

세종특별자치시 출퇴근 버스 노선 제안

세종특별자치시 인구 밀집 지역 버스 노선 설계



요약 >>

요약 >>

요약 >>

01 아이디어 기획 배경 및 필요성

| 대중교통 위주 도시 설계 vs. 현재 상황

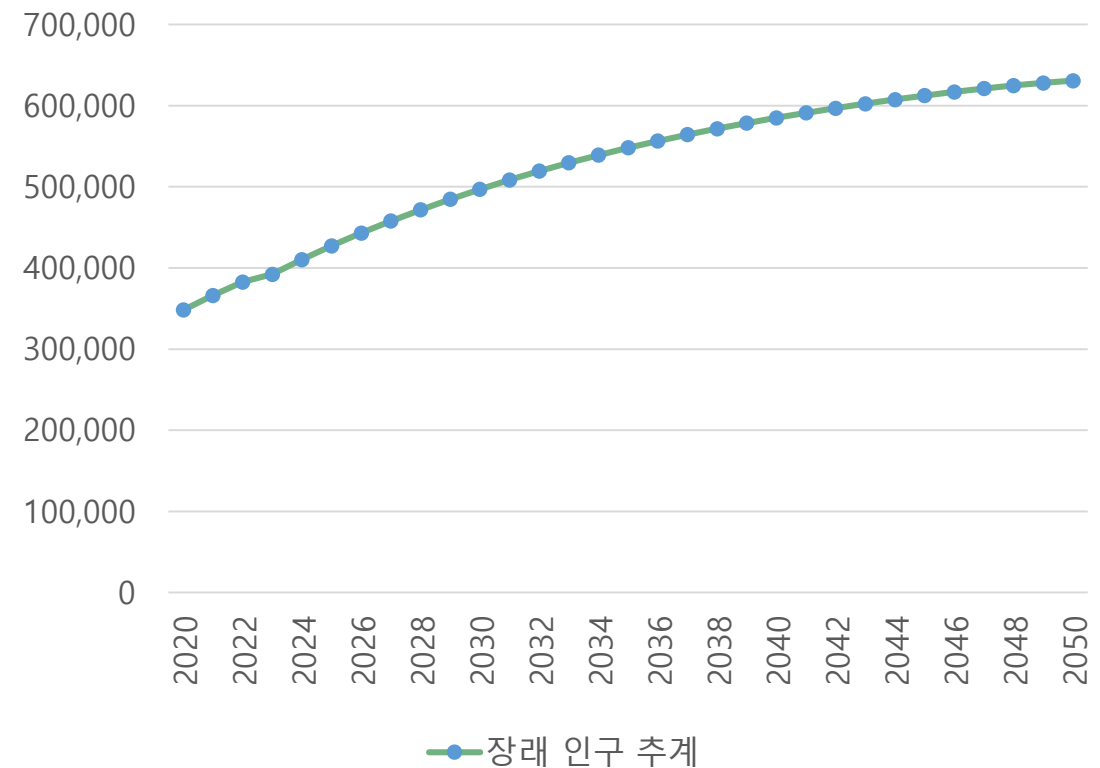


‘행복도시’의 기존 설계

대중교통 2링구조로, 자가용이 필요 없는 방식의 설계

- 도시 건설 때부터 일반도로의 차선을 줄임
- 간선도로 중앙에 BRT 전용차로 만들
- 자전거 도로와 인도 너비를 넓힘

장래 인구 추계



2050년까지 인구 60만명이 넘을 것으로 예상

01 아이디어 기획 배경 및 필요성

| 대중교통 위주 도시 설계 vs. 현재 상황



2012년 출범 당시 100,751명에서
2023년 현재 391,376명으로
인구가 **4배 가량** 늘어난 지금도
교통량을 감당하기 힘들

| 교차로명 | 구분 | 교통량 (대/시) | 평균제어지체 (초/대) | 서비스수준 (LOS) |
|-----------|----|--------------|-----------------|----------------|
| 합강교차로 | 4지 | 4,821 | 409.4 | FFF |
| 은하수교차로 | 4지 | 3,538 | 70.3 | E |
| 어진교차로 | 4지 | 3,403 | 64.7 | D |
| 세종교차로 | 4지 | 4,067 | 54.8 | D |
| 성금교차로 | 4지 | 3,806 | 54.3 | D |
| 파란달교차로 | 4지 | 3,501 | 51.8 | D |
| 새샘교차로 | 4지 | 4,381 | 50.6 | D |
| 가람교차로 | 4지 | 2,771 | 46.8 | C |
| 첫마을교차로 | 4지 | 2,309 | 38.3 | C |
| 청사교차로 | 4지 | 2,091 | 37.6 | C |
| 종합운동장 교차로 | 3지 | 3,283 | 36.4 | C |
| 햇무리교동측네거리 | 4지 | 3,127 | 28.5 | B |
| 나성4교남측삼거리 | 3지 | 2,233 | 21.6 | B |
| 보름교동측삼거리 | 3지 | 3,750 | 17.8 | B |

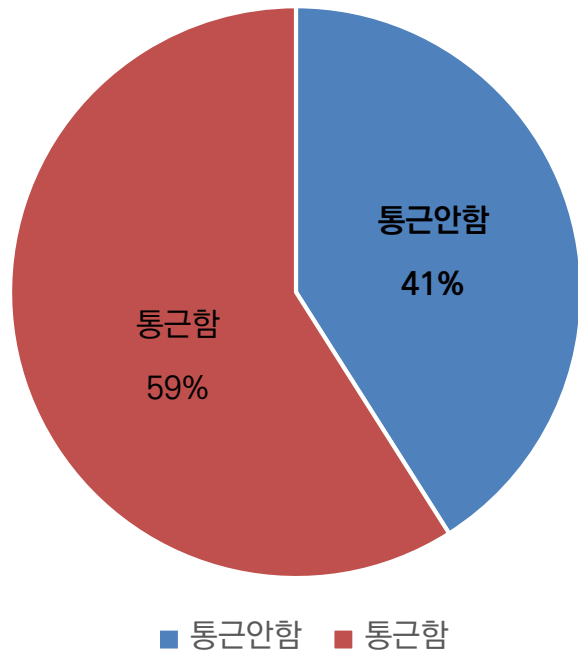
2020 행복도시 인근 교차로 서비스 수준

출처: 지난해 11월 기준 교차로 서비스 수준 분석 결과(세종시 제공)
디트NEWS24(<http://www.dtnews24.com>)
<https://www.sjpost.co.kr/news/articleView.html?idxno=55025>
2021년 세종시 내부 교통망 서비스 수준(LOS) 분석 현황

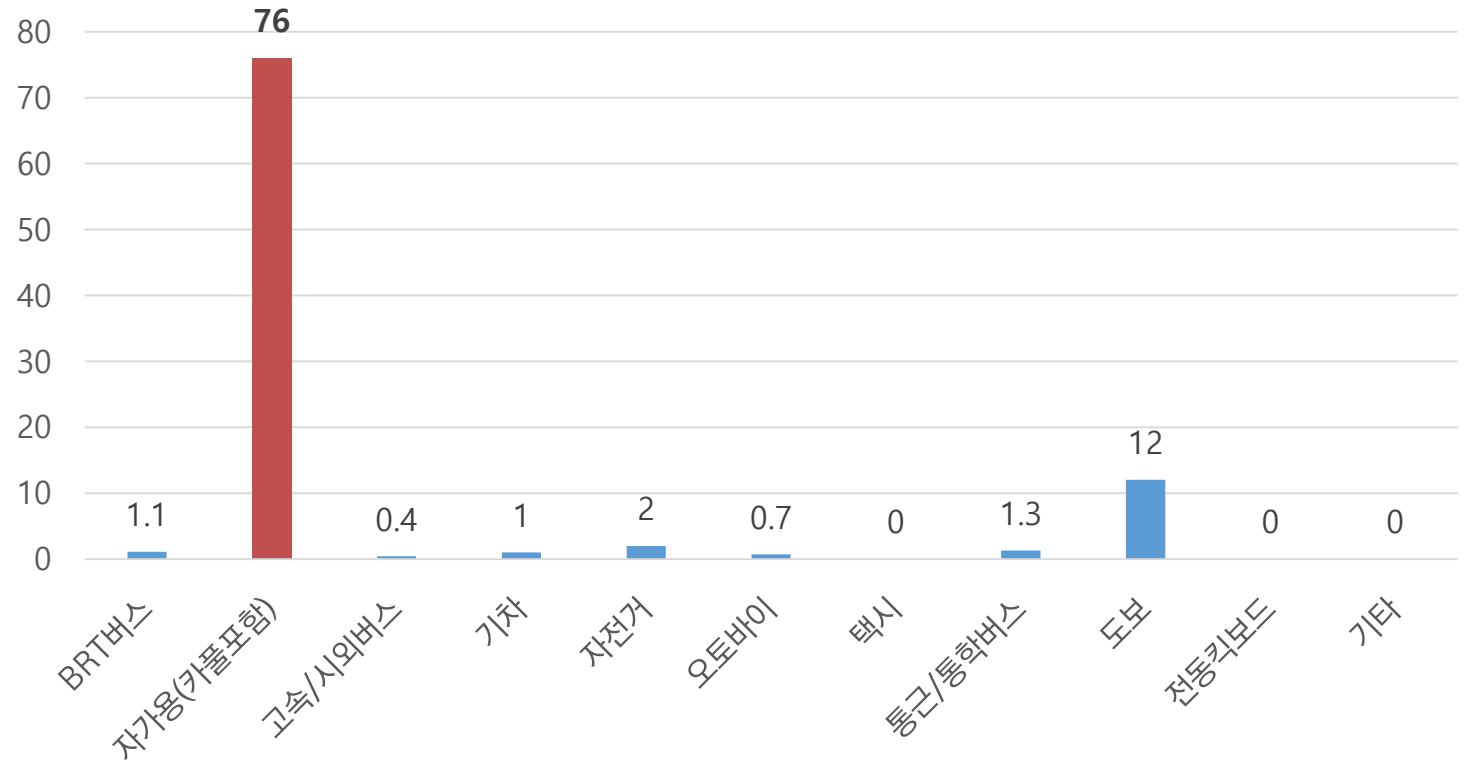
01 아이디어 기획 배경 및 필요성

| '대중교통 중심 도시'라는 초기 계획이 무색하게, 지속되는 자가용 이용 증가 및 대중교통 이용률 감소

21년 세종시민 통근 비율



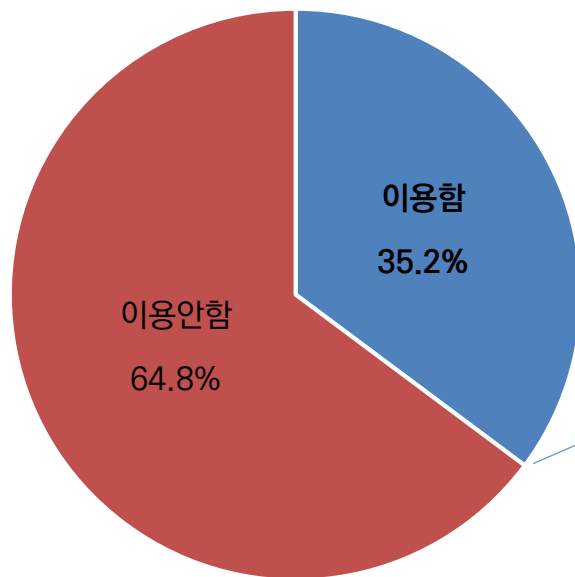
이용하는 주요 교통수단



01 아이디어 기획 배경 및 필요성

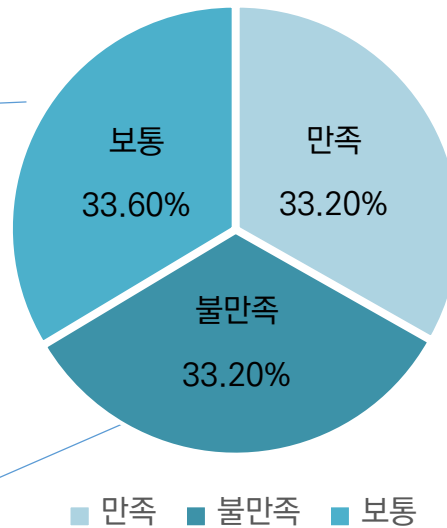
| 대중교통 문제점 - 1. 배차간격이 김

일반 시내버스/마을버스



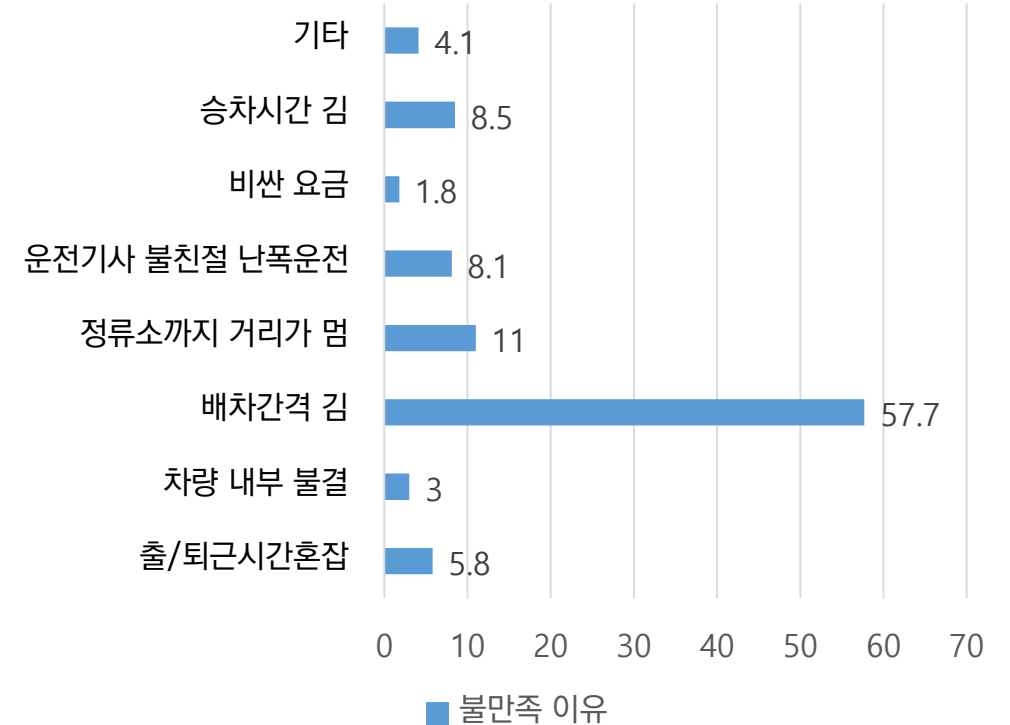
■ 이용함 ■ 이용안함

이용 만족도



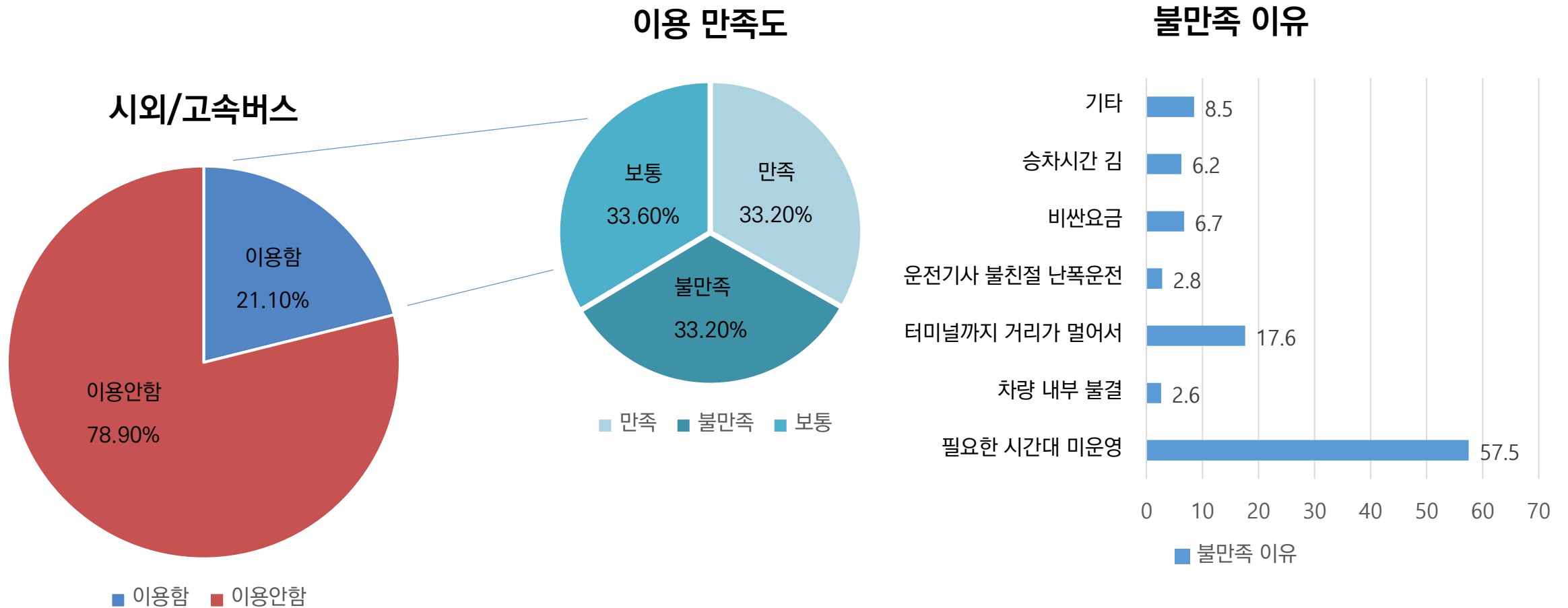
■ 만족 ■ 불만족 ■ 보통

불만족 이유



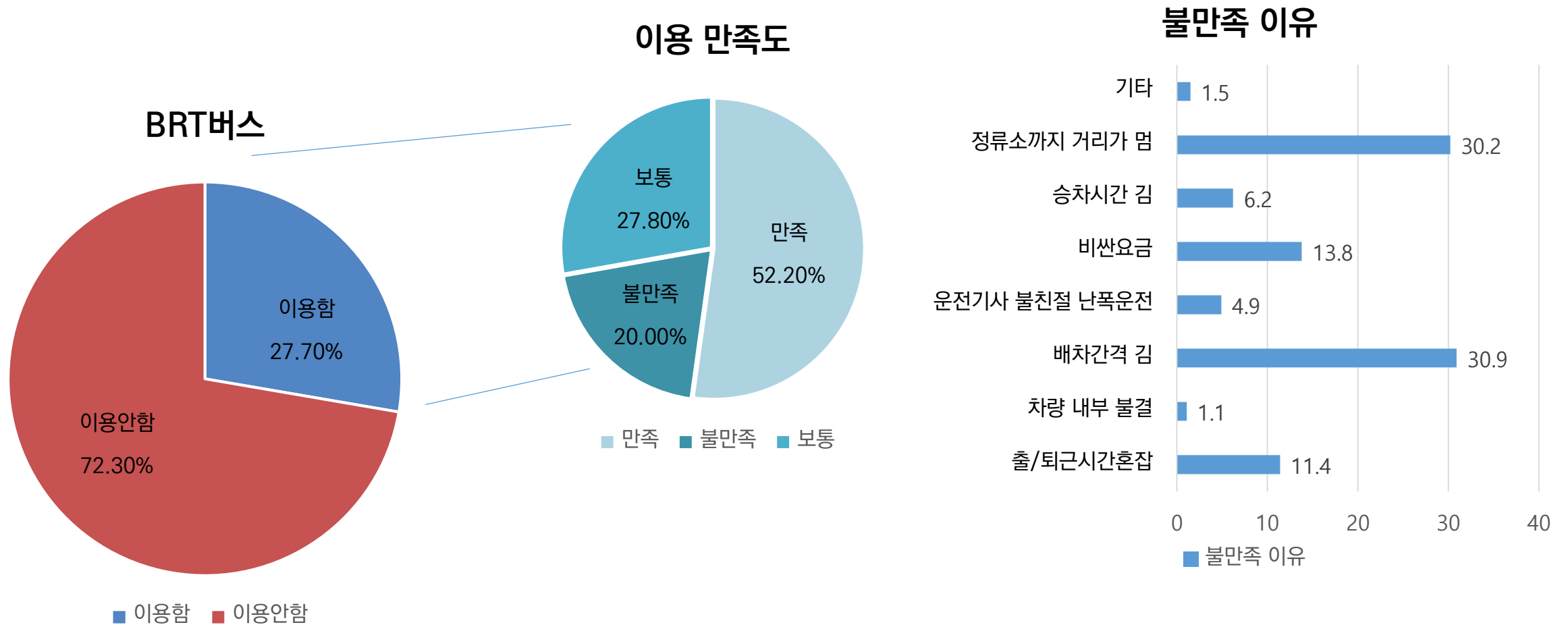
01 아이디어 기획 배경 및 필요성

| 대중교통 문제점 - 2. 필요한 시간대 미운영



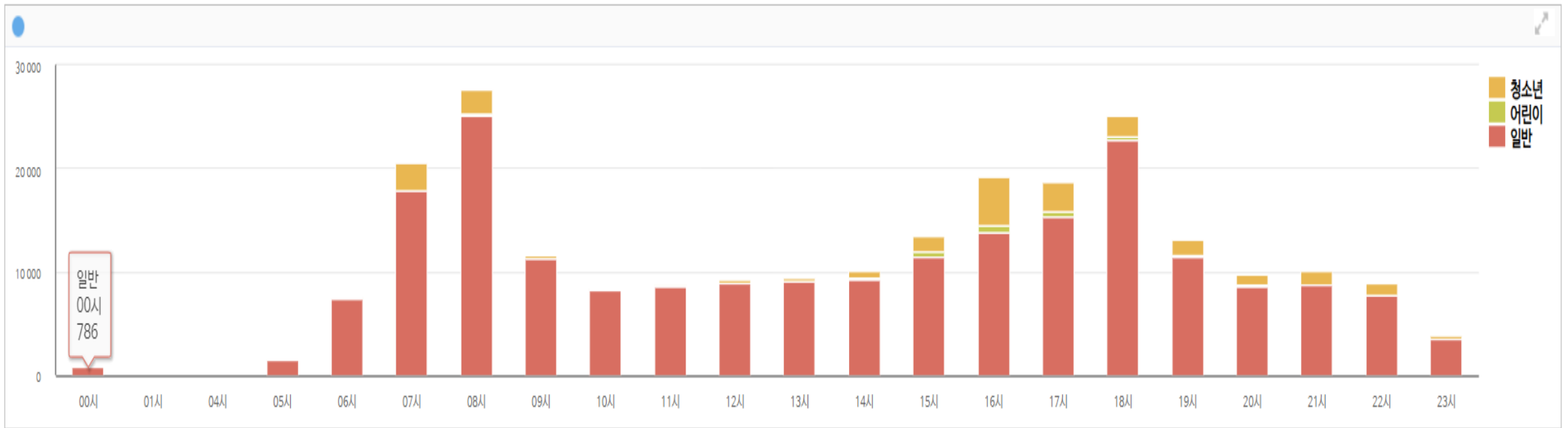
01 아이디어 기획 배경 및 필요성

| 대중교통 문제점 - 3. 필요한 위치에 정류장 부재



01 아이디어 기획 배경 및 필요성

| 출퇴근 시간에 발생하는 혼잡 상황



출근/퇴근 시간인 8시와 18시 부근에 가장 많은 인원이 몰리는 것을 확인할 수 있음.

01 아이디어 기획 배경 및 필요성

세종시 대중교통의 문제점

뒷받침되지 못하는
세종시의 좁은 도로 상태



배차 간격

출퇴근 시간 혼잡

운행 시간

정류장 위치



교통 복지 강화 시급

- 세종시 주민들의 편의 증대
- 대중교통 이용 활성화



세종시
인구 밀집 지역
출퇴근 버스
노선 설계

| 행정구역 | 인구수 |
|------|--------|
| 조치원읍 | 41,529 |
| 고운동 | 35,172 |
| 다정동 | 28,550 |
| 종촌동 | 28,381 |
| 반곡동 | 28,139 |
| 새롬동 | 26,560 |
| 도담동 | 25,419 |
| 아름동 | 23,583 |
| 소담동 | 21,850 |
| 보람동 | 18,986 |
| 한솔동 | 18,370 |
| 나성동 | 13,354 |
| 대평동 | 11,262 |
| 어진동 | 11,211 |
| 해밀동 | 8,953 |

세종시 내 인구 밀집 지역

01 아이디어 기획 배경 및 필요성

| 무료화로는 개선될 수 없는 대중교통 문제점

기대효과

- 대중교통 수송분담률 상승
- 탄소배출 저감 효과

대중교통
수송분담률
70% 달성

선순환
경제도시
실현

탄소배출
저감 효과

모두 어디든 갈 수 있는
편한 세종



교통 약자,
교통 소외지역
주민 모두 대중교통을
보다 더 쉽게 이용

교통비를 여민전으로
돌려주는
같이 잘사는 세종



지역화폐 이용으로
지역 경제는 더 활발하게
소득재분배 효과와
SOC투자비
절감 효과까지!

아름다운 정원도시
깨끗한 세종



교통 혼잡도 줄이고
미세먼지, 탄소배출도
줄이고~



**세종특별자치시,
2025년부터 시내버스 요금 전면 '무료'**

: 무료화 정책을 통한 대중교통 이용률 증가 기대

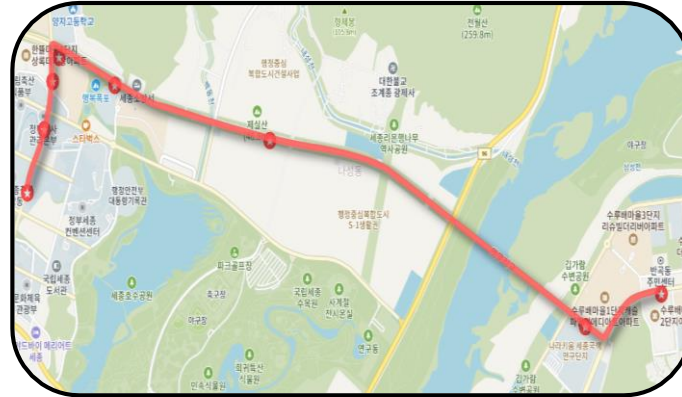
→ 근본적인 세종특별자치시 내 대중교통의
문제점은 개선 불가능하다고 판단

아이디어 제안

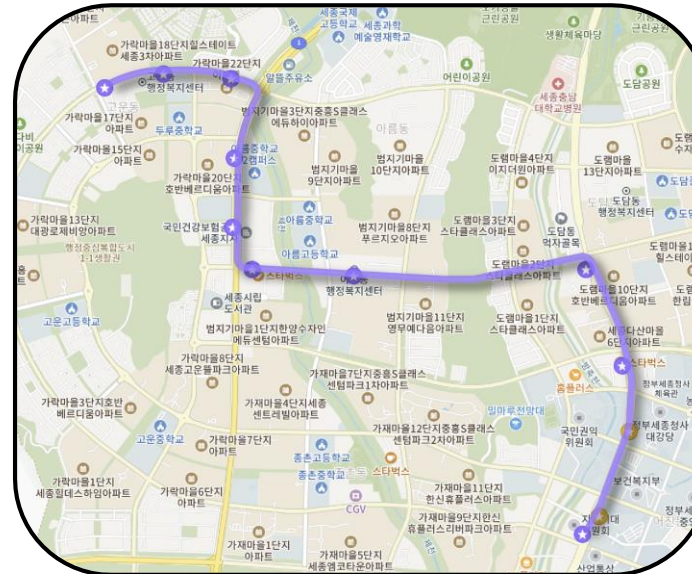


세종 ELT '얼른타' 버스

- 출퇴근 시간 위주 운영
(7:00-10:00 , 17:00-20:00)
- 버스 노선 4-5km 이내 구간 왕복
- 인구 밀집 지역 및 대중교통 서비스 불편 지역 위주로 노선 설정



노선 1



노선 3

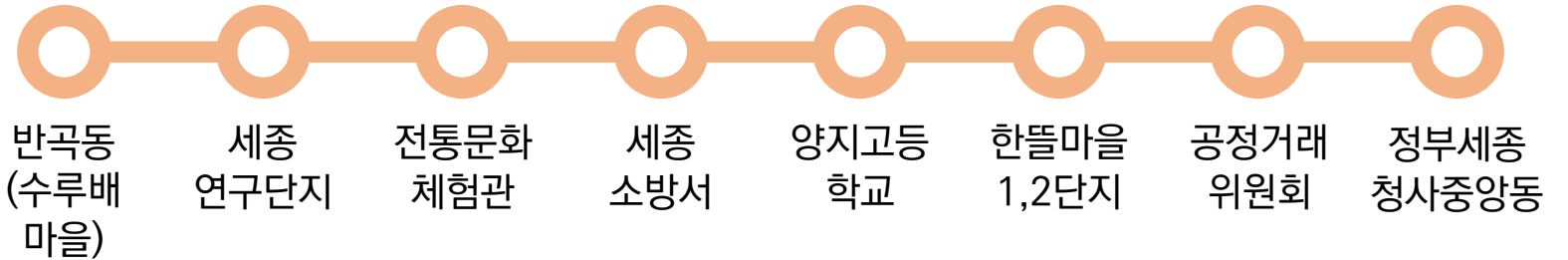


노선 2

아이디어 제안

세종 ELT(얼른타) 버스 노선 제안

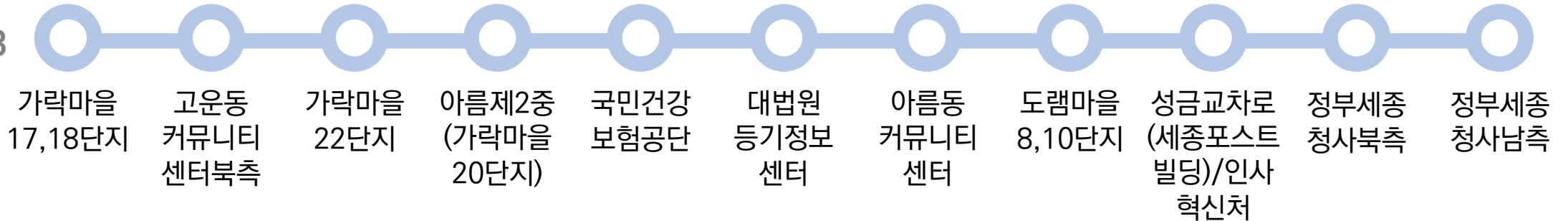
노선 1



노선 2



노선 3



02 기존 서비스와의 차별성 및 독창성

| 기존 세종특별자치시 버스 종류



두루타 버스

이용대상은 주로 고령층으로, 주민이동편의를 위해 생긴 콜버스 (면지역 위주 운행)
앱으로 실시간 예약 시 "대중교통 이용이 불편한 지역"에서도 쉽게 이용 가능

면지역 위주 운행 → 행복도시 + 조치원 시민은 출퇴근용으로 이용 불가

셔클

정해진 노선에 따라 다니는 버스 X = 합승해도 되는 택시
수요응답형으로 최적의 운행거리를 따져 호출한 이용객을 태우러 감



구독형 요금제로 꽤 비싸고, 오히려 사용자가 증가하면서 대기시간이 길어짐 → 출퇴근용으로 부적합함

02 기존 서비스와의 차별성 및 독창성

| 기존 세종특별자치시 버스 종류

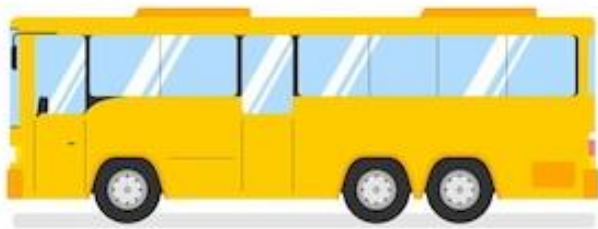


BRT 바로타 버스

세종시와 주변 도시를 연결하는 광역간선 급행버스
주변 도시로 출퇴근 하는 사람들에게는 적합한 버스

세종시 내 출퇴근용으로는 돌아가는 노선과 긴 배차간격을 지니고 있어 부적합

신설되는 ELT 얼론타 버스



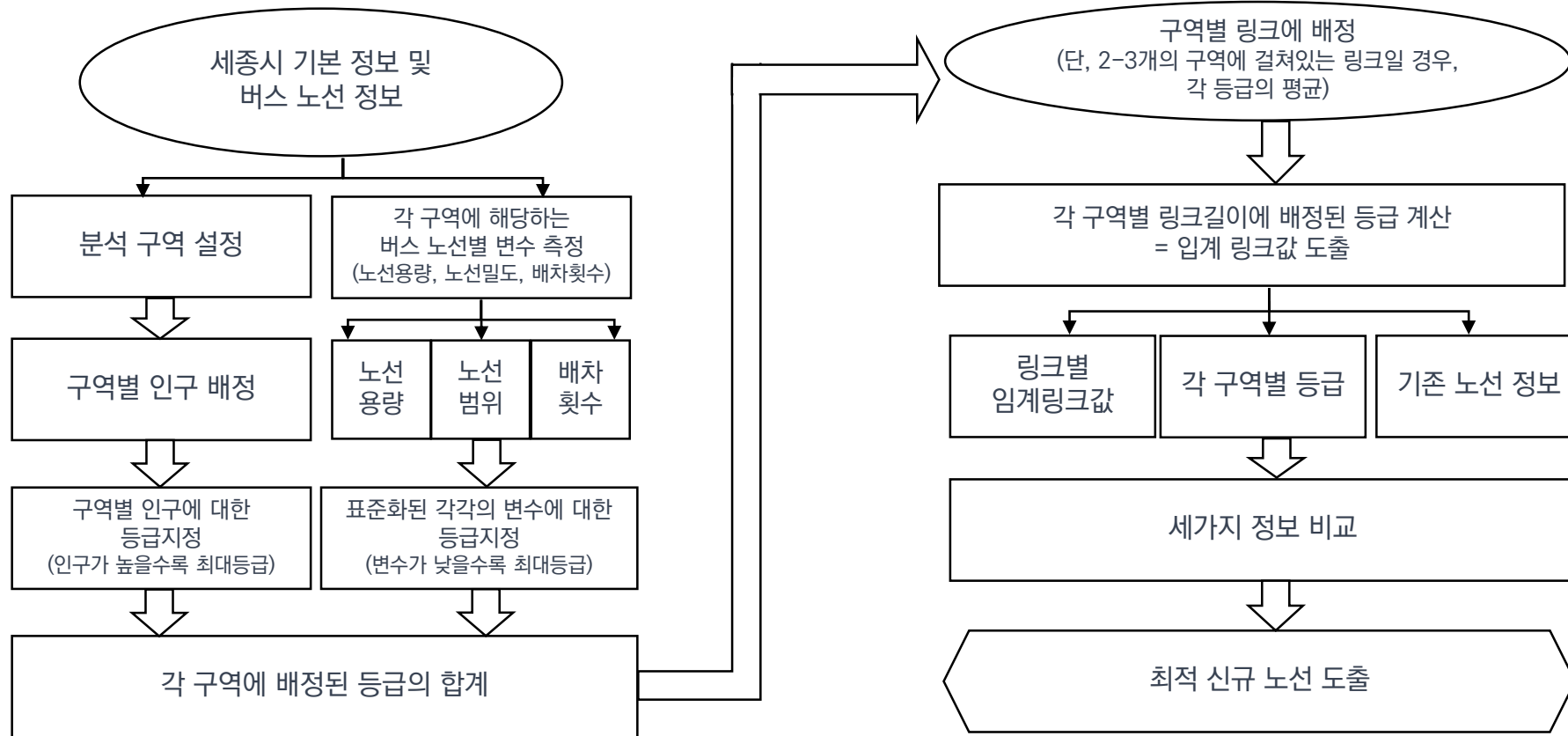
차별성

- ✓ 배차 간격 단축
- ✓ 출퇴근 시간 위주 운행
- ✓ 짧은 노선 길이
- ✓ 밀집 인구 지역 운행

독창성

- ✓ 노선 유연성
: 수요에 따라 노선 조정
- ✓ 마일리지 혜택 도입
- ✓ 친환경 차량 도입

03 아이디어의 실현 가능성 및 완성도



- 구역별 사회 경제 지표와 버스 노선 운행 특성 등을 고려한 합리적 노선 개편 방법
- 노선 투입의 우선순위를 정하는 방법론으로 활용 가능

데이터 분석 과정

- 목적 함수(Objective Function)

Min{Travel Time}
: Minimize Travel Time
⇒ 최단시간경로

Max{Pop, Density}
: Maximize Trip Generation Coverage
⇒ 통행인구밀집지역을 운행구역으로 최대한 포함

Min{Route Overlap}
: Serve the underserved area
⇒ 노선중복(overlapping/duplication)을 최소화

- 버스 용량(Bus Capacity)

– 인구 대비 운행 노선 버스 좌석수를 의미

$$CA = \frac{BS \times L}{Pop}$$

CA : Capacity score

L : Route length

Pop : Population

BS : Total daily bus seats

현재의 버스 공급 정책이
지역주민의 인구에 비하여
얼마나 제공되는가를 알 수 있는 지표

| 구역 | CA |
|------|----------|
| 조치원읍 | 16.17996 |
| 고운동 | 14.12645 |
| 다정동 | 15.33833 |
| 종촌동 | 18.7792 |
| 반곡동 | 33.98219 |
| 새롬동 | 12.83615 |
| 도담동 | 84.44475 |
| 아름동 | 35.25287 |
| 소담동 | 72.40523 |
| 보람동 | 75.82536 |
| 한솔동 | 29.23483 |
| 나성동 | 182.0821 |
| 대평동 | 185.3579 |
| 어진동 | 213.6264 |
| 해밀동 | 206.3086 |

해당 구역은 인구 밀집 지역 순으로 산출

데이터 분석 과정

• 버스의 공간적 분포(Bus Coverage)

- 단위 면적 당 분포되어 있는 버스 정류장의 수

$$CO = \frac{S}{A}$$

CO: Coverage score

S : Number of bus stops

A : Area

각 구역별 버스 서비스의
접근성 정도를 알 수 있는 지표

| 구역 | CO |
|------|----------|
| 조치원읍 | 7.890855 |
| 고운동 | 9.158879 |
| 다정동 | 12.35294 |
| 종촌동 | 19.13043 |
| 반곡동 | 11.25 |
| 새롬동 | 14.86486 |
| 도담동 | 12.2549 |
| 아름동 | 10.50228 |
| 소담동 | 20.51282 |
| 보람동 | 13.53383 |
| 한솔동 | 4.814815 |
| 나성동 | 15.29412 |
| 대평동 | 13.15789 |
| 어진동 | 16.60377 |
| 해밀동 | 7.692308 |

• 버스 운행 횟수(Bus Route Frequency)

- 하루동안 구역 내 버스 정류장을 통과한 총 버스 대수

$$F = TB$$

F : Frequency score

TB : Total number of buses

배차간격과 관련 있는 지표로서
버스 대기시간 정도를 알 수 있는 지표

| 구역 | F |
|------|------|
| 조치원읍 | 485 |
| 고운동 | 384 |
| 다정동 | 393 |
| 종촌동 | 384 |
| 반곡동 | 544 |
| 새롬동 | 321 |
| 도담동 | 871 |
| 아름동 | 571 |
| 소담동 | 629 |
| 보람동 | 569 |
| 한솔동 | 419 |
| 나성동 | 986 |
| 대평동 | 992 |
| 어진동 | 1087 |
| 해밀동 | 571 |

데이터 분석 과정

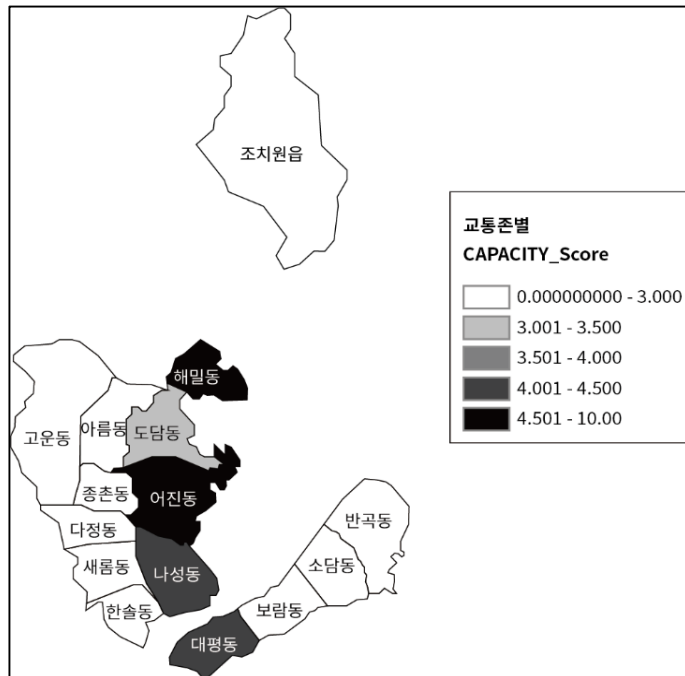
• 지역 대중교통 이용도 지수(LITA: Local Index of Transit Availability)

현재 버스서비스수준(버스용량, 버스정류장밀도, 운행횟수)과 버스이용도를 결합한 서비스 이용도 지수

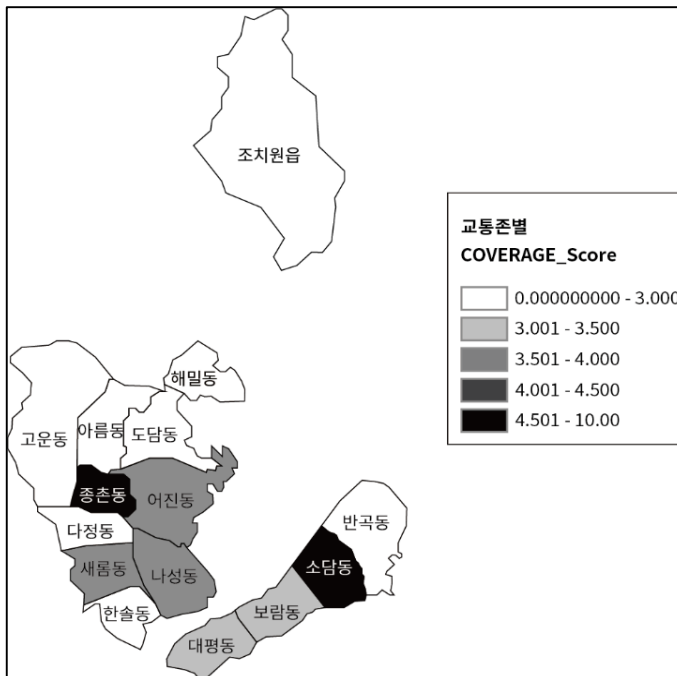
인구밀도가 높은 지역이면서, Capacity, Coverage, Frequency 값들이 적을수록 버스 서비스 공급이 필요한 지역으로 판단

$$\text{Standardized score} = \frac{[\text{Capacity, Coverage, or Frequency score}] - [\text{mean of distribution}]}{[\text{Standard deviation}]}$$

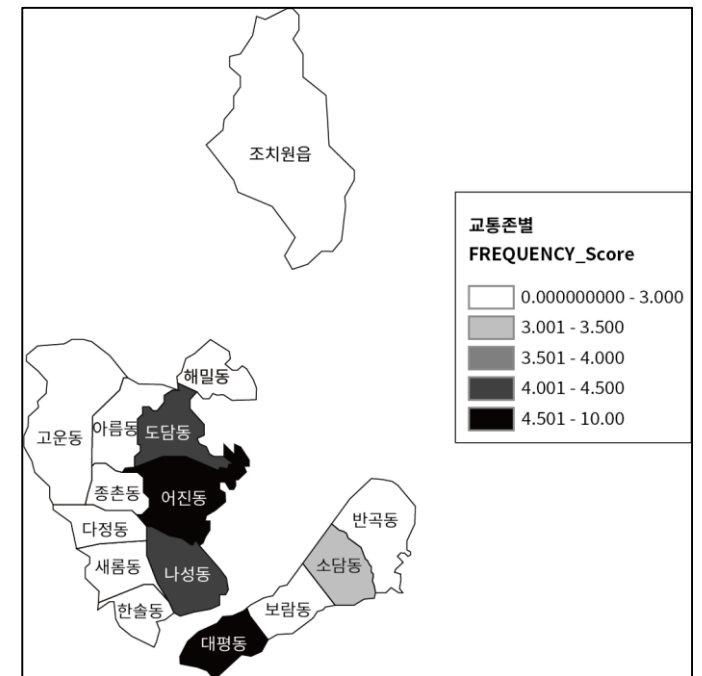
Rescaling : Standardized score + 3



구역별 Capacity Score



구역별 Coverage Score

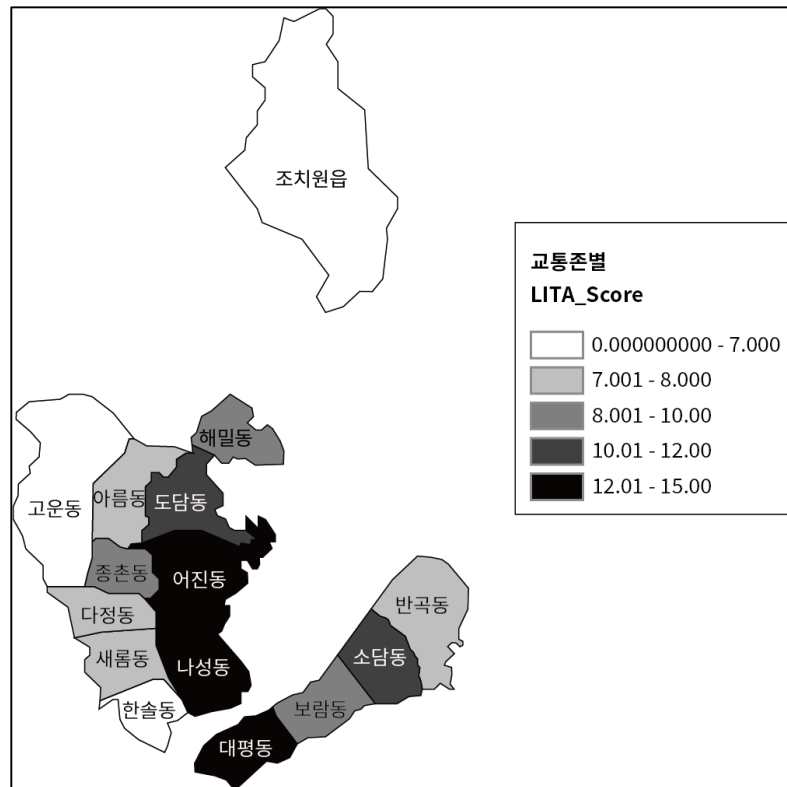


구역별 Frequency Score

데이터 분석 과정

• 지역 대중교통 이용도 지수(LITA: Local Index of Transit Availability)

$$\text{Overall LITA score} = [\text{Capacity score}] + [\text{Coverage score}] + [\text{Frequency score}]$$



구역별 LITA Score

| No.TAZ | 인구수 | Index | Capacity | Index | Coverage | Index | Frequency | Index | Total Index |
|--------|-------|-------|----------|-------|----------|-------|-----------|-------|-------------|
| 조치원읍 | 41529 | 5 | 2.1747 | 5 | 1.907 | 4 | 2.4857 | 4 | 18 |
| 고운동 | 35172 | 5 | 2.148 | 5 | 2.2012 | 4 | 2.0821 | 5 | 19 |
| 다정동 | 28550 | 5 | 2.1637 | 5 | 2.9424 | 3 | 2.1181 | 4 | 17 |
| 종촌동 | 28381 | 5 | 2.2084 | 4 | 4.5152 | 1 | 2.0821 | 5 | 15 |
| 반곡동 | 28139 | 5 | 2.4059 | 4 | 2.6865 | 3 | 2.7214 | 3 | 15 |
| 새롬동 | 26560 | 4 | 2.1312 | 5 | 3.5254 | 2 | 1.8304 | 5 | 16 |
| 도담동 | 25419 | 4 | 3.0614 | 3 | 2.9197 | 3 | 4.0279 | 2 | 12 |
| 아름동 | 23583 | 4 | 2.4224 | 4 | 2.513 | 3 | 2.8293 | 3 | 14 |
| 소담동 | 21850 | 3 | 2.905 | 3 | 4.836 | 1 | 3.061 | 3 | 10 |
| 보람동 | 18986 | 3 | 2.9494 | 3 | 3.2165 | 2 | 2.8213 | 3 | 11 |
| 한솔동 | 18370 | 3 | 2.3442 | 4 | 1.1932 | 5 | 2.222 | 4 | 16 |
| 나성동 | 13354 | 2 | 4.3297 | 2 | 3.625 | 2 | 4.4873 | 1 | 7 |
| 대평동 | 11262 | 1 | 4.3722 | 2 | 3.1292 | 2 | 4.5113 | 1 | 6 |
| 어진동 | 11211 | 1 | 4.7394 | 1 | 3.9289 | 1 | 4.8909 | 1 | 4 |
| 해밀동 | 8953 | 1 | 4.6443 | 1 | 1.8609 | 5 | 2.8293 | 3 | 10 |

Maximization

Population =if(Tazi)>27000,"5",if(Tazi)>22000,"4",if(Tazi)>17000,"3",if(Tazi)>12000,"2","1"))))

Minimization

Capacity =if(Tazi)>4.4,"1",if(Tazi)>3.2,"2",if(Tazi)>2.7,"3",if(Tazi)>2.2,"4","5"))))

Coverage =if(Tazi)>3.9,"1",if(Tazi)>3.1,"2",if(Tazi)>2.5,"3",if(Tazi)>1.9,"4","5"))))

Frequency =if(Tazi)>4.1,"1",if(Tazi)>3.4,"2",if(Tazi)>2.6,"3",if(Tazi)>2.1,"4","5"))))

데이터 분석 과정

• 링크 저항 계수(Impedance Length)

앞서 구한 Total Index 값을 이용하여 지수값(S_i)를 부여

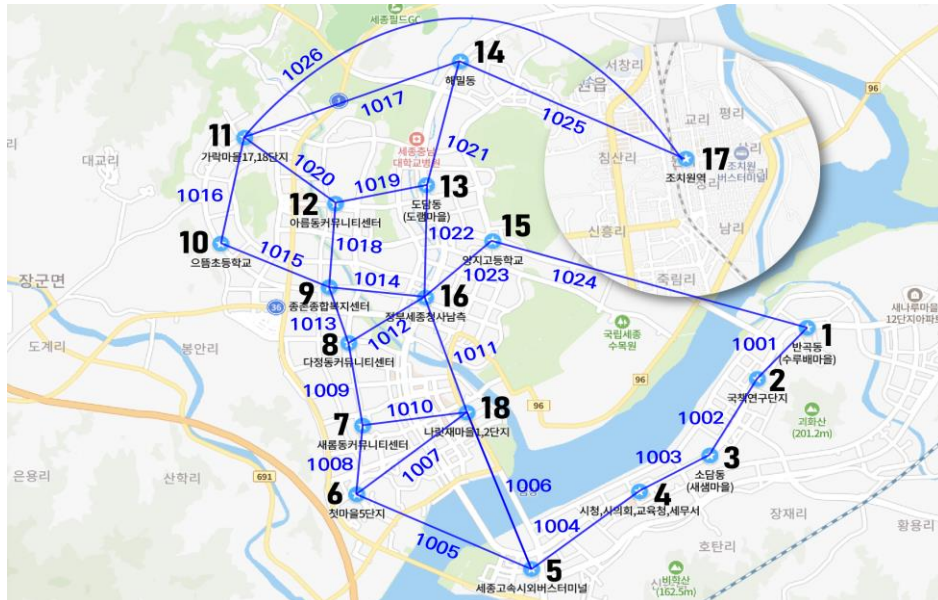
이때, 지수값은 링크 기반 지수값으로 변환하기 위해
각 존내 버스 노선링크 길이에 따라 링크저항계수를 산출

$$C_i = (S_{max}/s_i) \times L_i$$

S_{max} : the maximum value of s_i of all segments

s_i : the optimization score of segment i

L_i : length of the segment i in kilometers



| Link ID | Li | Si | Smax/Si | Impedance Length(Ci) | -> Ci Round(3) |
|---------|-----|------|-----------|----------------------|----------------|
| 1001 | 1.1 | 12.5 | 0.6578947 | 0.723684211 | 0.724 |
| 1002 | 1.1 | 12.5 | 0.6578947 | 0.723684211 | 0.724 |
| 1003 | 0.9 | 10.5 | 0.5526316 | 0.497368421 | 0.497 |
| 1004 | 2.8 | 8.5 | 0.4473684 | 1.252631579 | 1.253 |
| 1005 | 3.1 | 11 | 0.5789474 | 1.794736842 | 1.795 |
| 1006 | 2.4 | 6.5 | 0.3421053 | 0.821052632 | 0.821 |
| 1007 | 1.9 | 11.5 | 0.6052632 | 1.15 | 1.15 |
| 1008 | 1.2 | 16 | 0.8421053 | 1.010526316 | 1.011 |
| 1009 | 0.9 | 16.5 | 0.8684211 | 0.781578947 | 0.782 |
| 1010 | 1.7 | 11.5 | 0.6052632 | 1.028947368 | 1.029 |
| 1011 | 1.9 | 5.5 | 0.2894737 | 0.55 | 0.55 |
| 1012 | 1.7 | 10.5 | 0.5526316 | 0.939473684 | 0.939 |
| 1013 | 1.2 | 16 | 0.8421053 | 1.010526316 | 1.011 |
| 1014 | 1.3 | 9.5 | 0.5 | 0.65 | 0.65 |
| 1015 | 1.5 | 17 | 0.8947368 | 1.342105263 | 1.342 |
| 1016 | 1.6 | 19 | 1 | 1.6 | 1.6 |
| 1017 | 3.5 | 14.5 | 0.7631579 | 2.671052632 | 2.671 |
| 1018 | 1.6 | 14.5 | 0.7631579 | 1.221052632 | 1.221 |
| 1019 | 1.3 | 13 | 0.6842105 | 0.889473684 | 0.889 |
| 1020 | 2.1 | 16.5 | 0.8684211 | 1.823684211 | 1.824 |
| 1021 | 1.8 | 11 | 0.5789474 | 1.042105263 | 1.042 |
| 1022 | 1.5 | 8 | 0.4210526 | 0.631578947 | 0.632 |
| 1023 | 1.3 | 4 | 0.2105263 | 0.273684211 | 0.274 |
| 1024 | 4.5 | 9.5 | 0.5 | 2.25 | 2.25 |
| 1025 | 11 | 14 | 0.7368421 | 8.105263158 | 8.105 |
| 1026 | 12 | 18.5 | 0.9736842 | 11.68421053 | 11.684 |

데이터 분석 과정

• 신설노선(안) 생성(Dijkstra's Algorithm*)

지역내 버스노선체계 내에서 새로운 버스노선투입이 필요한 기종점 교통존을 파악하여 기종점간 링크의 링크 저항 계수값이 최소값을 갖는 최적노선을 도출

| Start | End | Route | Existing Bus | |
|-------|-----|-------------|------------------------------------|------|
| 1 | 16 | 1 15 16 | B0 | 노선 1 |
| 2 | 16 | 2 1 15 16 | 221, B0 | |
| 3 | 16 | 3 4 5 18 16 | 991, B1 | |
| 4 | 16 | 4 5 18 16 | 991, B1 | |
| 5 | 16 | 5 18 16 | 991, B0, B2, B3 | |
| 6 | 16 | 6 18 16 | 601 | 노선 2 |
| 7 | 16 | 7 18 16 | | |
| 8 | 16 | 8 16 | 203, 204 | |
| 9 | 16 | 9 16 | 201, 52 | |
| 10 | 16 | 10 9 16 | 201, 202 | |
| 11 | 16 | 11 12 13 16 | 201 | 노선 3 |
| 12 | 16 | 12 13 16 | 202 | |
| 13 | 16 | 13 16 | 201, 430, 550, 601, 991, B0, B1 | |
| 14 | 16 | 14 13 16 | 550, 551, 601, 991, B0, B1, B2, B3 | |
| 15 | 16 | 15 16 | 204, 222 | |
| 16 | 16 | 16 | X | |
| 17 | 16 | 17 14 13 16 | 430, 551, 601, 991 | |
| 18 | 16 | 18 16 | 550, 601, 991, 1000 | |

```

#include <iostream>
#include <vector>
#include <deque>
#include <queue>
#include <algorithm>
#include <fstream>

struct Edge {
    int dst;
    float weight;
};

struct Node {
    int data;
    std::vector<Edge> adjList;
};

struct Graph {
    std::vector<Node> nodes;

    void addNode(int data) {
        nodes.push_back({ data });
    }

    void addEdge(int from, int to, float w) {
        nodes[from].adjList.push_back({ to, w });
        nodes[to].adjList.push_back({ from, w });
    }

    void minimumSpanningTree(int start, int end) const {
        struct NodeInfo {
            int node;
            float dist;
        };

        struct NodeInfoCompare {
            bool operator()(const NodeInfo& a, const NodeInfo& b) {
                return a.dist > b.dist;
            }
        };

        std::priority_queue<NodeInfo, std::vector<NodeInfo>, NodeInfoCompare> open;
        std::vector<bool> closed(nodes.size(), false);
        std::vector<float> dist(nodes.size(), 0);
        std::vector<int> parents(nodes.size(), -1);

        open.push({ start, 0 });
        dist[start] = 0;

        while (!open.empty()) {
            NodeInfo info = open.top();
            open.pop();

            if (closed[info.node]) continue;
            closed[info.node] = true;
            if (closed[end]) break;

            for (auto edge : nodes[info.node].adjList) {
                int dst = edge.dst;

                if (closed[dst]) continue;

                if (dist[dst] > edge.weight + dist[info.node]) {
                    parents[dst] = info.node;
                    dist[dst] = edge.weight + dist[info.node];
                    open.push({ dst, dist[dst] });
                }
            }
        }

        std::vector<int> backtracking;
        int cur = end;

        while (cur != start) {
            backtracking.push_back(cur);
            cur = parents[cur];
        }

        backtracking.push_back(start);
        std::reverse(backtracking.begin(), backtracking.end());

        for (int i = 0; i < backtracking.size(); i++)
            std::cout << backtracking[i] + 1 << " ";

        std::cout << std::endl;
    }
};
    
```

*다익스트라 알고리즘: 도로 교통망 등에서 나타낼 수 있는 그래프로, 꼭짓점 간의 최단 경로를 찾는 알고리즘

데이터 분석 과정

• 신설노선(안) 생성(Dijkstra's Algorithm)



| Start | End | Route | Existing Bus | |
|-------|-----|---------|--------------|------|
| 1 | 16 | 1 15 16 | B0 | 노선 1 |

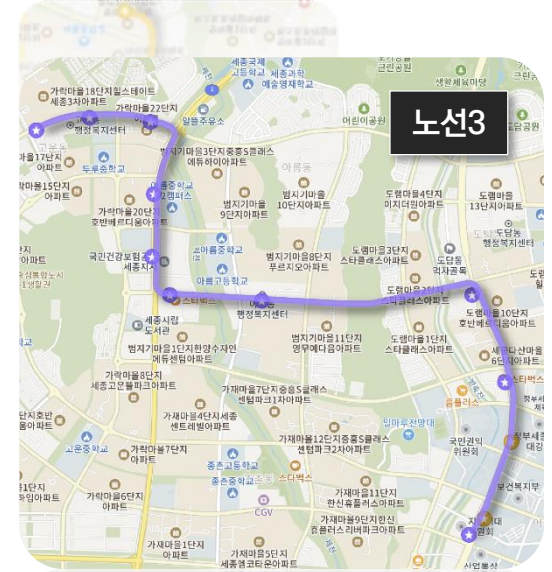
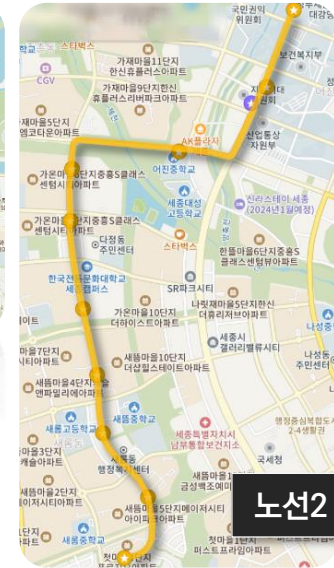
- 노선1 주요 정류장: 반곡동(수루배마을) → 양지고등학교 → 정부세종청사중앙동

| | | | | |
|---|----|---------|----------|------|
| 6 | 16 | 6 18 16 | 601 | |
| 7 | 16 | 7 18 16 | | 노선 2 |
| 8 | 16 | 8 16 | 203, 204 | |

- 노선2 주요 정류장: 첫마을5단지 → 새롬동커뮤니티센터 → 다정동커뮤니티센터 → 정부세종청사남측

| | | | | |
|----|----|-------------|-----|------|
| 11 | 16 | 11 12 13 16 | 201 | 노선 3 |
| 12 | 16 | 12 13 16 | 202 | |

- 노선3 주요 정류장: 가락마을17,18단지 → 아람동커뮤니티센터 → 도람마을8,10단지 → 정부세종청사남측



04 활용방안 및 기대효과

04 활용방안 및 기대효과

참고문헌 및 출처 >>