

# 은평 한옥마을 관광객의 선호도 및 이동패턴을 분석하여 최적의 관광지 경로를 도출

임수린 · 김주원 · 이준영

하나고등학교

## “Analyzing Tourist Preferences and Movement Patterns in Eunpyeong Hanok Village to Derive Optimal Tourist Routes”

Su-Lyn Im · Joo-Won Kim · Jun-Young Lee

Hana Academy Seoul

**Abstract** Hana High School is located near Bukhansan Mountain and Eunpyeong Hanok Village, areas often congested with tourists. High population density in these areas can cause mobility issues and safety concerns. This study aims to analyze the preferences and movement patterns of tourists in Eunpyeong Hanok Village using mathematical and computational approaches to derive optimized tourist routes. By alleviating congestion and distributing visitors evenly, the project enhances tourist satisfaction, strengthens safety, and boosts local economic activity. The results are expected to provide valuable data for sustainable tourism management in the Eunpyeong district and beyond.

**Key Words** : Tourist Route Optimization, Movement Pattern Analysis, Sustainable Tourism, Local Economy Boost, Congestion Management

### 1. 서론

현대 관광 산업은 방문객의 경험과 만족도를 극대화하면서 지역 경제 활성화를 도모하는 데 초점을 맞추고 있다. 도시형 관광지는 공간적 제약과 높은 밀집도로 인해 혼잡도 관리와 동선 최적화가 중요한 과제로 부각되고 있으며, 은평 한옥마을은 전통적 미학과 현대적 접근성을 동시에 갖춘 대표적인 도시형 관광지로서 이와 같은 문제를 안고 있다. 매년 증가하는 관광객 수는 특정 시간대와 장소의 혼잡도를 심화시키고 있으며, 이는 관광객 이동의 불편함과 지역 안전 문제로 이어지고 있다.

기존 연구들은 관광지 혼잡도를 완화하기 위해 다양한 최적화 알고리즘을 제안했지만, 관광객의 이동 패턴과 선호도를 기반으로 한 종합적인 분석은 상대적으로 미흡했다. 예를 들어, Xiao et al. (2021)은 관광지 내 혼잡도를 완화하기 위해 군집 기반 접근법을 제안했으나, 이는 주로 정적인 데이터에 기반

을 두고 있어 실시간 변화를 반영하기 어렵다는 한계를 가진다. 또한, Kim et al. (2020)은 빅데이터와 기계 학습을 활용해 관광객 동선을 예측하려 했지만, 이동 패턴에 대한 확률적 접근보다는 데이터 기반 규칙 설정에 중점을 두었다.

본 연구는 은평 한옥마을 관광객의 이동 패턴, 선호도, 체류 시간을 수학적 모델링과 정보학적 분석을 통해 이해하고, 이를 기반으로 관광객 만족도를 극대화하면서 혼잡도를 완화할 수 있는 최적의 경로를 제시하는 것을 목표로 한다. 특히, 마르코프 체인(Markov Chain)을 활용하여 관광객의 동적 이동 확률을 계산하고, 이를 최적화 알고리즘과 결합하여 경로를 설계할 예정이다. 이는 관광객의 분산을 통해 혼잡도를 완화하고 지역 경제를 활성화하며, 지속 가능한 관광 모델을 제시하는 데 기여할 것이다.

예를 들어, Shoval et al. (2018)은 관광객의 이동 데이터를 분석해 시계열 모델을 적용한 연구를 통해 관광지 관리의 새로운 가능성을 제시했지만, 특정 시간대와 공간의 혼잡도 최적화에 한정된 접근법이 었다. 본 연구는 이러한 기존 연구의 한계를 보완하고자 마르코프 체인의 동적 특성과 최적화 알고리즘을 결합하여 보다 실질적이고 적용 가능한 결과를 도출하고자 한다.

은평 한옥마을과 연결된 북한산 둘레길까지를 포함한 통합적 경로 설계는 자연 관광과 도시형 관광의 융합적 경험을 제공하며, 도심과 자연이 공존하는 새로운 관광 모델을 구현하는 데 중요한 시사점을 제공할 것으로 기대된다. 이는 Shoval et al. (2018) 및 Xiao et al. (2021)의 결과를 확장하여 실시간 데이터 기반의 최적화 경로 설계 가능성을 제시하며, 지역 관광지 관리 및 계획의 새로운 가능성을 열어줄 것이다.

## 2. 이론적 배경

### 2.1. 마르코프 체인 (Markov Chain)

마르코프 체인 (Markov Chain)은 확률 이론에서 사용되는 도구로, 시간에 따라 변화하는 상태의 연속적인 과정을 모델링할 때 유용하다. 마르코프 체인의 가장 큰 특징은 마르코프 성질로, 이는 현재 상태가 과거의 상태와 독립적이고 오직 현재 상태에 의해서만 미래

상태가 결정된다는 것을 의미한다. 이를 수학적으로 표현하면 다음과 같다.

$$P(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P\{X_{n+1} = x_{n+1} | X_n = x_n\} \#(1)$$

여기서  $X_n$  은 시간  $n$ 에서의 상태를 나타낸다. 마르코프 체인은 상태 공간 (State Space)  $S = \{s_1, s_2, \dots, s_m\}$ 과 전이 확률 행렬 (Transition Probability Matrix)  $P$ 로 정의된다. 상태  $i$ 에서  $j$ 로 전이할 확률은 다음과 같다.

$$P_{ij} = P(X_{n+1} = s_j | X_n = s_i) \#(2)$$

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix}$$

위의 행렬  $P$ 는 각 행의 합이 1이 되는 확률 행렬의 성질은 만족한다. 즉,

$$\sum_{j=1}^m P_{ij} = 1, \forall i$$

### 2.1.1. 상태의 분류와 특징

마르코프 체인의 상태는 다음과 같이 분류된다.

- 일시적 상태 (Transient State): 한번 방문한 후 다시 방문하지 않은 확률이 양수인 상태
  - 재귀적 상태 (Recurrent State): 한번 방문한 후 반드시 다시 방문할 확률이 1인 상태
  - 불변 상태 (Invariant State): 체인이 시간이 무한히 경과한 뒤에도 상태가 변하지 않는 상태
- 이러한 상태 분류를 통해 체인의 장기 거동을 분석할 수 있다.

### 2.1.2. 마르코프 체인의 안정 상태 (Steady State)

시간이 충분히 경과한 후 마르코프 체인이 수렴하는 상태 분포를 안정 상태 (Steady State)라고 한다. 안정 상태에서 각 상태  $s_i$ 의 확률  $\pi_i$ 는 다음을 만족한다.

$$\pi P = \pi$$

$$\sum_{i=1}^m \pi_i = 1$$

여기서  $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ 는 안정 상태 확률 벡터 (Steady State Probability Vector)이다.

### 2.1.3. 응용: 이동 패턴 분석

관광객의 이동 패턴 분석에서 마르코프 체인은 관광객이 특정 지점에서 다른 지점으로 이동할 확률을 모델링하는 데 유용하다. 예를 들어, 상태 공간  $S$ 가 관광지의 집합이고, 각 상태는 특정 관광지를 나타낸다고 하자.  $P_{ij}$ 는 관광객이  $i$ 번 관광지에서  $j$ 번 관광지로 이동할 확률을 의미한다.

관광객의 초기 상태를 나타내는 확률 분포  $\pi^{(0)}$ 가 주어졌을 때,  $n$  단계 후 상태 분포는 다음과 같이 계산된다.

$$\pi^{(n)} = \pi^{(0)} P^n$$

이를 통해 관광객의 장기적인 이동 패턴 및 특정 관광지에 체류할 확률을 추정할 수 있다.

#### 2.1.4. 마르코프 체인의 활용

마르코프 체인은 관광지 경로 최적화 문제에 효과적으로 사용될 수 있다. 관광객의 이동 데이터를 바탕으로 전이 확률  $P$ 를 추정하고, 특정 관광지의 방문 빈도를 계산하여 주요 경로를 도출할 수 있다. 또한, 안정 상태 확률  $\pi$ 를 이용하여 관광객이 가장 선호하는 지점을 식별할 수 있다.

이처럼 마르코프 체인은 단순한 이동 패턴부터 복잡한 경로 최적화까지 다양한 문제를 해결하는 데 적합한 기법으로, 관광지 경로 분석에 적합하다.

### 3. 연구 방법 및 절차

#### 3.1. 연구 방법

먼저 각 관광지에서의 관광객들의 특성을 파악하기 위해, 은평 한옥마을 방문객들을 대상으로 설문 조사를 진행하였다. 설문 조사 문항은 각 관광지들의 체류 시간을 10분 단위를 기준으로 객관식으로 묻고, 그 후에 가장 마음에 들었던 관광지에 순위를 매기는 형식으로 진행하였다. 설문 조사를 통해 데이터 자료를 수집한 후에는 해당 데이터를 가지고 마르코프 행렬을 만드는 과정을 진행하였다. 기존 마르코프 행렬의 문제점을 보완하기 위해 각각의 관광지를 한 정점으로 두는 것이 아닌, 시간에 따라 체류시간 10분 단위로 정점을 분할하였다. 해당 정점을 분할 한 이후로는, 마르코프 행렬의 각 원소들을 주어진 데이터를 가지고 계산하였다. 행렬의 원소는 이동 시간과, 선호도를 중심으로 결정하였고 계산이 끝난 후에는 둘의 곱 연산을 통해 최종적으로 행렬을 산출하였다. 이렇게 만들어진 행렬을 거듭제곱하면서 시간 단위로 전이 행렬을 계산하였고, 이렇게 최종적으로 최적의 확률을 계산하였다. 확률을 계산 한 후에는 행렬을 통해 계산한 첫 점과 끝 점을 가지고 다이나믹 프로그래밍을 진행하여 이동 경로를 추적하였고, 최종적으로 이동 경로를 계산할 수 있었다.

#### 3.2. 연구 절차

먼저 은평 한옥마을의 지도를 분석하여 주요 관광지를 도출한다. 도출된 관광지는 입구, 은평 역사 한옥마을 박물관, 진관사 입구, 찻집인 물다움, 상업거리, 거주공간에 위치한 나무, 터널 앞 음식점 구간 등이다. 이를 바탕으로 설문지를 작성하여 관광객들의 선호도와 체류 시간을 조사할 수 있도록 한다. 설문조사는 각 방문 장소별 체류 시간을 10분 단위로 기록할 수 있는 형태로 설계되며, 특정 지점에 QR 코드와 링크를 배치하여 참여를 유도한다. 또한, 외국인 관광객 및 특정 수요층을 대상으로 인터뷰를 진행하여 보다 심층적인 자료를 수집한다.

설문조사를 기반으로 각 관광지의 선호도와 체류 시간, 이동 시간 등을 조사한다. 조사된 데이터를 활용하여 마르코프 행렬의 원소값들을 계산하였다. 각각의 값들은 설문조사를 통해 얻은 선호도 데이터, 체류 시간 데이터 각각에 대한 가중치를 계산한 후, 해당 두 값들의 곱연산을 통해 구한다.. 이렇게 도출된 값을 마르코프 행렬의 초기 조건으로 사용하여 행렬 연산을 진행한다.

해당 마르코프 행렬을 구성 한 이후에는 동적 계획법을 활용하여 연산한다. 계산된 마르코프 행렬을 통해 최적 경로를 탐색하며, 이를 시각화한다. 시각화는 주요 관광지를 점으로 표현한 그래프 형태로 진행하여 최적 경로를 직관적으로 확인할 수 있도록 한다.

이 모든 과정은 은평 한옥마을의 관광 흐름과 선호도를 분석하고, 효율적인 관광 경로를 제안하기 위해 체계적으로 이루어진다.

Table 1. Locations and Participations

|     | 10대   | 20대   | 30대   | 40대   | 50대   | 60대& 그 이상 |
|-----|-------|-------|-------|-------|-------|-----------|
| Age | 23.1% | 23.1% | 18.5% | 21.5% | 10.8% | 3.0%      |

Table 2. Residence time

|   | 없음    | 10분   | 20분   | 30분   | 40분   | 50분   | 60분& 그 이상 |
|---|-------|-------|-------|-------|-------|-------|-----------|
| Eunpyeong History Hanok Museum  | 18.5% | 4.6%  | 1.5%  | 6.2%  | 10.8% | 12.3% | 46.2%     |
| Jingwan Temple  | 18.5% | 4.6%  | 4.6%  | 21.5% | 12.3% | 10.8% | 27.7%     |
| Muldaum Tea House   | 47.7% | 6.2%  | 7.7%  | 12.3% | 4.6%  | 7.7%  | 13.8%     |
| Commercial Street(Twosome Place, Convenience store)                                   | 7.7%  | 15.4% | 9.2%  | 16.9% | 4.6%  | 12.3% | 33.8%     |
| Bukhansan Dulle-Gil Area  | 36.9% | 1.5%  | 3.1%  | 7.7%  | 7.7%  | 15.4% | 27.7%     |
| The tree(at the center of the Hanok Village)  | 15.4% | 18.5% | 16.9% | 23.1% | 9.2%  | 7.7%  | 9.2%      |
| Restaurants near the tunnel in Eunpyeong Hanok Village(Seogwamyeonok, Longbread etc.) | 18.5% | 4.6%  | 1.5%  | 6.2%  | 10.8% | 12.3% | 46.2%     |

Table 3. preference of tourist attraction

|    | Eunpyeong History Hanok Museum | Jingwannon Temple | Muldau Tea House | Commercial Street(Twosome Place, Convenience store) | Bukhansan Dulle-Gil Area | The Tree(at the center of the Hanok Villiage) | Restaurants near the tunnel in Eunpyeong Hanok Village (Seogwanmyeonok, Longbread etc.) |
|----|--------------------------------|-------------------|------------------|---|--------------------------|---|---|
| 1위 | 5.7%                           | 22.9%             | 1.4%             | 21.4%   | 10.0%                    | 18.6%   | 20.0%   |
| 2위 | 8.9%                           | 16.4%             | 1.5%             | 37.3%   | 6.0%                     | 9.0%  | 20.9%   |
| 3위 | 7.4%                           | 25.0%             | 7.3%             | 17.7%   | 8.8%                     | 14.7%   | 19.1%   |

4. 연구 결과

앞선 방법들을 활용하여, 최종적으로 관광지 경로를 계산하는 알고리즘을 만들 수 있었다. 해당 코드는 다음과 같다.

```
import math

n=7 #place
size=6*n #node (한 관광지를 시간별로 6개의 정점으로 나눠 생각)

#체류 시간
data=[[3.1,9.2,12.3,9.2,12.3,20],
      [4.6,4.6,21.5,12.3,10.8,27.7],
      [6.2,7.7,12.3,4.6,7.7,13.8],
      [15.4,9.2,16.9,4.6,12.3,33.8],
      [18.5,16.9,23.1,9.2,7.7,9.2],
      [1.5,3.1,7.7,7.7,15.4,27.7],
      [4.6,1.5,6.2,10.8,12.3,46.2]]
```

```

#matrix1 -> 해당 지역 (상태)에 남은 확률 계산
matrix1=[[0 for _ in range(size)] for _ in range(size)]
for i in range(n):
    for j in range(5):
        matrix1[6*i+j][6*i+j+1]=sum(data[i][j+1:])/sum(data[i][j:])

#선호도
prefer=[[4,16,1,15,7,13,14],
        [6,11,1,25,4,6,14],
        [5,17,5,12,6,10,13]]

score=[0 for _ in range(n)]
sc=[1,0.5,0.3]

for i in range(3):
    for j in range(n):
        score[j]+=prefer[i][j]*sc[i]

matrix2=[0 for _ in range(n)]

for i in range(n):
    matrix2[i]=1/(1+3*math.exp(-0.1*score[i]))

#이동 거리
data_distance=[[0,556.8,516.8,341.6,296.5,516.7,230.1],
               [556.8,0,70.2,318.8,190,604.5,645.7],
               [516.8,70.2,0,268.8,140,554.5,595.7],
               [341.6,318.8,268.8,0,238.2,635.1,284],
               [296.5,190,140,238.2,0,540.1,390.5],
               [516.7,604.5,554.5,635.1,540.1,0,605.6],
               [230.1,645.7,595.7,284,390.5,605.6,0]]

matrix3=[[0 for _ in range(n)] for _ in range(n)]

for i in range(n):
    for j in range(n):
        if i!=j:
            matrix3[i][j]=(100/data_distance[i][j])**0.5

#선호도+이동 거리
matrix4=[[0 for _ in range(n)] for _ in range(n)]
for i in range(n):
    for j in range(n):
        matrix4[i][j]=matrix2[i]*matrix3[i][j]

```

*#정규화*

```
for i in range(n):
    rowSum=sum(matrix4[i])
    for j in range(n):
        matrix4[i][j]/=rowSum
```

```
final_matrix=[[0 for _ in range(size)] for _ in range(size)]
```

```
for i in range(size):
    if i%6==5:
        r=1
    else:
        r=1-matrix1[i][i+1]
        final_matrix[i][i+1]=matrix1[i][i+1]
    for j in range(n):
        final_matrix[i][6*j]=matrix4[i//6][j]*r
```

```
def matrix_power (matrix1,matrix2,size):
    new=[[0.0 for _ in range(size)] for _ in range(size)]
    for i in range(size):
        for j in range(size):
            for k in range(size):
                new[i][j]+=matrix1[i][k]*matrix2[k][j]
    return new
```

power= 15 *# 150분 (관광 시간 = power \* 10 min)*

```
def markovchain (power,final_matrix):
    result=[[0.0 for _ in range(size)] for _ in range(size)]
    for i in range(size):
        result[i][i]=1.0
    binary=[]
    while power!=0:
        binary.append(power%2)
        power//=2
    for i in range(len(binary)-1,-1,-1):
        result=matrix_power(result,result,size)
        result=matrix_power(result,final_matrix,size)
    x,y=0,0
    maximum=-1
    for i in range(0,size,6):
        for j in range(size):
            if maximum<result[i][j]:
                x,y=i,j
                maximum=result[i][j]
    return [x,y]
```



```

print(markovchain(power,final_matrix)) # --> [6,12]

dp=[] for _ in range(size)
p=[0 for _ in range(size)]

# 초기 조건
dp[6].append(6)
p[6]=1.0

#dp를 통한 최적의 경로 계산
for i in range(power):
    ndp=[] for _ in range(size)
    np=[0 for _ in range(size)]
    for j in range(size):
        if p[j]!=0:
            for k in range(size):
                if final_matrix[j][k]!=0 and k not in dp[j]:
                    if np[k]<p[j]*final_matrix[j][k]:
                        np[k]=p[j]*final_matrix[j][k]
                        ndp[k]=dp[j][:]
                        ndp[k].append(k)

    dp=ndp[:]
    p=np[:]

maximum=0
final=0
for f in range(size):
    if p[f]>maximum:
        final=f
        maximum=p[f]

print(maximum)
print(f)
print(dp[f])

```

위 코드는 마르코프 체인 모델을 사용하여 관광이나 이동경로와 같은 상황에서 최적의 장소를 찾기 위한 목적으로 만들었다. 핵심은 체류시간, 선호도, 이동거리 등을 고려하여, 주어진 시간동안 어떤 장소에서 가장 많은 영향을 받는지를 계산하는 것이다.

#### 4-1. 초기 설정

장소의 개수는 7개로 설정( $n=7$ )되며, 각 장소마다 6개의 상태(노드)가 정의된다. 따라서 총 노드의 개수는  $\text{size} = 6 \times n = 42$ 개이다. 여기서 각 노드는 특정 상태(체류 시간, 이동, 선호도 등)를 나타내며, 마르코프 체인의 상태로 사용된다. 초기 데이터로는 체류 시간(data), 선호도(prefer), 이동 거리(data\_distance)가 주어지며, 이를 기반으로 각 장소 간 전이 확률을 계산한다.

#### 4-2. 체류 시간 기반 전이 확률 계산

각 장소 내에서 체류 시간에 따라 특정 상태에서 다음 상태로 전환될 확률을 계산하기 위해 matrix1을 생성한다.

예를 들어, 첫 번째 장소에서 첫 번째 상태에서 두 번째 상태로 전환될 확률은 다음 상태로 머무르는 총 시간을 현재 상태에 머무르는 총 시간으로 나눈 값으로 계산된다.

이를 통해 장소별로 체류 시간 기반 전이 확률 행렬이 구성되며, 관광객이 한 장소에서 다음 상태로 이동할 확률을 나타낸다.

#### 4-3. 장소 선호도 계산

관광객의 선호도를 반영하기 위해 각 장소에 대한 선호도 점수를 계산한다. 선호도 데이터(prefer)는 세 그룹(예: 낮, 중간, 높은 중요도)에 대해 가중치(1, 0.5, 0.3)를 부여하여 최종 점수를 도출한다.

이 점수를 바탕으로 각 장소에 대한 선호도를 확률로 변환하기 위해 함수  $1 / (1 + 3 * \exp(-0.1 * \text{score}))$ 가 적용된다. 이로써 관광객의 선호도가 높을수록 높은 확률이 할당되며, 이를 matrix2로 저장한다.

#### 4-4. 이동 거리 기반 전이 확률 계산

이동 거리를 기반으로 장소 간 전환 비용을 계산하기 위해 matrix3을 생성한다.

이동 거리 데이터(data\_distance)를 사용하여 각 장소 간의 거리의 역제곱근 값을 전환 확률로 정의한다. 예를 들어, 두 장소 간 거리가 가까울수록 전이 확률이 높아지며, 멀어질수록 낮아진다.

이를 통해 장소 간 거리의 물리적 제약이 반영된다.

#### 4-5. 최종 전이 확률 행렬 계산

선호도(matrix2)와 이동 거리(matrix3)를 결합하여 최종적인 장소 간 전이 확률 행렬 matrix4를 생성한다.

이후 각 장소 내 상태 전이 확률(matrix1)을 반영하여 최종적인 전이 확률 행렬 final\_matrix를 계산한다. 이 행렬은 관광객이 특정 상태에서 다음 상태나 다른 장소로 이동할 확률을 종합적으로 나타낸다.

#### 4-6. 마르코프 체인을 이용한 경로 탐색

마르코프 체인의 기본 원리를 이용하여 150분(15번 전이) 동안의 최적 경로를 예측한다.

markovchain 함수는 행렬 거듭제곱을 사용하여 초기 상태에서 모든 상태로의 도달 확률을 계산하며, 가장 높은 도달 확률을 가지는 상태와 노드를 반환한다. 예를 들어, 출력값 [6, 12]는 6번 노드에서 시작하여 12번 노드로 이동하는 것이 최적임을 나타낸다.

#### 4-7. DP를 활용한 경로 추적

마르코프 체인 결과로 얻은 최적 경로를 추적하기 위해 DP(Dynamic Programming)를 사용한다.

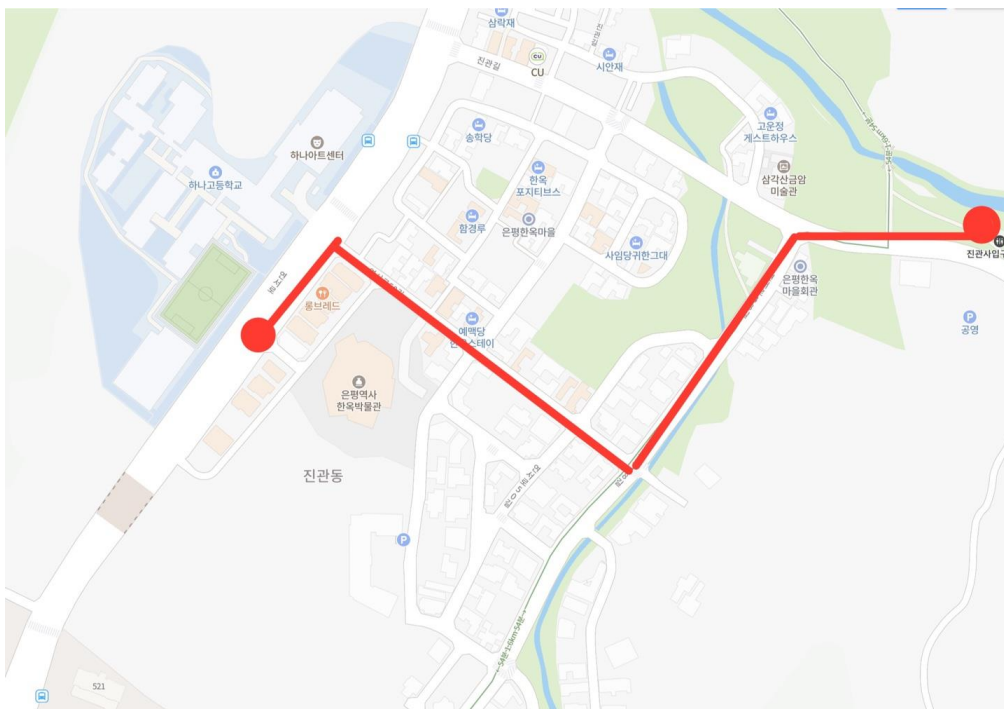
초기 조건으로 특정 시작 노드에서 확률  $p[6]=1.0$ 을 부여하며, 각 전이 단계마다 가능한 모든 경로와 확률을 갱신한다. 각 노드마다 가장 높은 확률을 유지하도록 설계되었으며, 마지막 단계에서 최대 확률을 가진 경로를 반환한다.

#### 4-8. 결과 해석

코드 출력에서 [6, 12]와 같은 값은 6번 노드에서 시작하여 12번 노드로 이동하는 경로가 최적임을 나타낸다. 추가적으로 DP 결과를 통해 가장 높은 확률로 도달 가능한 경로를 시각화할 수 있다. 이 결과는 체류 시간, 선호도, 이동 거리라는 세 가지 주요 요소가 복합적으로 작용했음을 보여준다.

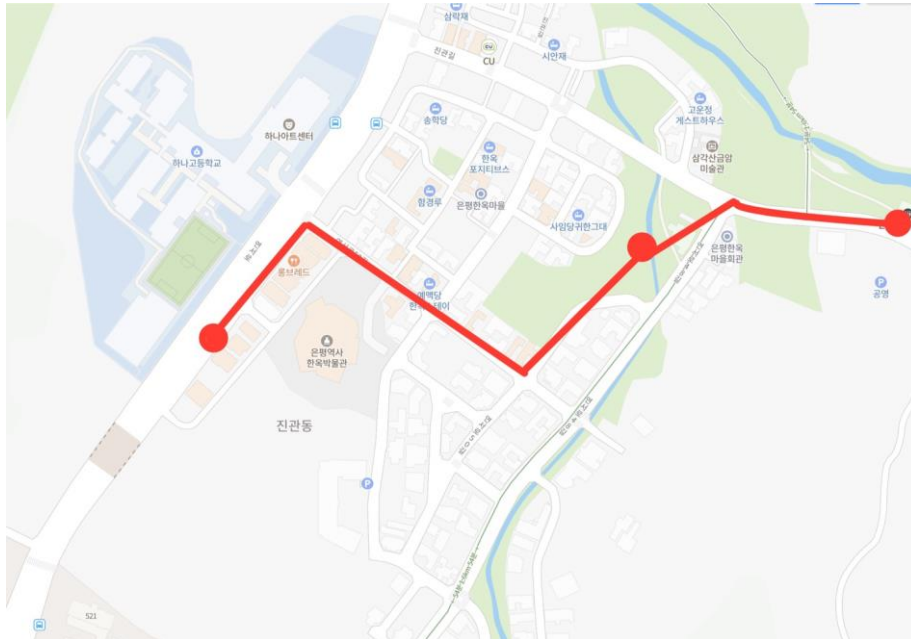
#### 4-9. 시각화

해당 코드를 통한 연구 결과, 100분 동안의 관광에서의 최적의 관광지 경로는  
진관사 40분 -> 터널 앞 음식점 60분



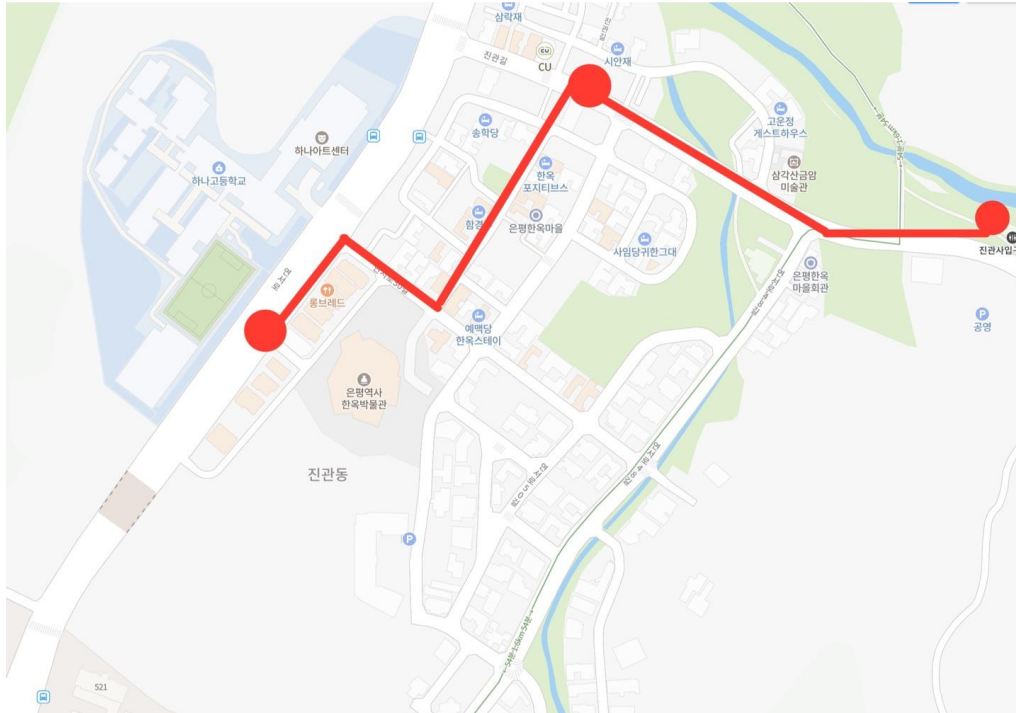
**figure1. Optimal tourist route for 100 minutes**

**130분 동안의 관광에서의 최적의 관광지 경로는  
진관사 50분 -> 산책로 20분 -> 터널 앞 음식점 60분**



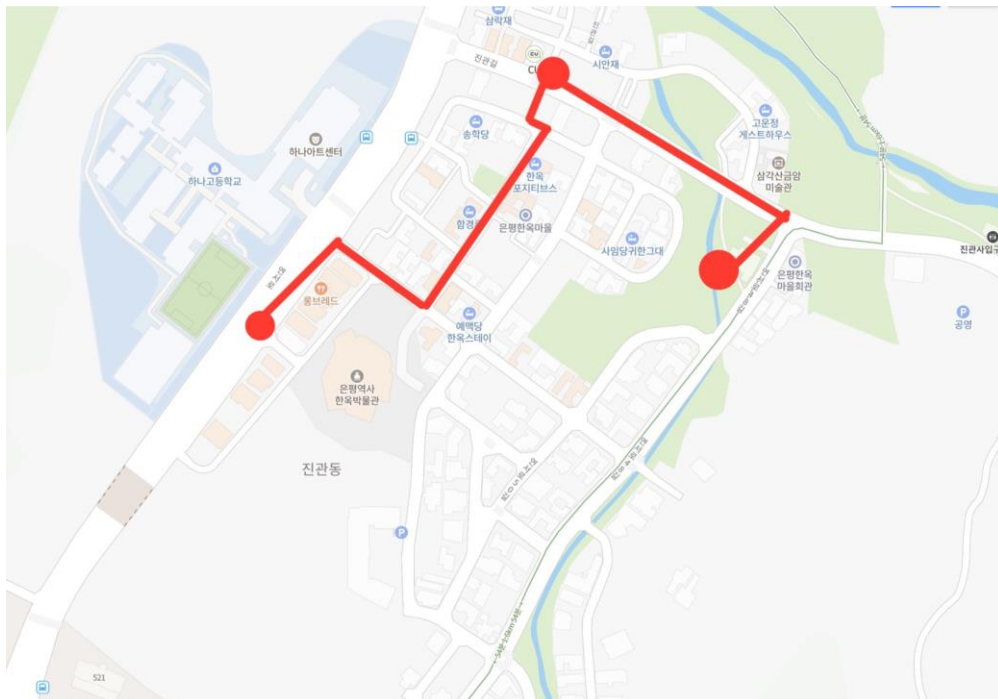
**figure2. Optimal tourist route for 130 minutes**

**150분 동안의 관광에서의 최적의 관광지 경로는  
진관사 30분 -> 상업 거리 60분 -> 터널 앞 음식점 60분**



**figure3. Optimal tourist route for 150 minutes**

170분 동안의 관광에서의 최적의 관광지 경로는  
산책로 50분 → 상업 거리 60분 → 터널 앞 음식점 60분



**figure4. Optimal tourist route for 170 minutes**

200분 동안의 관광에서의 최적의 관광지 경로는

산책로 30분 -> 물 다움 찾집 60분 -> 진관사 50분 -> 터널 앞 음식점 60분

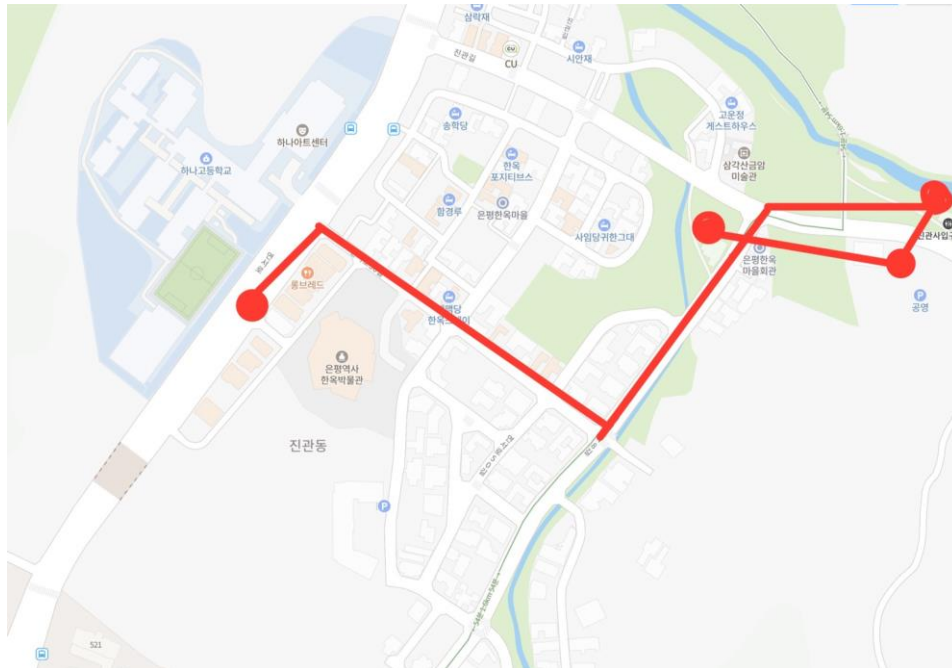


figure5. Optimal tourist route for 200 minutes

250분 동안의 관광에서의 최적의 관광지 경로는

산책로 20분 -> 물다움 찾집 60분 -> 진관사 60분 -> 상업거리 50분 -> 터널 앞 음식점 60분  
으로 계산되었다.

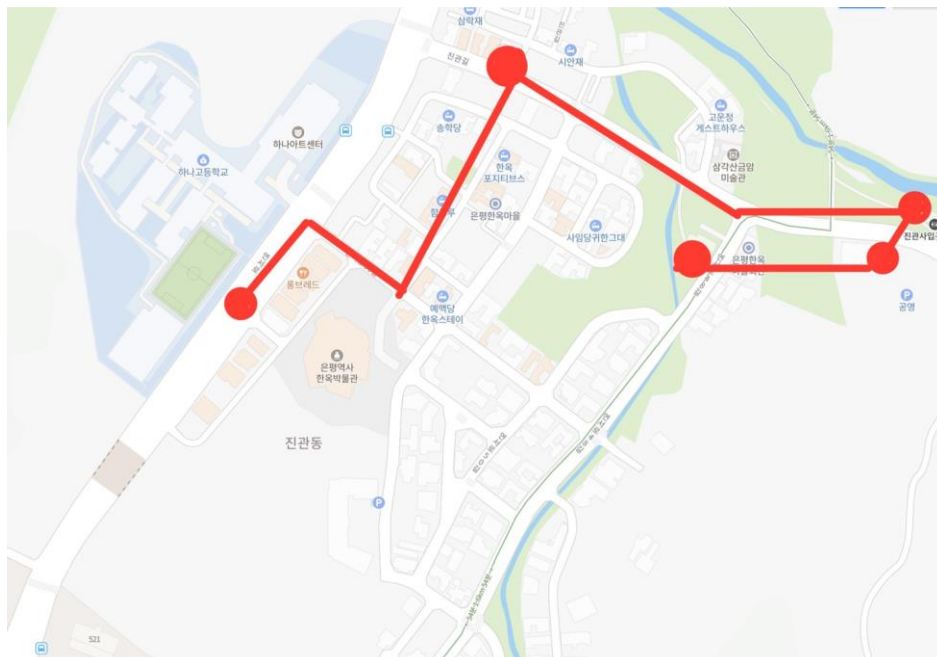


figure6. Optimal tourist route for 250 minutes

## 5. 결론 및 논의

본 연구는 마르코프 체인 기반 경로 최적화 모델을 통해 관광객의 이동 패턴을 분석하고, 선호도, 체류 시간, 이동 거리 등 다양한 요소를 종합적으로 고려하여 최적의 경로를 제시하고자 하였다. 연구 결과, 최적 경로는 음식점, 상업 거리, 진관사 등 특정 장소에 집중되는 경향을 보였으나, 한옥 마을 박물관과 같은 장소는 선택되지 않는 한계점이 드러났다. 이는 마르코프 행렬을 통해 관광지 경로를 계산할 때, 무조건적으로 가중치에 의거한 최적의 경로만을 도출하기에 관광 명소의 특성과 개개인의 관광 목적 및 선호도 등에 대해서 일일이 고려하기 힘든 점이 존재한다는 걸 보여준다.

본 연구는 기존 마르코프 체인의 문제점인, 체류 시간에 대한 데이터의 정보를 직접적으로, 그리고 세부적으로 적용할 수 없다는 문제점을 해결하기 위해, 같은 관광지의 시간에 따른 정점 분할을 활용하였다. 정점 분할을 10분의 단위 시간에 대해 진행함을 통해 기존의 문제를 해결하면서 조금 더 최적화된 결론에 도달할 수 있었다. 해당 방법은 앞으로 마르코프 체인을 활용한 다른 관광지 경로 분석에서도 활용할 수 있을 것이다.

본 연구의 주요 한계점 중 하나는 선호도를 평가하는 기준과 가중치가 단순화되어 관광객의 복합적인 선택 요인을 완전히 반영하지 못했다는 점이다. 또한, 이동 거리와 체류 시간의 영향이 과도하게 강조되면서 상대적으로 거리가 먼 관광 명소는 선택 가능성에서 배제되는 경향을 보였다. 마지막으로, 모델이 설계된 방식이 특정 조건에서 최적화된 결과를 도출하도록 제한되어 있어, 다양한 시나리오에 대한 적용 가능성이 낮다는 점도 한계로 지적할 수 있다.

이러한 한계점을 보완하기 위해 추후 연구에서는 선호도 평가 기준을 세분화하고, 방문지의 문화적 가치나 관광객의 개별 취향을 반영할 수 있는 다양한 가중치를 적용할 필요가 있다. 예를 들어, 관광객의 연령대, 여행 목적(휴식, 문화 체험, 쇼핑 등), 또는 관광지의 역사적·문화적 중요성을 반영한 가중치를 도입함으로써, 보다 정확하고 개인화된 관광 경로를 제시할 수 있을 것이다. 또한, 이동 거리나 시간 제약의 영향을 완화하기 위해, 기존의 마르코프 체인 모델에 유연성을 부여할 수 있는 방법을 모색해야 한다. 예를 들어, 이동 시간에 대한 가중치를 조정하거나, 제한된 시간 내에 최적 경로를 도출하는 다목적 최적화 기법을 도입하여, 더 다양한 상황에 적용 가능한 모델을 개발할 수 있을 것이다.

특히, 본 연구에서 대상에는 포함되었지만 최종적인 경로에서는 제외되었던 한옥 마을 박물관과 같은 관광지들이 경로에 포함될 수 있게, 거리나 시간 제약 외에도 관광지의 특성에 따른 선택을 반영할 수 있는 시뮬레이션을 진행하는 것이 중요하다. 이를 통해, 특정 지역이나 관광 명소를 배제하지 않고, 다양한 조건을 고려한 경로 최적화를 실현할 수 있을 것이다. 이러한 개선을 통해 모델의 적용 범위가 확장되고, 실제 관광 경로 설계에 있어 더욱 효율적이고 개인화된 솔루션을 제공할 수 있을 것으로 기대된다.

## 참고문헌

- Bellman, R. (1957). A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6(5), 679-684.
- Ghosh, M., & Sen, P. K. (1985). On the Use of Markov Chains in the Study of Visitor Behavior in Tourism Research. *Journal of Travel Research*, 23(2), 28-33.
- Li, Q., Zhang, Y., & Wong, W. C. (2015). A Markov Chain Approach for Modeling Tourist Movement Patterns in Heritage Sites. *Tourism Management*, 48(1), 293-300.
- Yang, Y., Fik, T., & Zhang, J. (2013). Modeling Sequential Tourist Stops with a Markov Chain Approach. *Annals of Tourism Research*, 41, 49-65.
- Huang, Z., Li, W., & Zhang, J. (2020). Optimizing Tourist Routes Using Graph Theory and Markov Chains. *Computational Management Science*, 17(3), 421-437.
- 김영석 (2012). 마르코프 체인을 이용한 관광객 이동 경로 모델링. *관광학연구*, 36(4), 123-134.
- 박지은 & 김민수 (2018). 마르코프 모델을 활용한 관광지 방문 패턴 분석. *관광연구저널*, 42(2), 56-68.
- 이경환, 최정윤 & 정하운 (2015). 관광객의 이동 패턴 최적화를 위한 마르코프 체인 모델 적용. *한국정보과학회논문지: 컴퓨터 시스템*, 32(1), 99-108.
- 김상호 & 이현주 (2019). 마르코프 체인을 활용한 관광지 경로 최적화 연구. *교통학회지*, 37(2), 45-58.
- 정수민, 박정훈 & 김재현 (2017). 관광지 이동 경로 분석을 위한 마르코프 모델의 응용. *지리학연구*, 59(3), 233-244.