

실시간 요청 및 수요불균형을 고려한 차량 할당 및 재배치 온라인 알고리즘

2017. 06. 02

권보배, 우샤오룽, 정세윤, 하병현

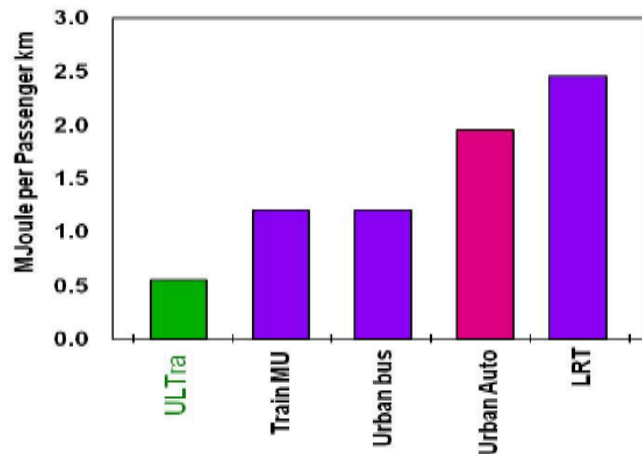
부산대학교 산업공학과

구성

1. 서론
2. 문제 정의
3. 접근방법
4. 알고리즘
5. 실험결과
6. 결론

1. 서론

- PRT (personal rapid transit)
 - 1~6명이 탑승 가능한 소형 무인 궤도 차량
 - 무인 자동운전으로 운행
 - 24시간 이용가능
 - 택시처럼 고객을 서비스함
 - 수요응답형
 - 출발지에서 목적지까지 무정차운행
 - 친환경 대중교통수단
 - 전기 동력 사용
 - 차량 자체가 가벼워 에너지 효율이 높음



Km당 에너지 사용량

Source: USA PRT, Ultra fairwood green transport

1. 서론

- PRT 운영 환경

- 승객은 동적으로 도착
- 실시간 정보를 이용한 중앙 집중식 제어
 - 예: PRT 위치, 역의 고객 대기 현황 등

- PRT 운영을 위한 의사결정 사항

- 배차(dispatching)
 - 동적으로 도착하는 승객의 즉각적인 운송 서비스 요청에 대응
- 재배치(relocation)
 - 공차가 된 차량을 특정 위치로 미리 배치시켜 차량 부족문제를 해결하고 향후 도착할 고객에 미리 대응
- 계획수정
 - 승객이 도착하는 등 새로운 정보가 발생할 시 확정된 배차/재배치 계획을 수정

Source: ULTra PRT sustainable transit

- PRT 시스템 운영은 동적으로 도착하는 승객을 차량에 배차하여 운송 서비스를 하는 것으로 동적수송문제(dynamic pick-up and delivery problem)이다.

1. 서론

- PRT 관련연구

- 배차와 재배치

- Ingmar (2003): 탑승 요청이 발생하면 요청지로부터 가까운 거리에 있는 공차를 배차하고, 과거 이력 데이터를 통계적으로 분석하여 공차를 미리 정류장에 배치
 - Lees-Miller (2011): 차량 배차 및 재배치를 동시에 하는 SV (sampling and voting) 알고리즘을 제시
 - Fatnassi and Siala (2013): 에너지 소비를 고려한 배차와 재배치 전략을 제시

- ❖ 한계점

- ✓ 계획수정은 고려되지 않음
 - ✓ 공차 이동중인 차량은 배차계획 시 고려되지 않음

- 배차와 계획수정

- Han et al. (2014): 차량의 상태를 고려하여 계획수정을 허용하는 배차 방법론을 제시
 - Han et al. (2015): 기존 연구를 확장하여 차량의 정체를 고려한 차량경로계획 방법론도 함께 제안

- ❖ 한계점

- ✓ 공차가 발생하는 위치는 일정하지 않고 수요가 발생하는 위치 또한 일정하지 않음
 - ✓ 승객이 발생한 후 배차(계획수정)를 할 경우 요청에 빠르게 대응하기 어려움

1. 서론

- 기여
 - **배차, 재배치 그리고 계획수정을 동시에 하는 방법론 제시**
 - 일반적인 배차계획의 의사결정과는 다른 **새로운 의사결정 변수 제안**
 - 시나리오 기반 **의사결정 대안 평가 방법** 제안
 - ※ 불확실한 상황에서 안정적으로 적용 가능
- 성능 확인
 - 시뮬레이션 기법을 사용
 - 성능 지표는 승객의 서비스수준을 평가할 수 있는 평균승객대기시간을 사용

2. 문제 정의

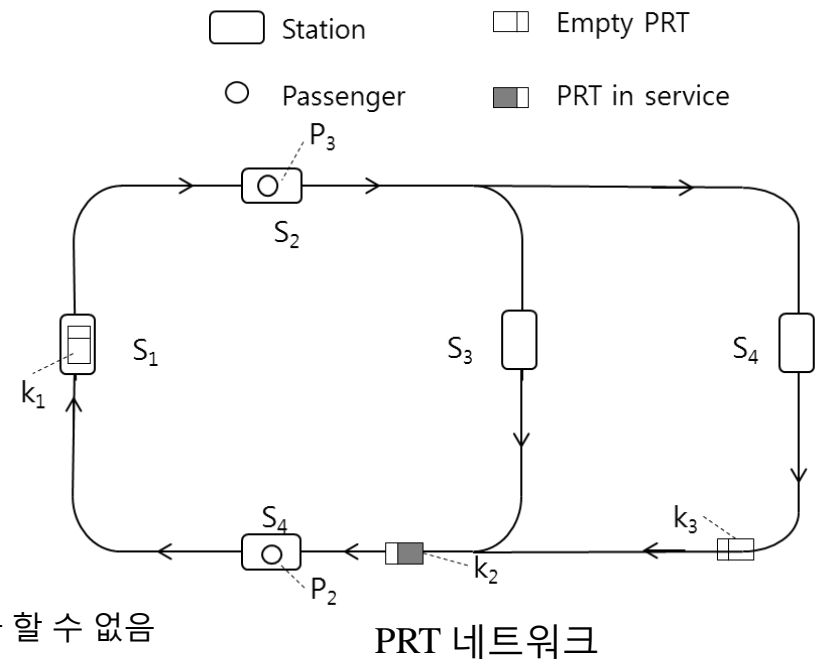
• PRT 시스템 구성요소 및 가정사항

▪ 네트워크(정류장, 선로)

- 정류장은 승객이 도착하여 운영 시스템에 차량 배차를 요청 후 대기하거나 PRT 차량에 승객이 탑승 혹은 하차하는 장소
 - 정류장에서 최대 정차 가능한 PRT 차량 수에 대한 제한은 없다고 가정함
- 단일방향(unidirectional)
- 정류장 i 에서 j 로 이동하는 시간 t_{ij} 는 일정함
 - 차량 간 간섭은 없다고 가정함
 - 정류장 간 이동 시간은 일정하다고 가정
 - 승객이 승/하차하는데 걸리는 시간은 없다고 가정함

▪ PRT

- 차량 종류는 동일하다고 가정함
- 주행 속도는 동일하다고 가정함
- 주행 경로는 항상 최단 경로를 사용한다고 가정함
- 공차 상태로 정류장에서 대기하고 선로 위에서는 정차 할 수 없음



2. 문제 정의

- PRT 시스템 구성요소 및 가정사항

- 승객

- 승객은 차량에 탑승하여 목적지를 입력한다고 가정함

- 승객이 도착하였을 경우 탑승 요청 시간 및 출발 장소는 알고 있지만 목적지는 알지 못함

- 정류장에 도착한 승객은 목적지에 도착하기 전까지 정류장을 떠나지 않는다고 가정함

- 승차 희망 시간 및 도착 희망 시간은 고려하지 않으므로 예약 승객은 없다고 가정함

- 합승 및 환승은 없다고 가정함

- 승객의 도착률(arrival rate) 및 출발지와 목적지의 분포는 알고있다고 가정함

- O-D 행렬의 각 요소는 출발지 i 에서 목적지 j 로 이동하는 승객의 도착률 λ_{ij} (명/단위 시간)를 의미함

O-D 행렬

Destination(j) Origin (i)	1	2	...	m
1	-	λ_{12}	:	λ_{1m}
2	λ_{21}	-	:	λ_{2m}
:	:	:	:	:
m	λ_{m1}	λ_{m2}	:	-

3. 접근방법

3.1 의사결정

- 배차 의사결정 범위

- 일반적인 배차계획: Nearest Neighbor (NN), Dynamic Nearest Neighbor (DNN) 등

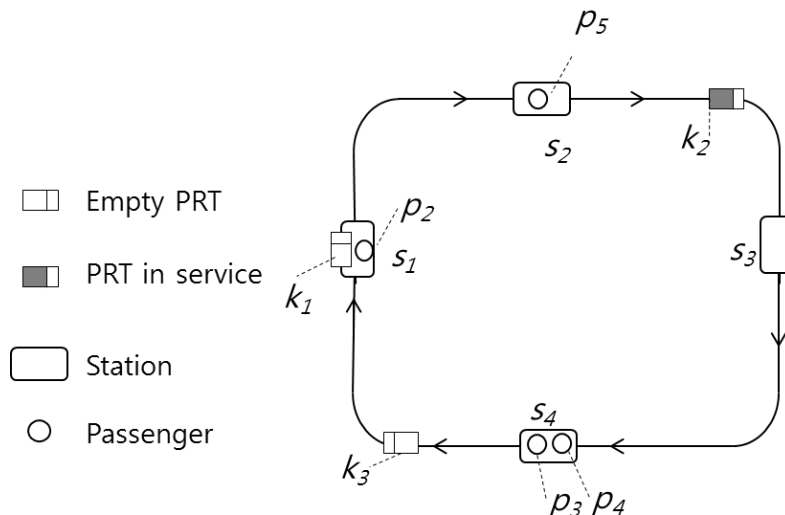
- 차량의 승객 운송 계획에 대한 차량 경로 혹은 승객 운송 서비스 순서를 지정

- 배차 계획 예시) $[k_1: s_1(p_2: \text{탑승}) - s_2 - s_3(p_2: \text{하차}) - s_4(p_4: \text{탑승}) \dots]$

- 일부 승객만 할당: Han et al. 2014

- 일부 승객만 차량에 할당하고 나머지 승객에 대해서는 의사결정을 미루는 방법론 제안

- 배차 계획 예시) $[k_1: s_1 - s_2(p_5: \text{탑승}) - s_3(p_5: \text{하차})]$



차량	탑승	배차
k_1		p_2, p_4
k_2	p_1	p_3
k_3		p_5

일반적인 배차 계획

차량	탑승	배차
k_1		p_5
k_2	p_1	
k_3		p_2

일부 승객만 배차

3. 접근방법

3.1 의사결정

- **장점**

- 불확실성이 높은 상황에서 **확실성이 높아질 때까지 의사결정을 지연시킴**
- 계획 수정을 빠르게 할 수 있음
- 일부 승객만 할당이 일반적인 배차계획을 하는 NN, DNN 보다 성능이 좋음

- **더 축소된 범위?**

- **가능한 최소 단위 의사결정**

- 정류장에 정차하거나 대기중인 차량에 대해서만 의사결정

- 기다리는 승객이 없는 곳에 정차하거나 대기중인 공차
 - 계속 대기할지
 - 다음 정류장으로 공차이동을 할지
 - 기다리는 승객이 있는 곳에서 정차한 공차
 - 승객을 태울지
 - 태우지 않고 다음 정류장으로 공차이동을 할지

3. 접근방법

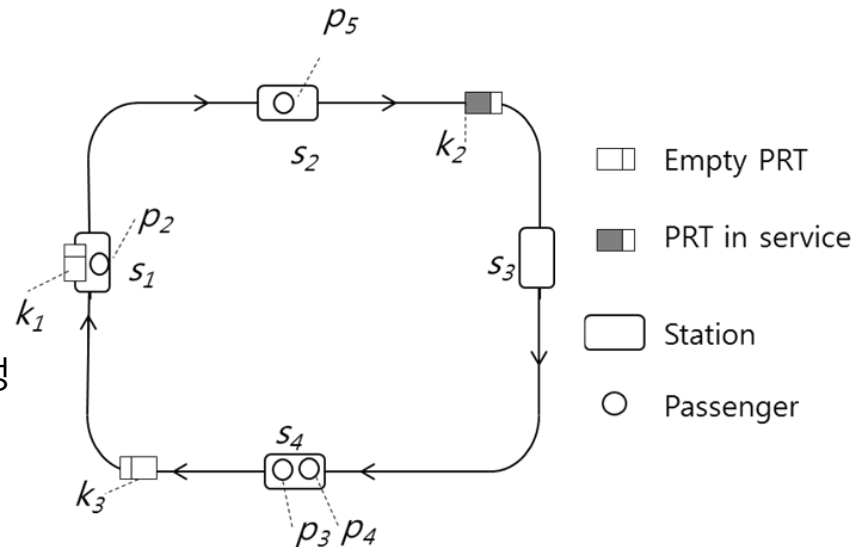
3.1 의사결정

- 대상
 - 정류장에 차량이 도착하거나, 현재 정류장에 대기중인 차량
- 최소 단위 의사결정
 - (대기) 차량이 정류장에 정차하였을 때 대기 승객이 없는 경우 승객을 기다릴지
 - (승객 탑승) 대기 승객이 있을 경우 대기 승객을 태우고 출발할지
 - (이동) 대기 승객이 있거나 없는 모든 경우 공차 상태로 정류장을 떠나 다음 어떤 정류장으로 출발 할 지

- 계획 시점
 - 새로운 승객이 왔을 때
 - 공차가 정류장에 도착했을 때
 - 승객이 목적지에 도착했을 때

최소 단위 의사결정

차량	의사결정
k_1	이동
k_2	-
k_3	-



3. 접근방법

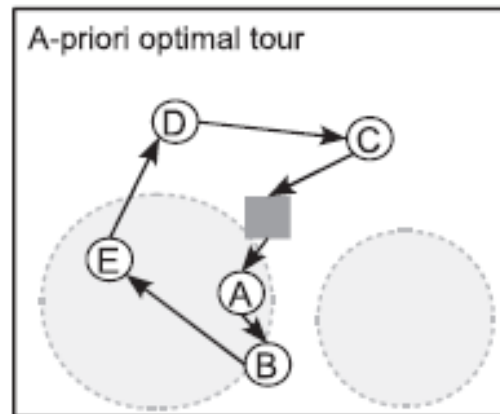
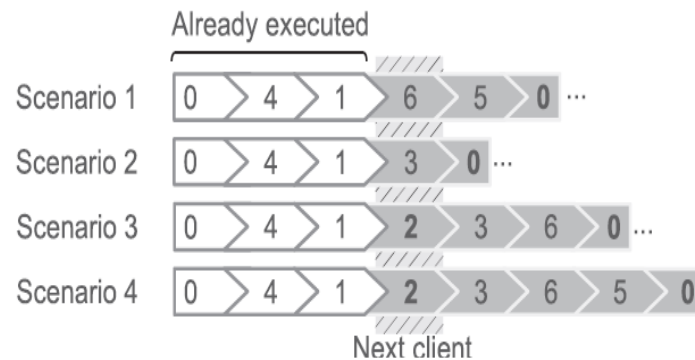
3.2 시나리오 생성

- 미래 승객 대비
 - 미래에 도착할 승객에 대응하여, 미리 차량을 승객이 도착할 위치로 배치
 - 일반적인 방법
 - 시나리오 생성
 - 미래 가상 수요를 N 번 생성하여 현재 수요와 합한 시나리오를 N 개 생성
 - 현재 대기 수요 + 가상 수요 = 시나리오
 - 생성된 시나리오를 바탕으로 의사결정을 계획함
 - 관련 연구: Bent and Hentenryck (2004), Hvattum et al. (2006), Ichoua (2006), Hvattum and Løkketangen (2007), Pillac et al. (2012)

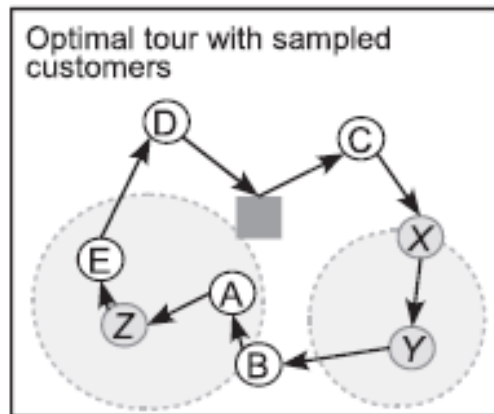
3. 접근방법

3.2 시나리오 생성

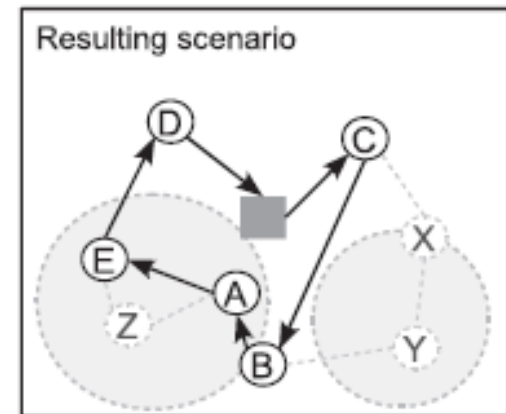
- 예시 (Pillac et al., 2012)



Legend: (A) (A) Customer (actual/sampled)



Depot → Tour



Customer distribution density

3. 접근방법

3.2 시나리오 생성

- 본 논문

- 각 승객 별 정보

- 고객 번호(현재, 가상), 고객 도착시간, 출발지 - 목적지

- 시나리오에서의 수요 구성

- 현재 대기 수요

- O-D 행렬을 이용하여 현재 대기 승객의 목적지 예측
 - 승객은 차량에 탑승하여 목적지를 입력한다고 가정함
 - 목적지 예측은 N 번 수행
 - C : 현재 도착해 있는 승객의 집합, (C_1, C_2, \dots, C_N)

- 가상 수요

- O-D 행렬을 이용하여 미래 가상 수요 생성
 - 승객의 도착률(arrival rate)은 알고 있다고 가정함
 - 승객이 도착하여 차량에 탑승하여 출발하는 출발지와 이동할 정류장인 목적지에 대한 분포는 알고 있다고 가정함
 - 가상 수요 생성은 N 번 수행
 - V : 가상으로 생성되는 승객의 집합, (V_1, V_2, \dots, V_N)



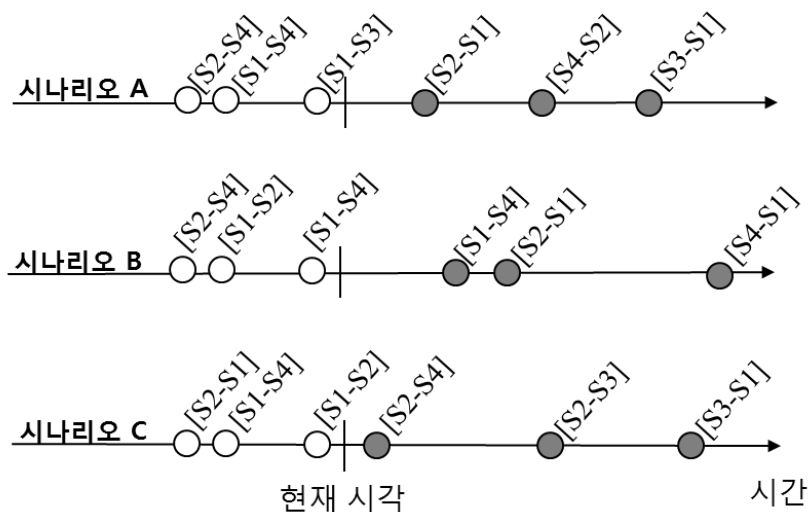
3. 접근방법

3.2 시나리오 생성

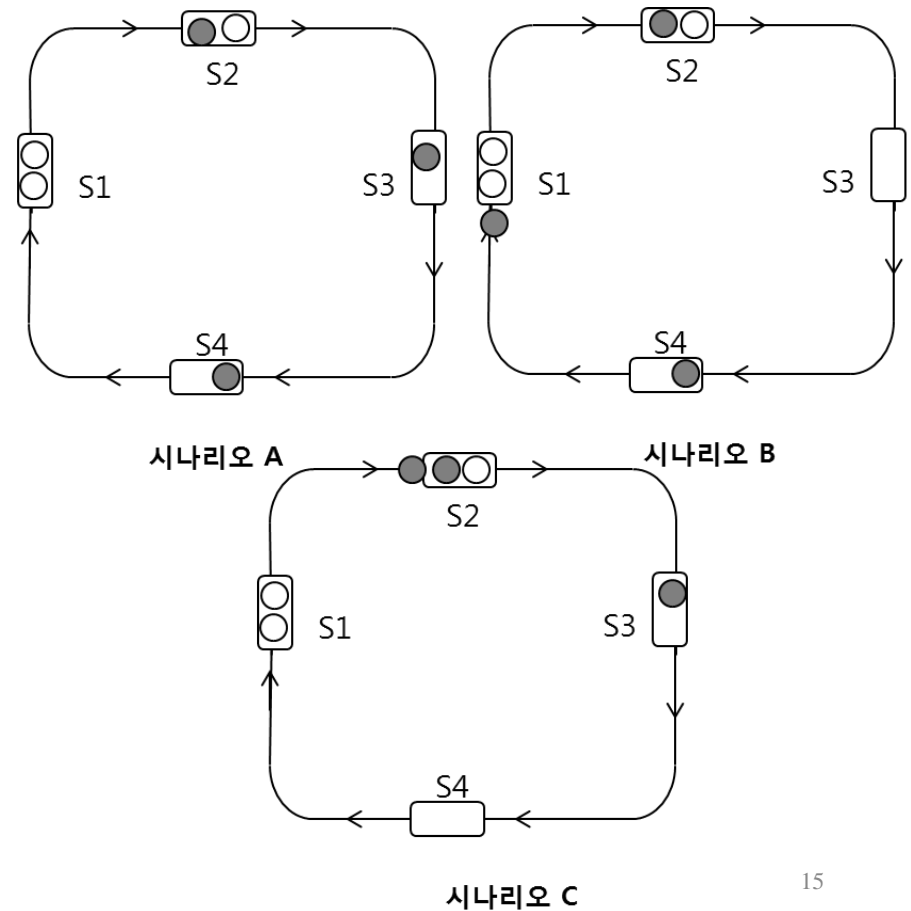
- 시나리오 생성

- 실제 대기 승객의 집합 C 와 가상 승객 집합 V 를 1:1로 매칭하여 시나리오 A 를 N 개 생성

- 예) i 번째 샘플링 된 각각의 C_i 와 V_i 를 합쳐 하나의 시나리오 A_i 을 생성



○ 대기 승객 ● 가상 승객



3. 접근방법

3.3 대안 생성 및 평가

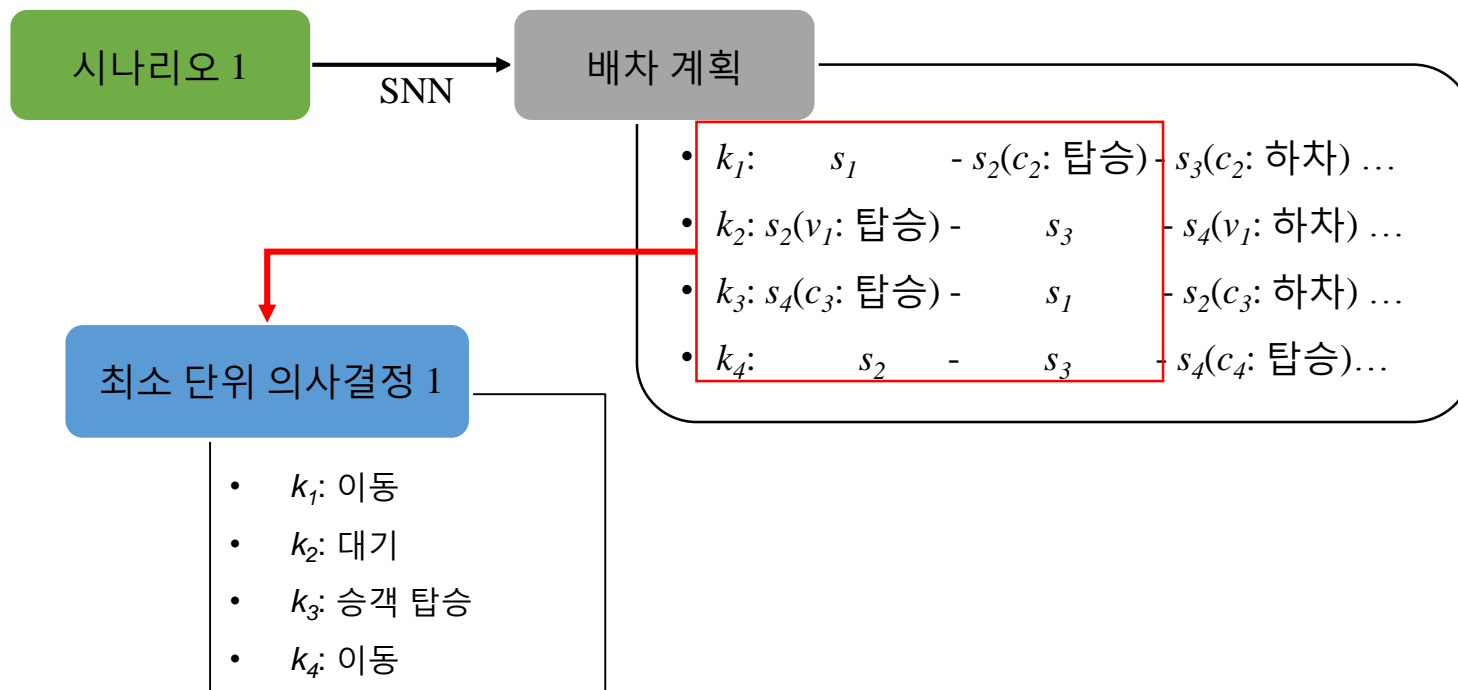
- 시나리오 사용
 - 계획 시점에 항상 재 생성하여 사용
 - 계획생성
 - 시나리오를 사용하여 최소 단위 의사결정 생성
 - 시나리오 기반 대안 평가
 - 최소 단위 의사결정을 적용하여 시나리오 확장

3. 접근방법

3.3 대안 생성 및 평가

- 계획생성

- 시나리오에서 SNN(static nearest neighbor)을 사용하여 배차계획 X_i 를 생성
- X_i 에서 대기, 승객 탑승, 이동이라는 의사결정을 추출하여 최소 단위 의사결정 Y_j 생성
- N개의 시나리오에서 생성될 최소 단위 의사결정 Y_j 의 총 개수는 M개 임 ($M \leq N$)



3. 접근방법

3.3 대안 생성 및 평가

- 계획생성

- 시나리오에서 SNN(static nearest neighbor)을 사용하여 배차계획 X_i 를 생성
- X_i 에서 대기, 승객 탑승, 이동이라는 의사결정을 추출하여 최소 단위 의사결정 Y_j 생성
- N개의 시나리오에서 생성될 최소 단위 의사결정 Y_j 의 총 개수는 M개 임 ($M \leq N$)



3. 접근방법

3.3 대안 생성 및 평가

- 시나리오 기반 대안 평가
 - 최소 단위 의사결정이 여러 상황에서 미래에 어떠한 영향을 미치는 지에 관한 확인

최소 단위 의사결정 1

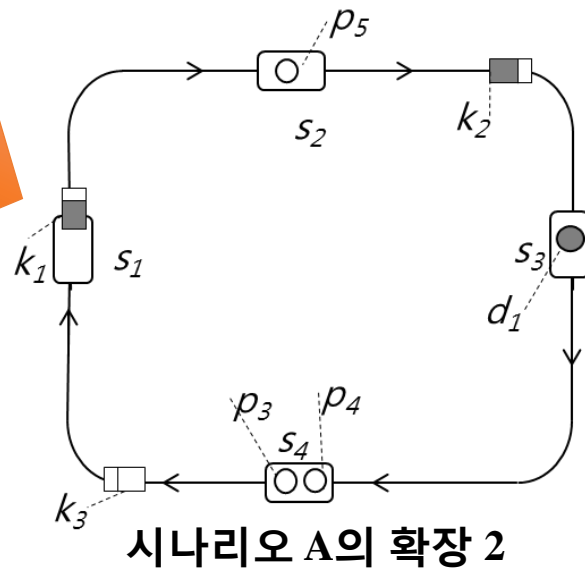
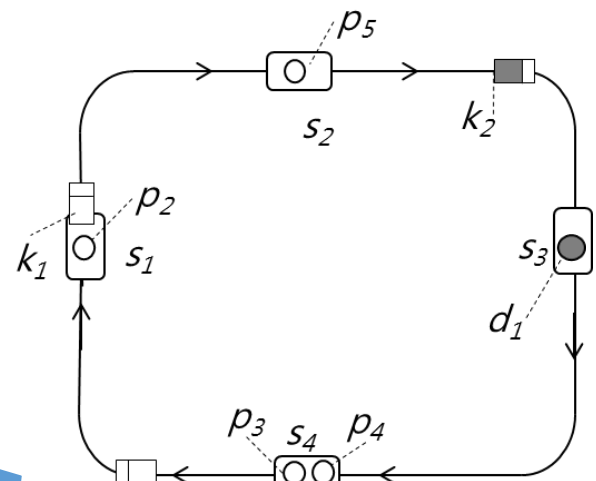
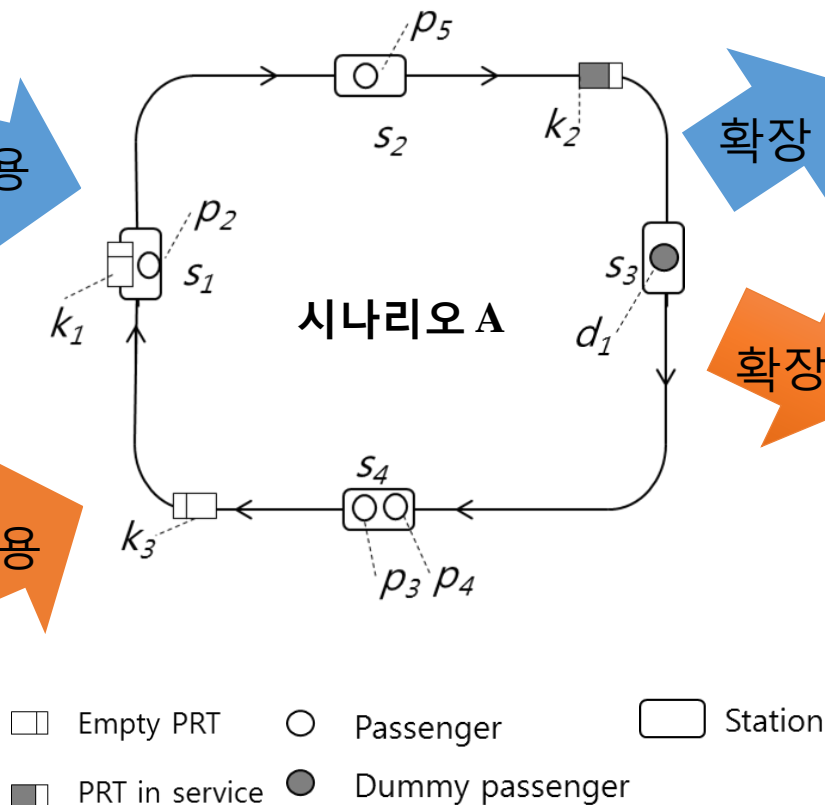
차량	의사결정
k_1	이동
K_2	-
k_3	-

적용

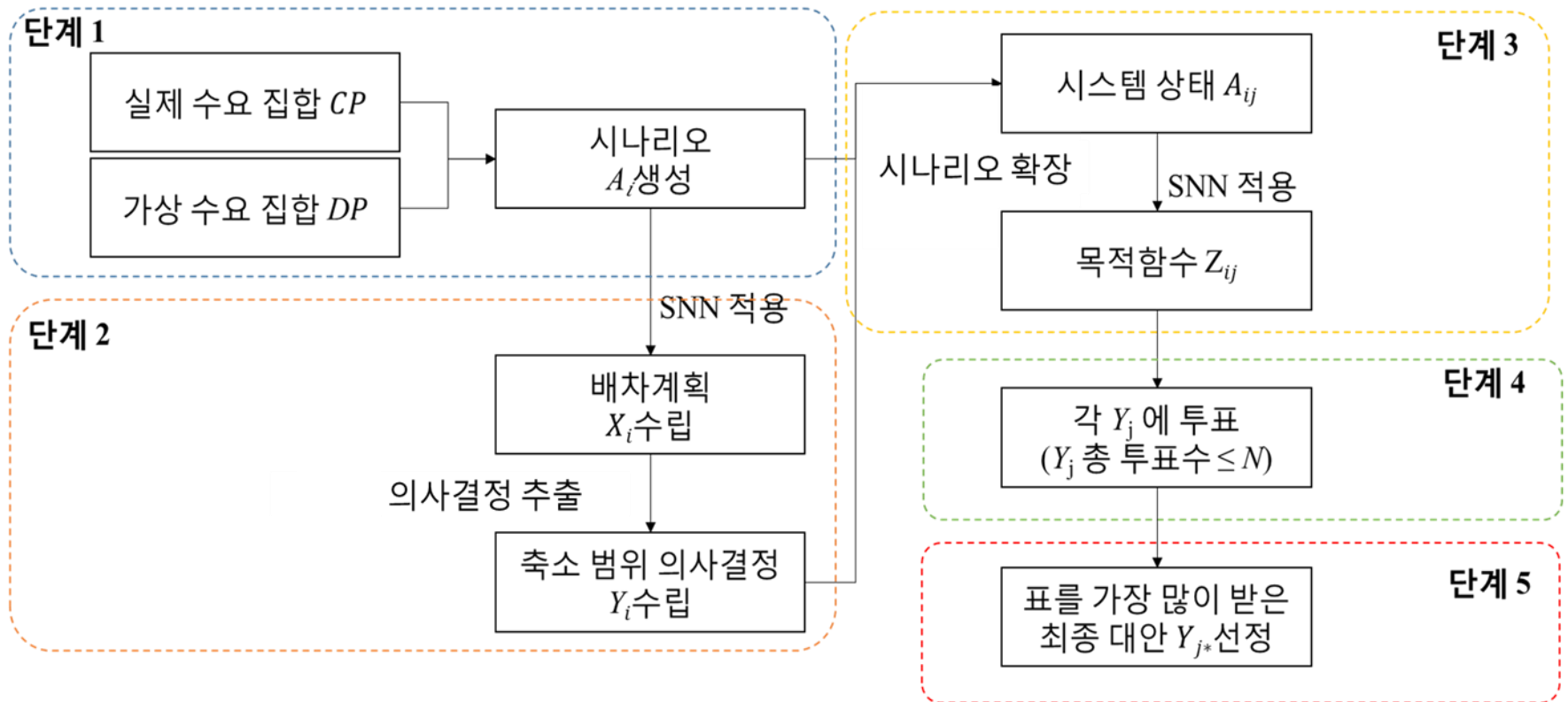
최소 단위 의사결정 2

차량	의사결정
k_1	승객 탑승
K_2	-
k_3	-

적용

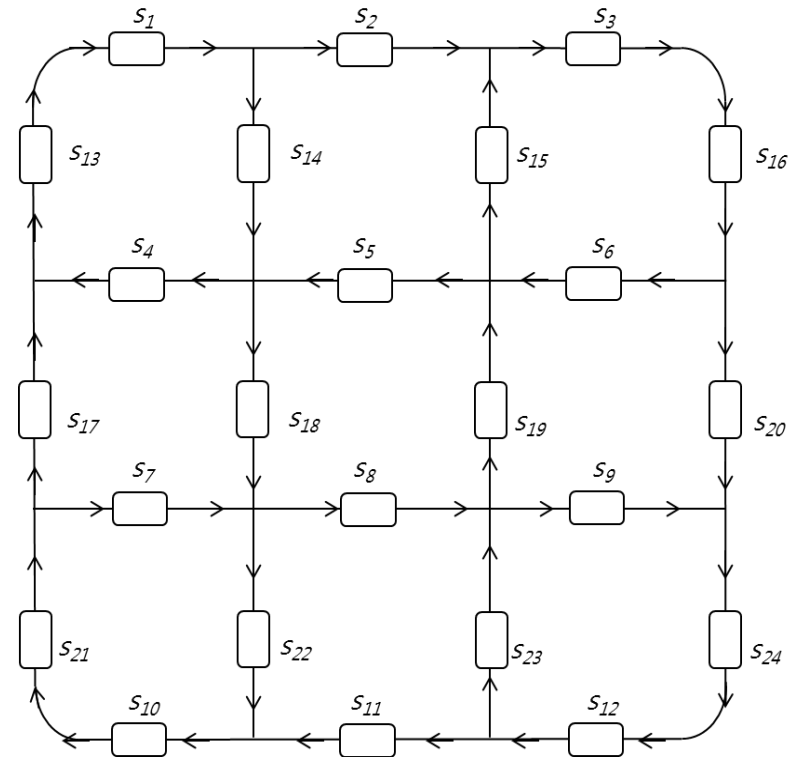


4. 알고리즘



5. 실험결과

- 실험 환경은 Lees-Miller (2011)와 동일(계속)
 - 승객 도착률은 포아송과정(Poisson Process)을 따름
 - 네트워크는 가상 grid 네트워크
 - 정류장은 총 24개
 - 각 정류장 사이의 거리는 800m
 - PRT 차량은 총 200대
 - 차량 전체의 주행 속도는 10m/s 로 동일함



Grid 네트워크

Source : Lees-Miller (2011)

5. 실험결과

- 실험 환경은 Lees-Miller (2011)와 동일
 - 승객이 각 정류장에 도착하는 비율과 이동하고자 하는 목적지 비율은 각각 다름
 - 승객의 흐름은 외곽에서 중앙으로의 이동이 가장 많이 발생
 - 중앙에서 외곽으로 이동하는 승객은 적게 발생

	5.0	5.0	5.0	
5.0		3.8	3.8	5.0
	3.8		2.5	3.8
5.0		2.5		5.0
	3.8		2.5	3.8
5.0		3.8		5.0
	5.0	5.0	5.0	

목적지 비율 (%)

	0.8	0.8	0.8	
0.8		3.8	3.8	0.8
	3.8		15.0	3.8
0.8		15.0		0.8
	3.8		15.0	3.8
0.8		3.8		0.8
	0.8	0.8	0.8	

도착지 비율 (%)

- 0.1 ~ 1.0까지의 강도에 대하여 실험
 - 고객도착비율은 강도(intensity)라고 함
 - 2035명/hour은 시스템의 **최대 허용 용량을 초과**하는 상태: 강도 1.0
 - 강도 별 도착승객수는 (강도 × 2035)명/hour로 계산됨

5. 실험결과

- 본 논문 설정 값
 - 실제 대기 수요와 가상 수요를 합친 시나리오 개수 N 은 50개
 - 시나리오에서 생성되는 가상승객수는 130명으로 함
 - 여러 번의 실험을 거쳐서 설정된 값
 - **가상수요가 너무 많이 생기면** 차량의 움직임이 많아져 선로 위에 이동중인 차량이 증가함
 - 정류장 도착 시 의사결정을 할 수 있는 차량의 수가 적음을 의미하고 적절한 배차 결정을 할 수 없게 됨
 - **가상수요가 충분하게 생성되지 않으면** 차량의 재배치가 안정적으로 이루어지지 않음
- 시뮬레이션 실험을 수행
 - 실험 결과는 20시간의 시뮬레이션 시간 중 시스템이 안정화된 상태의 30분 이후의 통계치를 수집
 - 이동경로는 최단경로를 사용

5. 실험결과

- 성능 비교 알고리즘

- BWNN (Bell and Wong Nearest Neighbors) (Wong and Bell, 2006)

- 동적인 상황에서 현재 도착한 고객만을 대상으로 차량을 승객에게 배차하는 알고리즘
 - 차량 배차 시 도착한 승객과 가장 가까운 위치에 있는 차량을 배차

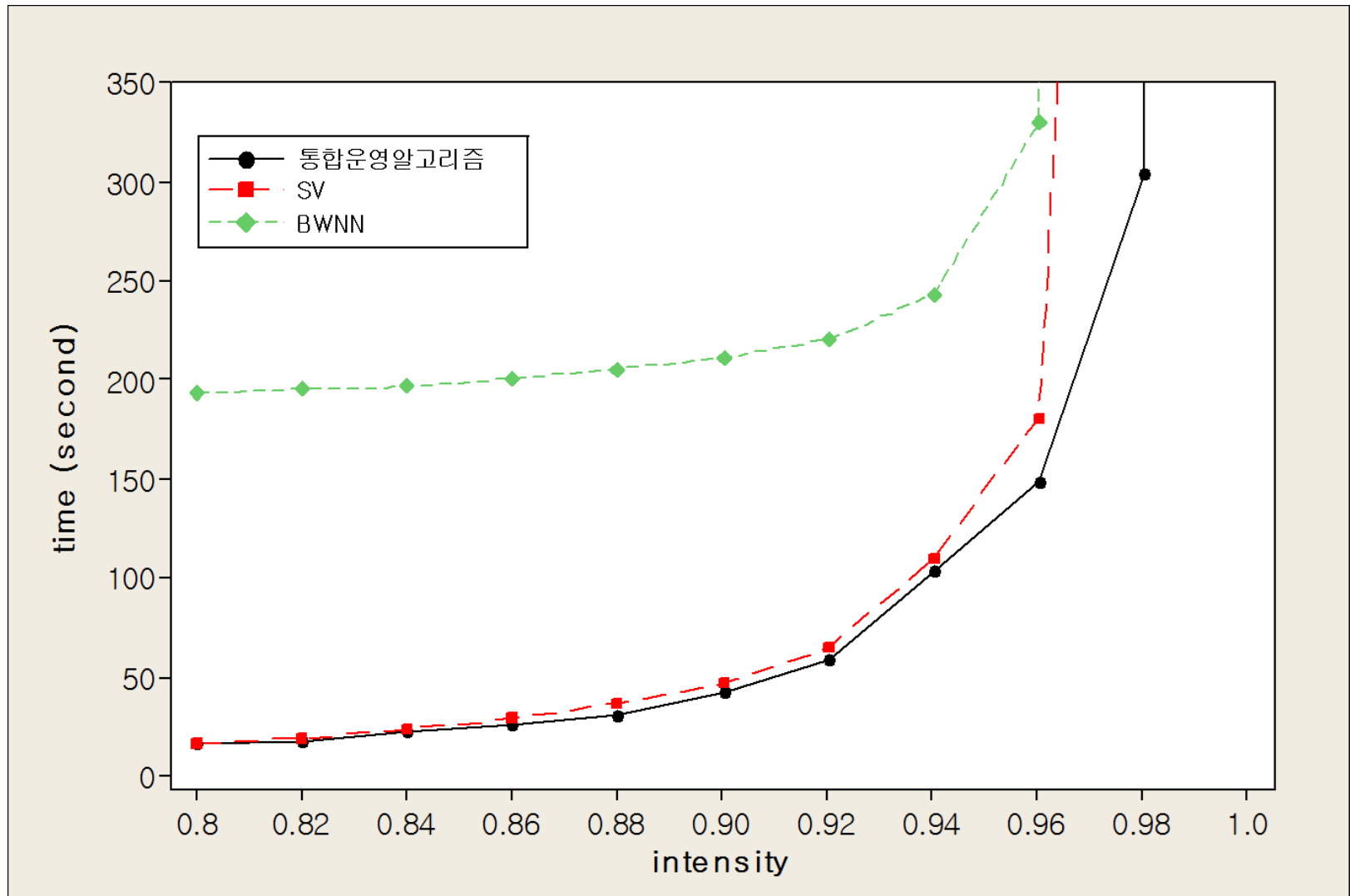
- SV 알고리즘 (Lees-Miller, 2011)

- 현재 승객을 배차하고 재배치하는 알고리즘
 - BWNN을 사용하여 현재 도착한 승객을 배차하고, 공차 발생 시 가상수요를 바탕으로 SNN을 사용하여 차량 재배치

- 차이점

- BWNN과 SV보다 더 불확실한 상황의 문제에 대한 방법론을 제시
 - BWNN과 SV는 도착한 승객의 **목적지는 이미 알고 있다고 가정**
 - 본 연구에서는 **목적지를 모른다고 가정**

5. 실험결과



5. 실험결과

Intensity	SV (초)	통합운영알고리즘(초)	개선율(%)
0.80	16.86	16.69	1.01
0.82	19.40	17.71	8.69
0.84	23.90	23.14	3.17
0.86	29.43	26.48	10.02
0.88	37.44	31.34	16.29
0.90	47.12	42.53	9.73
0.92	64.76	58.84	9.13
0.94	110.26	103.71	5.94
0.96	180.93	148.17	18.10
0.98	1067.09	304.38	71.48

6. 결론

- PRT 운영 문제에 관하여, 배차, 재배치 그리고 계획수정을 동시에 할 수 있는 통합운영 알고리즘을 제안
 - 결정 변수를 변화시킨 최소 단위 의사결정 방법론 제안
 - 시나리오 생성
 - 축소 단위 의사결정을 적용한 시나리오 확장
- 시뮬레이션 실험으로 제시한 알고리즘이 SV 보다 더 불확실한 상황에서 안정적으로 적용가능함을 보임
- 한계, 추후 연구
 - SNN 보다 제시한 알고리즘에 적합한 알고리즘 연구 필요
 - 순간 의사결정을 위한 배차계획과 미래상황 추정을 위한 배차계획을 실행할 시 동일하게 SNN을 사용
 - 실제 승객과 가상 승객의 가중치
 - 효과적인 시나리오 생성
 - 한번의 의사결정시 두 번의 배차계획을 해야 함
 - 전체 적으로 보면 알고리즘 시간이 늘어나게 됨



감사합니다.

Q & A

hajibin@pusan.ac.kr

SNN (static nearest neighbor)

- SNN 휴리스틱 (Lees-Miller, 2011)
 - 승객에 대한 배차 결과를 도출하는 것으로 해의 도출 시간이 짧고 승객의 대기시간을 줄이는 배차 계획을 도출

$$k^* = \operatorname{argmin}_k [\max\{0, a_k + t(d_k, O_r) - e_r\}] \quad (1)$$

- 승객: 대기중인 승객 $r \in R$
- k^* : 승객 r 에게 배차되는 차량
- 차량: $k \in K$
- d_k : 차량 k 에 할당된 마지막 수요의 종착지
- a_k : 차량 k 에 할당된 모든 수요를 처리한 후 d_k 에 도착하는 시간

