상품 수 대비 매출액 현황

(단위: 개, 백만 원)

대분류	중분류 명	상 품 수			매 출 액		
		2013년	2014년	2015년	2013년	2014년	2015년
가공식품	캔류, 레토르트, 씨리얼, 프리믹스, 면류, 식품기타	1,040	1,319	1,225	1,280	1,701	1,388
조미료	조미료류, 소스류, 장류, 케찹, 마요네즈, 식용유	1,272	1,613	1,499	1,566	2,080	1,698
기호식품	커피, 차, 스넥, 과자, 사탕, 초콜릿, 안주	687	784	893	294	412	413
음 료	탄산, 과즙, 이온, 기능, 건강, 음료기타	360	387	306	496	566	696
주 류	소주, 맥주, 꼬냑, 위스키, 민속주, 와인, 기타	-	-	264	-	-	6,011
세제류	세탁, 마용세제, 섬유린스, 주방세제, 표백 · 청소제	683	745	639	382	346	338
위생용품	화장용품, 위생용품, 화장지, 생리대	45	80	72	54	111	97
일배식품	유제품, 두부, 콩나물, 어묵맛살	-	105	127	-	12	63
기타 일차식품	양곡, 잡곡, 건어물 등	307	416	277	183	173	131
잡화류	생활잡화류	26	819	868	110	193	80
합 계		4,420	6,268	6,170	4,365	5,594	10,915

자료: 서울시 중소유통물류센터 내부자료, 2016

연매출: 100억
쌀(10kg) 가격: 28,000원
1년 영업일: 12 x 20 = 240
1일 영업시간: 8시간
AGV 운송가능량: 약 500kg

$$\text{시간당 물류량} = \frac{\text{연매출}}{\text{쌀}(10kg)\ \text{가격}} \div (\text{1년 영업일}) \div (\text{1일 영업시간})$$

$$\text{AGV 최소이동횟수}\left(\frac{Demand}{Capacity}\right) = \text{시간당 물류량} \div \text{AGV 운송가능량}$$

∴ 시간당 물류량 ≒ 100,000kg; AGV 최소이동횟수(1h) ≒ 180번
∴ 6분당 물류량 ≒ 16,667kg; **AGV 최소이동횟수(6min) ≒ 18번**

3) Setting

II. AGV 설정

- AGV기능 및 수행해야 할 Task 설정

AGV 기능

- 이동: 수평 및 수직 방향
- 속도: 3.6km/h(1m/s)
- Loading시간: 10s
- Rotating시간: 0s (Omni-directional AGV)

Given Task

- Task가 주어지는 시간: every 6 minutes
- AGV의 개수: Task를 소화할 수 있는 최소한의 수로 결정
- 한번에 한 Inventory작업 수행; 여러 군데 거치지 않음

(쌀 stack운반은 AGV의 capacity가 거의 full로 사용됨)

- Starting Point(0)에서 동시출발 불가능, 동시도착 가능

4-1) Module1-Key Idea

Key Idea 1. 유지비용 최소화 by AGV 이동거리 최소화

물류센터 내 유지비용 최소화를 위하여 주어진 물류량을 모두 처리 가능한 AGV 총 이동거리 최솟값을 IP모델링으로 계산한다.

Key Idea 2. 도입비용 최소화 by Min # of AGVs Needed

AGV 도입비용 최소화를 위하여 AGV 총 이동거리과 AGV 평균시속을 이용하여 필요한 AGV의 최소 개수를 구한다.

4-2) Module1-Method

IP Modeling Method

Objective

Min $60\,x_{9,1} + 124\,x_{9,2} + 156\,x_{9,3} + 76\,x_{10,1} + 108\,x_{10,2} + 140\,x_{10,3} + 92\,x_{11,1} + 92\,x_{11,2} + 124\,x_{11,3}$
$+106\,x_{12,1} + 108\,x_{12,2} + 108\,x_{12,3} + 92\,x_{3,1} + 64\,x_{3,2} + 106\,x_{4,1} + 108\,x_{4,2} + 80\,x_{4,3}$

Constraints

$\sum_{j}^{n} x_j \geq \frac{D_p}{C}$

x_j = Number of travels on route j to reach product p

n = Number of all routes that reach product p

D_p = Demand for product p

C = Carrying capacity per AGV

$x_h \leq k \sum_{h}^{q} x_h$

x_h = Number of travels on all routes that pass through node h

q = Number of alternate nodes

k = Bottleneck prevention coefficient

S.T

- $\sum_j \sum_i x_{i,j} = \frac{\text{demand}}{\text{capacity}} = 200$
- $x_n = 20$ (n=3,9), $x_m = 40$ (m=10,11,12,4)
- (1,5 alter): $x_1' \leq k \sum x_{i,j}$
- (2,7 alter): $x_7' \leq k \sum x_{i,j}$

k=0.7

4-2) Module1-Method

What is a Bottleneck Prevention Coefficient?

$$x_h \le k \sum_{h}^{q} x_h$$

x_h = Number of travels on all routes that pass through node h

q = Number of alternate nodes

k = Bottleneck prevention coefficient

한 쪽 경로로 몰릴 경우 AGV의 충돌 가능성 ↑
∴ 특정 경로 선택비율을 k 로 제약

```
>> IPhome
LP:                 Optimal objective value is 16640.000000.
```

<k=90%~70%>, 16640으로 값 동일

=> **k= 0.7**

```
>> IPhome
LP:                 Optimal objective value is 16660.000000.
```

<k=60%>, 총 이동거리 증가

5) Module1-Result

IP Modeling: Matlab Code

```
f=[60;124;156;76;108;140;92;92;124;106;108;108;92;64;106;108;80];
N=17;
A=[0,1,1,0,1,1,0,1,1,0,1,1,0,1,0,1,1;0,1,0,0,1,0,1,1,0,1,1,0,1,0,1,1,0];
b=[140;140];
Aeq=[1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
    0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0;
    0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0;
    0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0;
    0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0;
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1];
beq=[20;40;40;40;20;40];
lb=zeros(N,1);
intcon=1:17;

x = intlinprog(f,intcon,A,b,Aeq,beq,lb)
[x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb)
```

5) Module1-Result

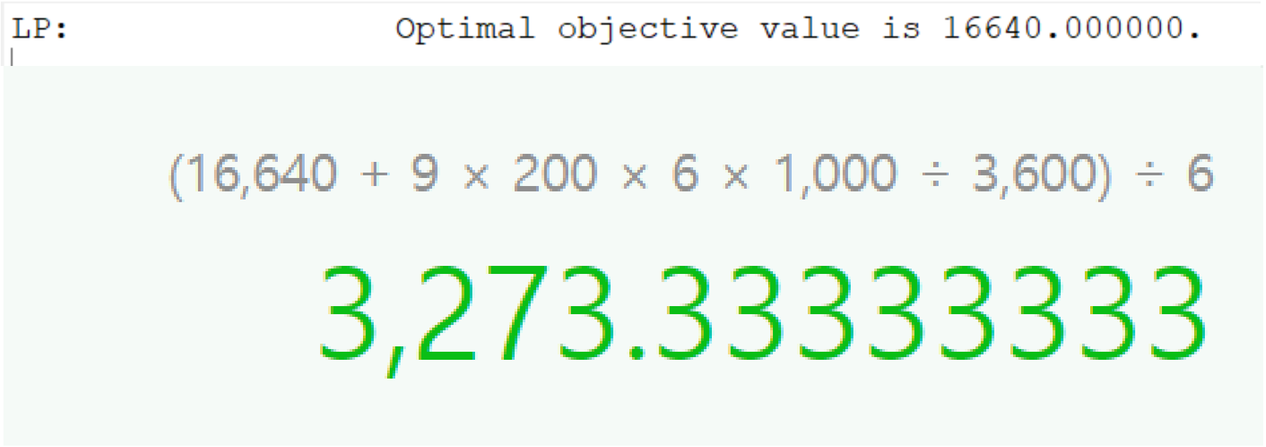
IP Modeling: Code Results → Number of Each Path Usage & Total Distance

```
x =

    20
     0
     0
    40
     0
     0
     0
    40
     0
    40
     0
     0
     0
    20
     0
     0
    40

LP:                Optimal objective value is 16640.000000.
```

Objective Value(AGV 총 이동거리) =16,440m

5) Module1-Result

Calculate Total Distance → Minimum Number of AGVs Needed



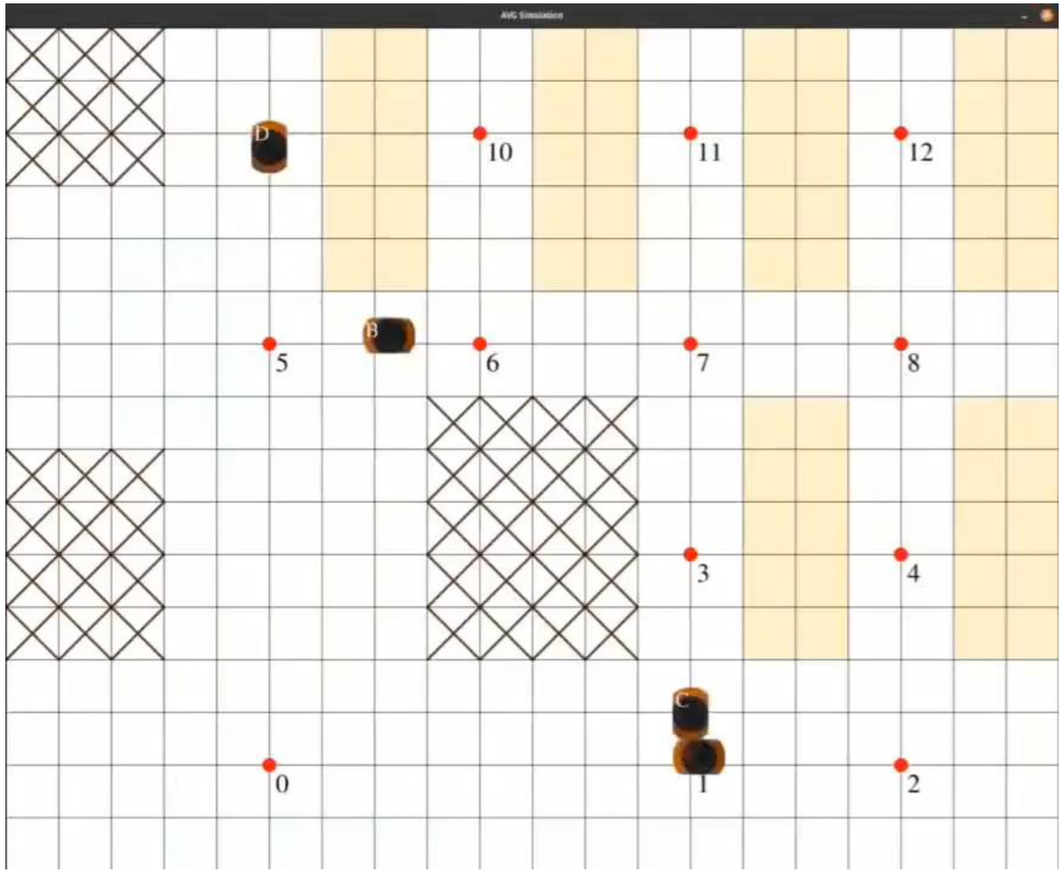
$$\frac{\text{총 이동거리+적재시간동안 이동가능거리}}{\text{AGV 이동속도(시속)}}$$

= 물류량 처리를 위한 최소 AGV수

- 대체경로들의 표준 편차 평균값= 18.1m
- 경로를 수정해도 최적 AGV수는 4개

5) Module1-Result

Module1 AGV simulation (영상 재생 가능)



6-1) Module2-Key Idea

1 For a given task list (assigned every 6 minutes), obtain reasonable routes for the AGVs to be introduced in the target warehouse.

2 Optimize AGV routes utilizing our Improved A* Algorithm.

3 Compare the performance of Unaltered A* Algorithm vs. Improved A* Algorithm by comparing makespan and number of collisions.

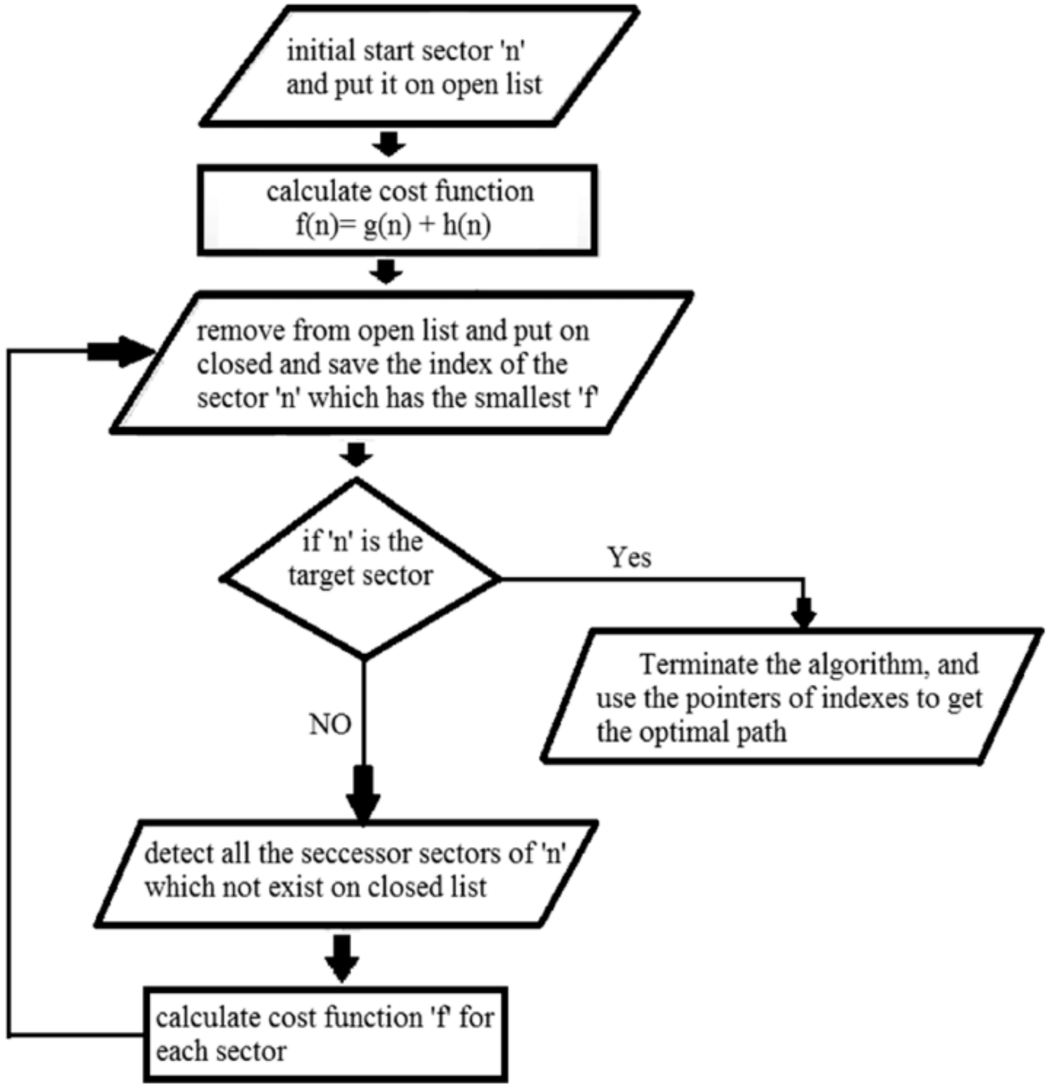
6-2) Module2-Method

1. A* Algorithm

- 주어진 출발 노드에서부터 목표 노드까지 가는 최단경로를 찾아내는 그래프 탐색 알고리즘
- f(n) = g(n) + h(n)식을 사용하여 계산

f(n) = g(n) + h(n)

- g(n): exact cost of the path from the starting node to current node n
- h(n): heuristic estimated cost from current node n to the target node



6-2) Module2-Method

1. A* Algorithm

```
def a_star(start_edge, blocked=None):
    edge_dict = {}
    edge_dict[f"{start_edge}"] = 0

    mod_allpaths = all_paths.copy()

    if blocked:
        a, b = blocked.split(",")
        mod_allpaths.remove(blocked)
        mod_allpaths.remove(f"{b},{a}")

    while True:
        for path in mod_allpaths:
            a, b = path.split(",")
            edge_dict[a] = edge_dict[b] + eucledian(a, b, edge_dict)

            edge_dict.pop(b)

        if not edge_dict:
            break

    return edge_dict
```

f(n) = g(n) + h(n)

계산 방법

1. 연속적인 시계열을 5초 단위의 이산적인 시계열로 분할
2. A*Algorithm 코드 구현을 통해 각 노드의 최적 경로 계산
3. 충돌 발생 시 오더 우선순위에 따라 AGV 1대는 대기
4. AGV들의 Makespan 계산

6-2) Module2-Method

2. Improved A* Algorithm 개요

Improved A*Algorithm = A*Algorithm + Heuristic

기존 A*Algorithm의 한계

시계열과 우회경로를 고려하지 못함

개선 방안

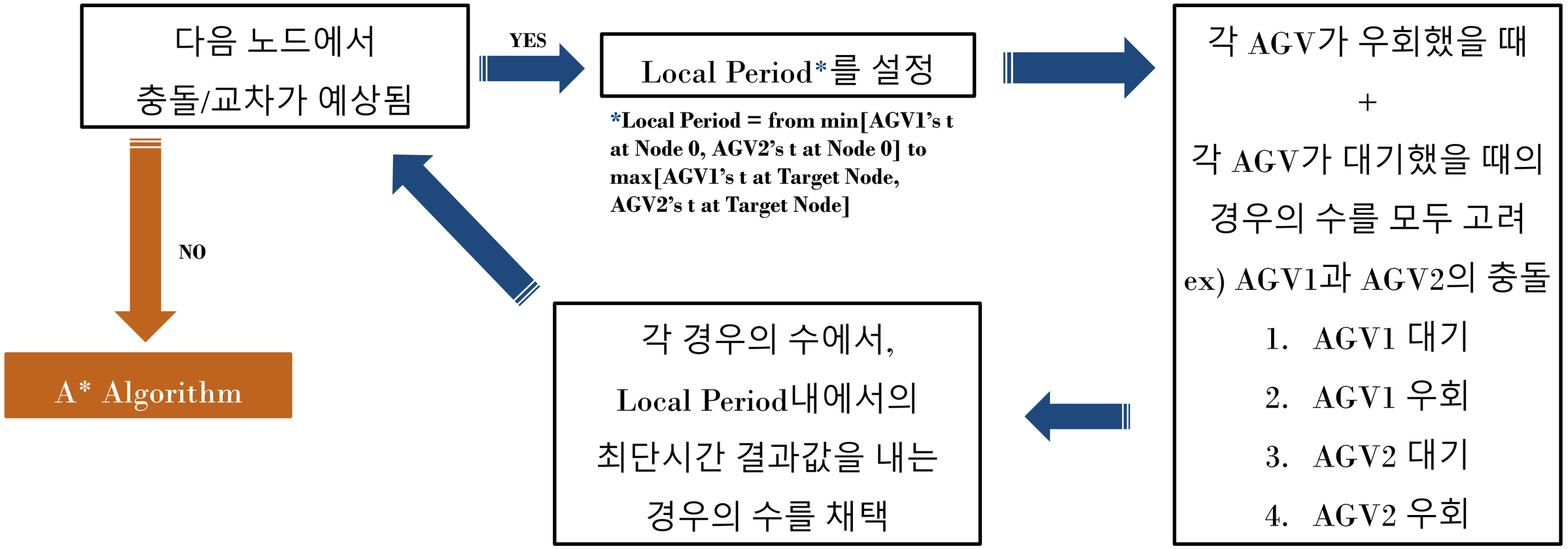
시계열과 우회를 고려한 휴리스틱을 사용하여 **A*Algorithm** 개선

Improved A* Algorithm 요약

전체 시계열 내 충돌 방지와 경로 최적화를 위한 경우의 수 계산은 **NP-Hard**이기 때문에, 휴리스틱을 통해 **Local Optimal Solution**을 사용한 **Global Optimal Solution** 근사값을 도출

6-2) Module2-Method

2. Improved A* Algorithm



7) Module2-Results

0. AGV에게 할당되는 오더

- 각 Inventory Node에 주문되는 횟수는 3회로 **동일**

- 각 Inventory node의 Order 순서는 **Random Order**

  (실제 물류센터 Order 순서 x , 물류 트럭 도착순 Order)

	Order 1	Order 2	Order 3
1번	A	A	B
2번	B	B	A
3번	E	A	B
4번	C	D	C
5번	F	B	A
6번	D	C	E
7번	B	D	A
8번	D	C	C
9번	C	E	F
10번	E	A	E
11번	A	F	C
12번	F	B	D
13번	D	E	F
14번	F	F	E
15번	A	C	D
16번	C	D	B
17번	B	E	F
18번	E	F	D

7) Module2-Results

1. AGV가 2대인 경우 – A* Algorithm

t = n
t(seconds) = 5n seconds

t	0	1	2	3	4	5	6
agv1	0	4	7	11	11	11	7
agv2	0	0	1	2	5	5	5

•
•
•

91	92	93	94	95	96	97	98
4							
0	2	5	5	5	2	1	0

충돌 대기 횟수	처리 시간
1회	490초

7) Module2-Results

1. AGV가 2대인 경우 – Improved A* Algorithm

t = n
t(seconds) = 5n seconds

t	0	1	2	3	4	5	6
agv1	0	4	7	11	11	11	7
agv2	0	0	1	2	5	5	5

•
•
•

91	92	93	94	95	96	97
2	5	5	5	2	1	0

충돌 대기 횟수	처리 시간
0 회	485초

두 알고리즘 모두

360초(6분)안에 물류 처리 실패

⇒ **AGV 추가 투입 필요**

7) Module2-Results

2. AGV가 3대인 경우 - A* Algorithm

t = n
t(seconds) = 5n seconds

t	0	1	2	3	4	5	6
agv1	0	4	7	8	12	12	12
agv2	0	0	4	7	11	11	11
agv3	0	0	0	4	7	8	8

•
•
•

64	65	66	67	68	69	70	71	72	73	74
9	10	14	14	14	10	9	8	7	4	0
1	0									
0										

충돌 대기 횟수	처리 시간
8 회	370초

7) Module2-Results

2. AGV가 3대인 경우 – Improved A* Algorithm

t = n
t(seconds) = 5n seconds

t	0	1	2	3	4	5	6
agv1	0	4	7	8	12	12	12
agv2	0	0	4	7	11	11	11
agv3	0	0	0	4	7	8	9

•
•
•

61	62	63	64	65	66	67	68	69	70
5	2	1	0						
1	2	3	6	6	6	3	2	1	0
9	8	7	4	0					

충돌 대기 횟수	처리 시간
3 회	350초

7) Module2-Results

3. 결과 분석

사용 알고리즘	평균 충돌 대기 횟수	평균 처리 시간
A*	6회	356.6초
Improved A*	2.33회	340 초

1. 서울 중소유통 물류센터에 필요한 최소 AGV수는 3대
2. AGV 수가 증가할수록 충돌대기횟수는 급격하게 증가
3. 모든 Random Order에서 **Makespan of Improved A* Algorithm < Makespan of A* Algorithm**

8) Module2-Conclusion

<결론>

- 3가지 Random Order 모두 Improved A* 모델이 A* 모델보다 makespan을 감소시킬 수 있었음
- AGV 수 증가할 수록 충돌 급격하게 증가하기 때문에 Improved A*의 충돌 방지 효과 상승

⇒ 3대 이상의 AGV 도입을 고려하는 물류센터의 경우 AGV Routing에 Improved A* 모델을 사용하는 것이 효과적

서울 중소유통 물류센터에 물류시스템 개선을 위해 AGV를 도입한다면,

최소 도입 AGV수가 3대이기 때문에 **Improved A* Algorithm을 사용하는 것이 효과적**일 수 있다.

<한계>

- 휴리스틱 기법을 사용, Improved A* Algorithm이 AGV 경로 최적화의 Global Optimal Solution인지는 판단 불가
- Short Term 분석을 진행하였기 때문에, AGV 도입의 사업성(Long Term) 분석 어려움

9) Reference

Yuan, Z. *et al.* (2020) "A bi-level path planning algorithm for multi-agv routing problem," *Electronics*, 9(9), p. 1351. Available at: https://doi.org/10.3390/electronics9091351.

Hasan, H.S. *et al.* (2019) "Automated Guided Vehicle Routing: Static, dynamic and free range," *International Journal of Engineering and Advanced Technology*, 8(5C), pp. 1–7. Available at: https://doi.org/10.35940/ijeat.e1001.0585c19.


THANK YOU