서울 지하철 혼잡도를 고려한 최적 경로 탐색 알고리즘

20230808 오승현

1. 문제 정의

본 과제에서는 서울 지하철 이용 시 단순한 최단 이동 시간이나 최단 이동 거리가 아닌, 열차 혼잡도와 환승 소요 시간을 함께 고려한 최적 이동 경로를 탐색하는 알고리즘을 구현한다.

기존의 지하철 경로 탐색 서비스는 이동 시간만을 기준으로 경로를 추천하는 경우가 많지만, 실제 이용자 입장에서는 출퇴근 시간대의 혼잡도나 잦은 환승으로 인한 피로도 또한 중요한 요소로 작용한다.

이에 본 과제에서는 서울교통공사에서 제공하는 실제 지하철 역간 소요 시간, 환승 정보, 혼잡도 데이터를 활용하여 지하철 노선을 그래프로 모델링하고, 각 이동 구간에 혼잡도 및 환승 페널티를 반영한 가중치 기반 최단 경로 탐색 알고리즘을 적용하였다.

특히 출발역: 철산역에서 도착역: 월곡역까지의 실제 통학 경로를 사례로 설정하여, 알고리즘이 현실적인 이동 부담을 얼마나 잘 반영하는지를 확인하고자 한다.

2. 데이터 전처리

본 과제에서는 공공데이터포털을 통해 제공되는 서울교통공사의 실제 지하철 데이터를 CSV 형태로 수집하여 사용하였다. 사용한 데이터는 역간 거리 및 소요시간 데이터, 환승역 거리 및 소요시간 정보, 지하철 혼잡도 정보이다.

역간 소요시간 데이터에서는 호선, 연번, 역명, 소요시간 컬럼을 사용하였으며, 소요시간은 알고리즘 처리의 편의성을 위해 초 단위로 변환하였다. 같은 호선 내에서 연번 순으로 정렬한 뒤, 인접한 두 역을 간선으로 연결하여 (역명, 호선)을 노드로 하는 가중 그래프를 구성하였다. 환승역 거리 및 소요시간 데이터에서는 환승역명, 기존 호선, 환승 노선, 환승 소요시간 정보를 이용하여 동일한 역명에서 서로 다른 호선으로 이동할 수 있도록 환승 간선을 추가하였다.

지하철 혼잡도 데이터는 요일 구분(평일 기준)과 특정 시간대(예: 오전 8시)를 선택하여, (호선, 출발역)을 기준으로 상선과 하선의 혼잡도 값을 평균하여 하나의 혼잡도 값으로 계산하였다. 해당 역이나 시간대의 데이터가 존재하지 않는 경우에는 전체 평균 혼잡도 값으로 대체하여 사용하였다.

3. 구현 방법 및 사용된 알고리즘

본 과제에서는 서울 지하철 노선을 그래프 구조로 모델링하여 최적 경로 탐색 문제로 정의하였다. 그래프의 각 노드는 (역명, 호선) 형태로 정의하였으며, 같은 역이라도 호선이

다를 경우 서로 다른 노드로 취급하였다. 이는 환승을 명시적으로 모델링하기 위함이다.

간선은 크게 두 종류로 구성된다. 첫째, 같은 호선 내에서 인접한 두 역 사이의 이동 간선으로, 역간 이동 시간과 혼잡도 페널티를 가중치로 부여하였다. 둘째, 환승역에서 호선이 변경되는 경우를 표현하기 위해 환승 소요시간을 가중치로 갖는 환승 간선을 추가하였다.

각 간선의 가중치는 다음과 같이 정의된다.

이동 간선 가중치 = 역간 이동시간 + β × 혼잡도

환승 간선 가중치 = α × 환승 소요시간

여기서 β는 혼잡도의 중요도를 조절하는 계수이며, α는 환승 소요시간의 영향을 조절하는 계수이다. 본 과제에서는 실험을 통해 β=2.0, α=1.0으로 설정하였다.

모든 간선 가중치는 0 이상의 값을 가지므로, 가중치 기반 최단 경로 문제를 해결하기 위해 Dijkstra 알고리즘을 사용하였다. Dijkstra 알고리즘은 출발 노드로부터 각 노드까지의 최소 비용 경로를 효율적으로 탐색할 수 있는 알고리즘이다.

4. 실행 결과 및 성능 분석

알고리즘의 실행 결과, 출발역: 철산역에서 도착역: 월곡역까지의 최적 경로는 7호선, 3호선, 5호선, 6호선을 순차적으로 이용하는 경로로 도출되었다. 경로 중 동일한 역명이 반복되는 구간은 환승을 의미하며, 서로 다른 호선 간 이동이 환승 간선으로 모델링되었음을 나타낸다.

총 비용은 이동 시간, 혼잡도 페널티, 환승 페널티를 종합적으로 고려한 값으로 계산되었으며, 단순 최단 이동 시간과는 다른 현실적인 경로가 선택되었다.

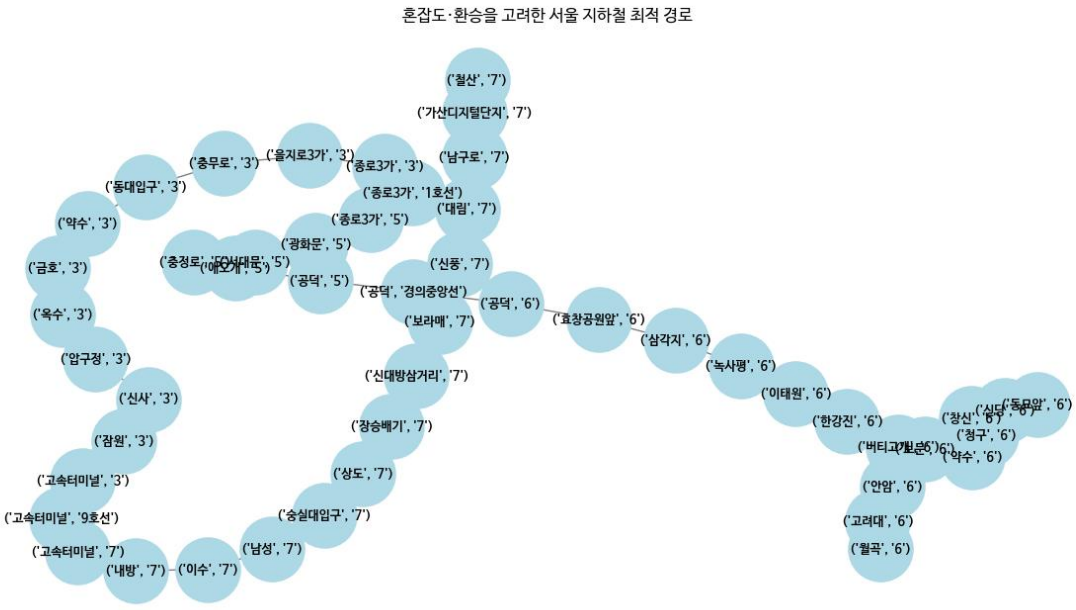
본 그래프의 노드 수는 약 400개 수준이며, Dijkstra 알고리즘의 시간 복잡도는 O((V + E) log V)이다. 실제 실행 결과, 경로 탐색에 소요된 시간은 약 0.001초로 매우 짧아, 본 문제 규모에서 알고리즘이 효율적으로 동작함을 확인하였다.

5. 시각화

알고리즘의 실행 결과를 직관적으로 확인하기 위해, 탐색된 최적 경로에 포함된 노드와 간선만을 추출하여 그래프 형태로 시각화하였다. 전체 지하철 네트워크 대신 최적 경로만을 시각화함으로써, 출발역부터 도착역까지의 이동 흐름과 환승 지점을 명확하게 확인할 수 있었다.

그림 1은 본 알고리즘을 적용하여 출발역: 철산역에서 도착역: 월곡역까지의 최적 경로를 시각화한 결과로, 노드는 (역명, 호선) 단위로 구성되며 동일 역에서의 호선 변경은 환승

을 의미한다.



(그림1) 혼잡도와 환승을 고려한 서울 지하철 최적 경로 시각화

6. 개선 방안 및 확장 가능성

향후 개선 방안으로는 시간대별 혼잡도를 자동으로 반영하여 출퇴근 시간, 주말 등 다양한 상황에 따른 경로를 추천할 수 있도록 확장할 수 있다. 또한 A* 알고리즘과의 성능 비교를 통해 탐색 효율을 추가로 분석할 수 있으며, 사용자가 혼잡도 회피 또는 이동 시간 최소화 중 선호 조건을 선택할 수 있도록 가중치를 조정하는 방식도 고려할 수 있다.