

2024학년도 겨울 학기 Logistics Lab

연구참여 보고서

작성일자 : 2025.02.10

지도 교수: 김병인

지도 사수: 홍성균

참여 학생 : 고현민

목차

1. 서론
2. 연구 개요
 - 2.1 URM: Route Allocation Improvement Function
 - 2.2 TSP Estimation
3. 연구 내용
 - 3.1 URM: Route Allocation Improvement Function
 - 3.2 TSP Estimation
4. 결론 및 향후 연구 방향
 - 4.1 연구 결론
 - 4.2 향후 연구 방향
5. 후기

1. 서론

본 보고서는 6주간 진행된 연구인턴십에서 수행한 연구 과제에 대한 진행 과정과 주요 성과를 정리한 것이다. 필자는 지난해 여름학기에도 단기 연구에 참여한 경험이 있으나, 당시에 대학원 진학에 대한 구체적인 계획은 없었다. 그러나 그 이후로 약 반년 동안 자연스럽게 연구 활동에 대한 흥미가 생겼고, 이에 따라 이번 인턴십 기간 동안 다양한 연구 경험을 쌓고자 하였다.

본 인턴십은 크게 두 가지 분야로 나누어 진행되었으며, 각각의 과제는 서로 직접적인 연관성 없이 별개의 목표와 방법론을 가진다.

1. Route Allocation Improvement Function :

- 항만에 입항하는 선박의 선석 및 야드 배정 문제를 다루며, 기존 휴리스틱 방식의 한계를 보완하고 경로 최적화를 개선할 수 있는 함수를 구현하였다.

2. TSP(Traveling Salesman Problem) Estimation :

- 조합 최적화 문제의 고전적 예시인 TSP를 대상으로, 머신러닝 모델을 활용해 경로 길이를 추정하는 실험을 수행하였다. FNN, GNN, Pointer Network 등 다양한 딥러닝 방법을 적용하여 예측 성능을 비교·분석하였다.

이 보고서에서는 먼저 URM 과제의 추진 배경과 문제 정의, 그리고 경로 변경 함수를 활용하여 선석·야드 배정 효율성을 높인 연구 과정을 소개한다. 이후에는 TSP Estimation 과제를 통해 수행한 머신러닝 모델 구현과 학습 결과를 제시하고, 각 모델별 장단점 및 향후 개선 가능성을 논의한다.

마지막으로, 이번 인턴십을 통해 얻은 연구 경험과 느낀 점을 간략히 제시하며 본 보고서를 마무리하고자 한다. 이러한 인턴십 경험은 조합 최적화와 머신러닝이 접목되는 하이브리드 의사결정 지원의 가능성을 몸소 체감하는 기회가 되었다고 생각한다.

2. 연구 개요

2.1 URM: Route Allocation Improvement Function

URM 연구의 주요 목표는 항만에 도착하는 선박을 대상으로 최적의 선석과 야드 배정을 자동으로 결정하여 물류 흐름을 원활하게 하고, 선박의 대기 시간을 최소화하는 것이다. 기존에 각 선박의 야드 적치 계획은 배치가

가능함에도 불구하고 루트가 배치되지 않는 문제가 제기되어 왔다. 본 연구에서는 기존 경로 변경 배치 함수를 도입하여, 보다 유연하고 효율적인 배정 방안을 모색하였다.

2.2 TSP Estimation

TSP Estimation 연구에서는 머신러닝 기반 예측 모델을 활용하여, TSP 인스턴스에 대한 최적 경로 길이를 추정하였다. 이를 위해 FNN(Feed-forward Neural Network), GNN(Graph Neural Network), Pointer Network 세 가지 딥러닝 모델을 구현하고 비교하였다. 이들 모델은 입력 형태와 학습 방법, 그리고 예측 성능에서 차이를 보이므로, 각각의 특성과 한계를 면밀히 분석함으로써 조합 최적화 문제에 대한 머신러닝 모델의 가능성을 평가하고자 하였다.

3. 연구 내용

3.1 URM: Route Allocation Improvement Function

1) 연구 배경 및 필요성

- 기존의 경로 할당 방식은 모든 선석으로부터 야드까지의 루트 배정에 있어서 최대 루트 배정을 하지 못하였다. 이로 인해 야드 적치 계획에서의 손실이 발생함을 보여준다.
- 루트 최적 배정을 수행하면 선박의 대기 시간을 단축하고, 물류 흐름을 원활하게 하여 항만 운영 효율을 높일 수 있다.

2) 연구 방법

- 경로 정보 리스트 생성: 주어진 선박과 야드 정보를 기반으로, 가능한 모든 경로 정보를 전처리하여 리스트로 구성한다.
- 할당 가능 여부 검사: 현재 사용 중인 경로와의 중복 및 경로 유효성을 검사하여 실시간 할당 가능성을 평가한다.
- 경로 최적화 알고리즘 적용: 기존에 할당된 경로를 하나씩 변경하는 방법과, 두 개씩 동시에 변경하는 방법을 비교하여 최적의 할당 방식을 탐색한다.

Algorithm 1 Route Assignment Algorithm

```

for each route do
  if route is not assigned then
    if assignment is possible then
      Assign route
    end if
  end if
  if all routes are assigned then
    return
  end if
end for
Reset routes
for each previously assigned route do
  Modify one existing route
  for each route do
    if route is not assigned then
      if assignment is possible then
        Assign route
      end if
    end if
    if all routes are assigned then
      return
    end if
  end for
end for
Reset routes
if unassigned routes exist then
  for each previously assigned route do
    Modify two existing routes
    for each route do
      if route is not assigned then
        if assignment is possible then
          Assign route
        end if
      end if
      if all routes are assigned then
        return
      end if
    end for
  end for
end if
end if

```

[그림 1 pseudo code]

3) 연구 결과 및 분석

- 경로 변경 방식을 도입함으로써, 선박의 최적 선석·야드 배정이 가능함을 확인하였다.
- 비록 데이터셋의 한계로 인해 다양한 경우에 적용되는 결과를 만들어내지는 못하였지만, 추후 보다 복잡한 상황이 주어졌을 때 해당 함수를 적용함에 따라 루트 배정 최적화로 인한 개선 효과가 기대된다.

```

Route Info List:
1-> Berth : MW04, Yard : 0, Route Index : 6
2-> Berth : MW06, Yard : G, Route Index : 193
3-> Berth : MW01, Yard : - 1, Route Index : -1
4-> Berth : MW03, Yard : A, Route Index : 36
모든 루트가 성공적으로 할당되었습니다.
Route Info List:
1-> Berth : MW04, Yard : 0, Route Index : 6
2-> Berth : MW06, Yard : G, Route Index : 193
3-> Berth : MW01, Yard : K, Route Index : 39
4-> Berth : MW03, Yard : A, Route Index : 36

```

[그림 2 경로 할당 후 출력]

3.2 TSP Estimation

1) 연구 목표

- 주어진 도시들의 2차원 좌표나 그래프 정보를 입력받아, 경로 길이 혹은 경로 길이와 실제 예측 경로를 예측하는 머신러닝 모델을 구축한다.
- 추정된 길이의 정확도를 평가하여, 전통적인 휴리스틱 혹은 최적화 기법(본 연구에서는 Nearest Neighborhood + 2opt)과 비교했을 때 머신러닝 접근법의 유의미성을 검증한다.

2) 연구 방법

- FNN(Feed-forward Neural Network)

입력 데이터: 도시들의 (x, y) 좌표

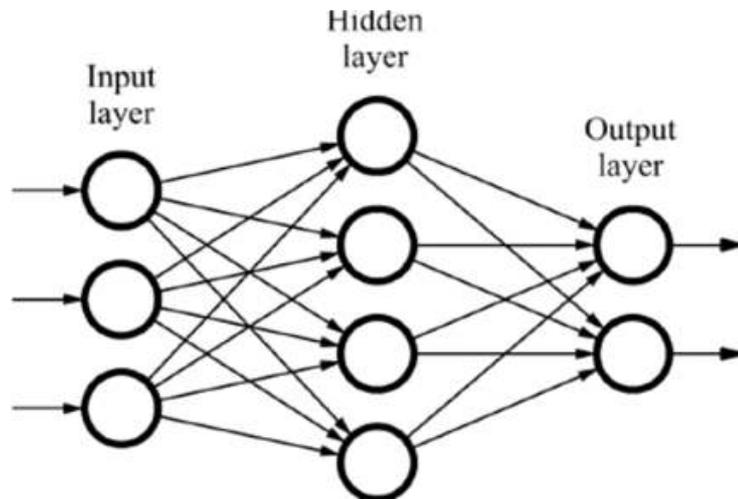
도시 수: 50~100개

학습 데이터: Nearest Neighbor + 2-opt 기법으로 사전에 해결한 10,000개 인스턴스

장점: 구조가 간단하고 빠른 학습 속도, 비교적 높은 예측 정확도

단점: 경로 자체를 제공하지 못하고, 단순히 길이만 예측이 가능

주요 특징: 대용량 데이터셋에 대해 비교적 손쉽게 확장이 가능하며, 최적화 문제의 목표치(경로 길이)를 회귀 분석으로 접근하기 수월함.



[그림 3 FNN 모델 개요도]

- GNN(Graph Neural Network)

입력 데이터: 도시들의 그래프 표현(노드와 엣지로 구조화)

도시 수: 10~20개 (학습 및 추론 시간 문제로 축소함)

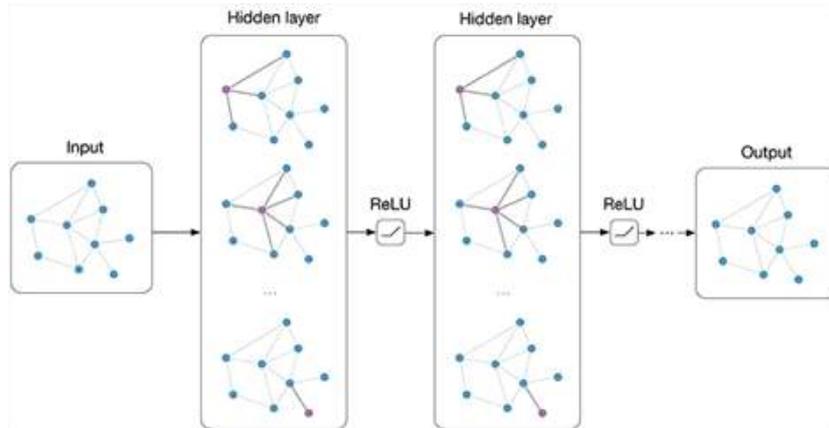
학습 데이터: 모드별 50개 인스턴스, BCE(Binary Cross-Entropy) 손실 및 회귀 손실을 혼합 적용

장점: 그래프 구조 정보를 직접 반영하여, 더 큰 규모의 네트워크에서도

잠재적으로 높은 효율성 기대

단점: 소규모 문제(10~20개)에서는 계산 오버헤드가 커서 휴리스틱 방법보다 시간을 더 소요할 수 있음(하지만 대규모 문제에서는 휴리스틱 방법보다 효율적일 것으로 기대함)

주요 특징: 노드 임베딩과 엣지 임베딩을 학습함으로써 도시 간 상관관계를 학습할 수 있으며, 그래프 기반 최적화 문제 일반화에 용이함.



[그림 4 GNN 모델 개요도]

- Pointer Network

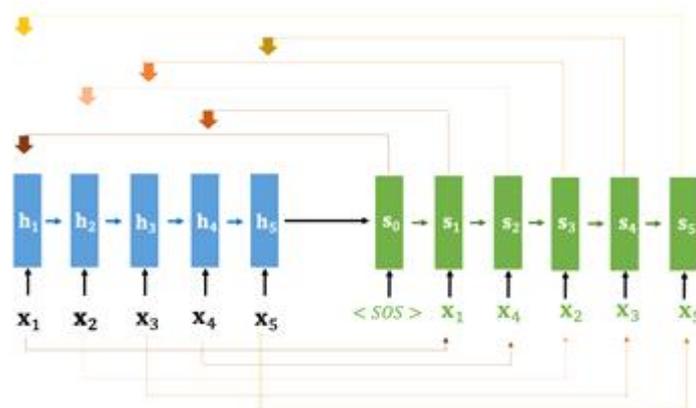
입력 데이터: 도시들의 2차원 좌표

도시 수: 10~20개 (학습 및 추론 시간 문제로 축소함)

학습 방법: 강화학습(Reward 기반)으로 진행하며, Reward 함수로는 (Nearest Neighbor + 2-opt 길이 - 예측 길이)를 사용

문제점: 학습 중 Reward가 수렴하지 않는 현상이 발생하여, 모델 구조 및 보상 함수 설정에 대한 추가 연구가 필요함

주요 특징: 이론적으로는 경로 순서까지 직접 학습하여, 단순 길이 예측을 넘어 실제 해를 생성할 수 있는 잠재력이 있음.



[그림 5 Pointer Network 모델 개요도]

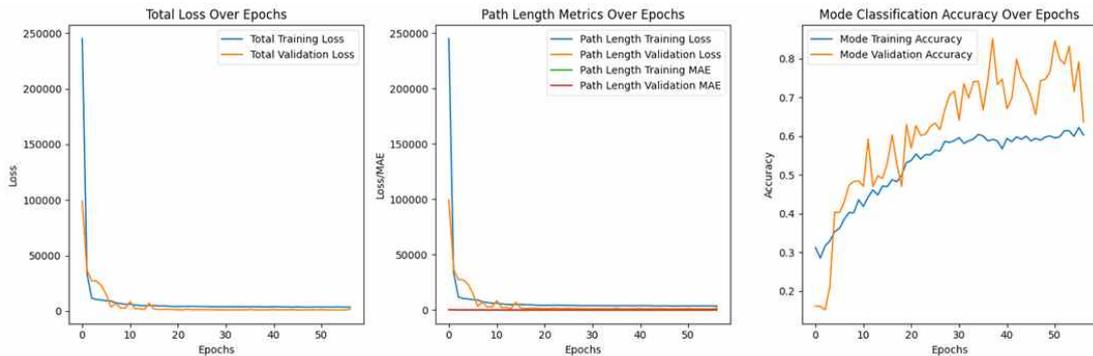
3) 연구 결과 및 분석

- FNN(Feed-forward Neural Network)

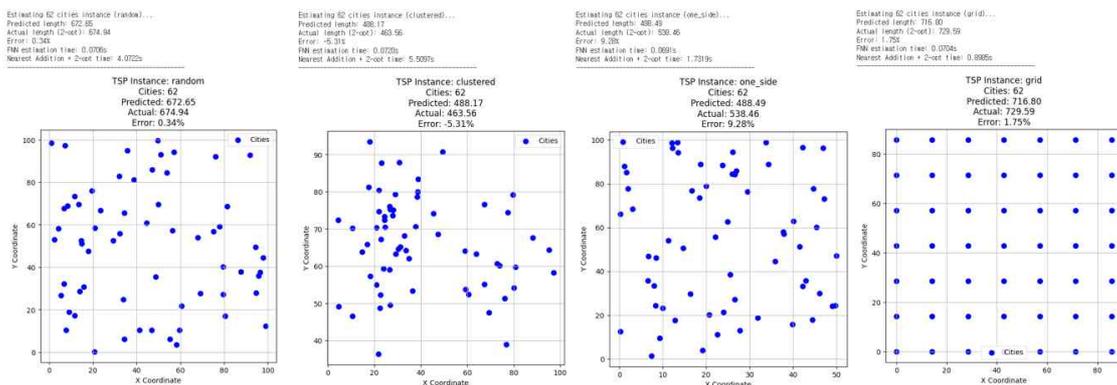
비교적 정확하게 경로 길이를 예측하였으며, 학습 과정도 안정적이었음.

인스턴스 규모가 커질수록 학습 데이터 확보가 용이하면, 더욱 강력한 일반화 성능을 보일 가능성이 있음.

다만, 최적 경로 자체를 산출하지 못한다는 단점이 있다. 또한 만약 예측 경로가 휴리스틱 경로보다 짧을 경우 실제로 해당 경로가 존재할지에 대해 확신할 수 없다는 한계가 존재한다.

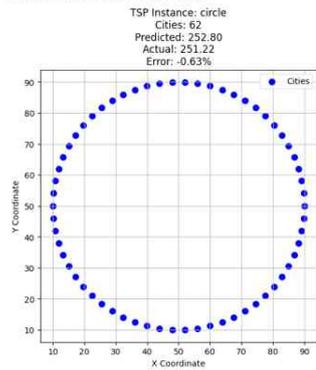


[그림 6 FNN 학습 결과]

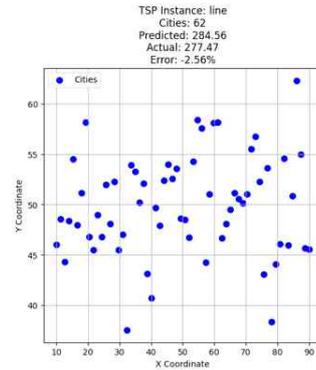


[그림 7 FNN 예측 결과 - 1]

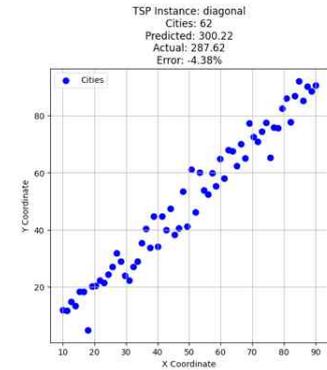
Estimating 62 cities instance (circle)...
 Predicted length: 252.80
 Actual length (2-opt): 251.22
 Error: -0.63%
 FNN estimation time: 0.0737s
 Nearest Addition + 2-opt time: 1.7445s



Estimating 62 cities instance (line)...
 Predicted length: 284.56
 Actual length (2-opt): 277.47
 Error: -2.56%
 FNN estimation time: 0.0696s
 Nearest Addition + 2-opt time: 3.2875s



Estimating 62 cities instance (diagonal)...
 Predicted length: 300.22
 Actual length (2-opt): 287.62
 Error: -4.38%
 FNN estimation time: 0.0939s
 Nearest Addition + 2-opt time: 3.2601s



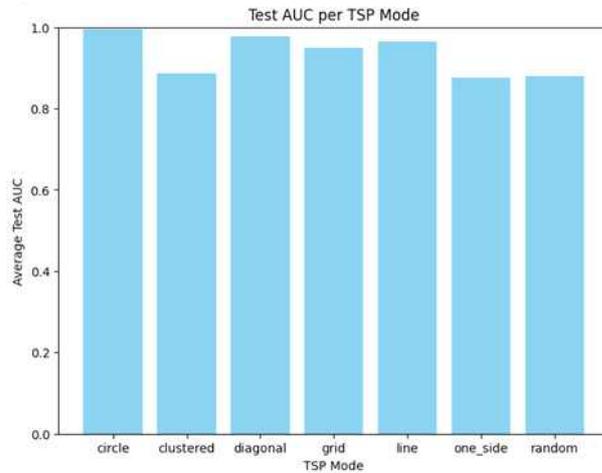
[그림 8 FNN 예측 결과 - 2]

- GNN(Graph Neural Network)

도시 수가 적은 실험 환경(10~20개)에서는 휴리스틱에 비해 학습 및 추론 시간이 더 길어지는 경향이 있었다.

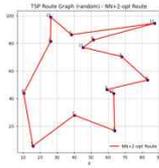
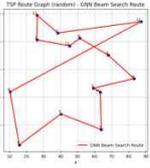
그러나 그래프 구조 이해를 바탕으로 실제 길이 추정 정확도가 향상되는 사례가 다수 관찰되었으며, 문제 규모가 커질수록 GNN의 이점이 두드러질 것으로 예상된다.

Test AUC per mode:
 random: 0.8810
 clustered: 0.8862
 one_side: 0.8766
 grid: 0.9487
 circle: 0.9944
 line: 0.9654
 diagonal: 0.9773

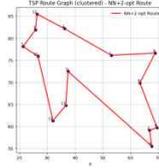
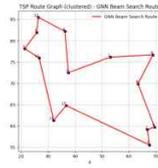


[그림 9 GNN 학습 결과]

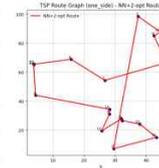
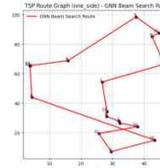
Estimating 14 cities instance (random)
 GNN Beam Search Decoded Path Length: 402.14
 Nearest Neighbor + 2-opt Path Length: 372.09
 Error between GNN (Beam Search) and NN2-opt: 8.08%
 Prediction Time: GNN = 0.0151 sec, NN2-opt = 0.0112 sec



Estimating 14 cities instance (clustered)
 GNN Beam Search Decoded Path Length: 147.74
 Nearest Neighbor + 2-opt Path Length: 149.91
 Error between GNN (Beam Search) and NN2-opt = 1.46%
 Prediction Time: GNN = 0.0153 sec, NN2-opt = 0.0062 sec

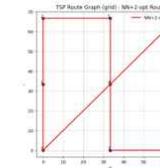
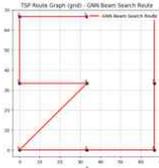


Estimating 18 cities instance (one_side)
 GNN Beam Search Decoded Path Length: 255.81
 Nearest Neighbor + 2-opt Path Length: 267.93
 Error between GNN (Beam Search) and NN2-opt = 4.75%
 Prediction Time: GNN = 0.0221 sec, NN2-opt = 0.0220 sec

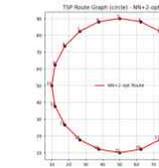
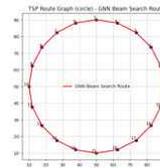


[그림 10 GNN 예측 결과 - 1]

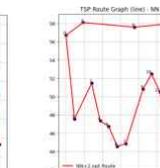
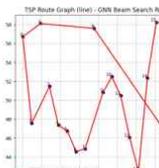
Estimating 14 cities instance (grid)
 GNN Beam Search Decoded Path Length: 313.81
 Nearest Neighbor + 2-opt Path Length: 300.95
 Error between GNN (Beam Search) and NN2-opt = 4.28%
 Prediction Time: GNN = 0.0090 sec, NN2-opt = 0.0062 sec



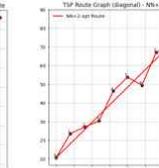
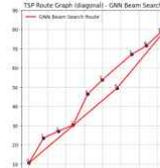
Estimating 20 cities instance (circle)
 GNN Beam Search Decoded Path Length: 250.30
 Nearest Neighbor + 2-opt Path Length: 250.30
 Error between GNN (Beam Search) and NN2-opt: 0.00%
 Prediction Time: GNN = 0.0301 sec, NN2-opt = 0.0108 sec



Estimating 18 cities instance (line)
 GNN Beam Search Decoded Path Length: 193.67
 Nearest Neighbor + 2-opt Path Length: 190.71
 Error between GNN (Beam Search) and NN2-opt: 1.55%
 Prediction Time: GNN = 0.0226 sec, NN2-opt = 0.0112 sec



Estimating 11 cities instance (diagonal)
 GNN Beam Search Decoded Path Length: 223.79
 Nearest Neighbor + 2-opt Path Length: 221.02
 Error between GNN (Beam Search) and NN2-opt: 1.26%
 Prediction Time: GNN = 0.0115 sec, NN2-opt = 0.0016 sec

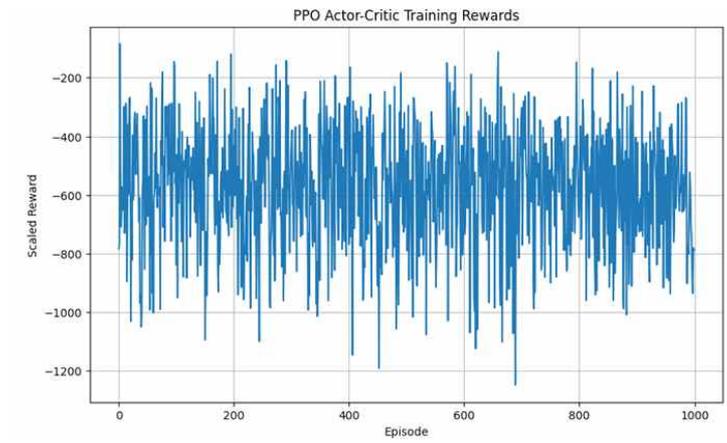


[그림 11 GNN 예측 결과 - 2]

- Pointer Network

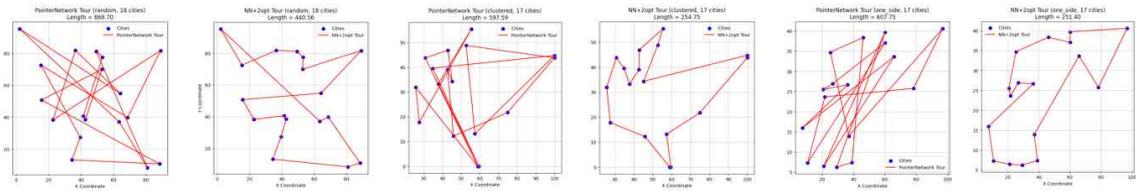
강화학습 기반 접근으로 경로 자체를 생성할 수 있다는 점에서 잠재력이 크지만, 학습 불안정성 및 Reward 수렴 문제가 해결 과제로 남았다.

보상 함수 설계와 네트워크 구조 변경을 통해, 학습 안정성을 높이는 추가 연구가 필요한 상황이다.



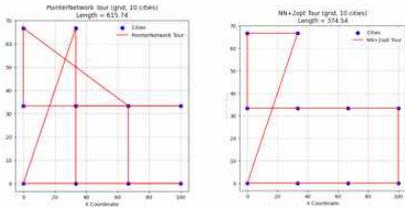
[그림 12 Pointer Network 학습 결과]

<p>Mode=random, #Cities=18 PointerNetwork Tour Length = 868.70 NN+2opt Tour Length = 440.56 Error = 97.18% PointerNet Inference Time = 0.1529s, NN+2opt Time = 0.0260s</p>	<p>Mode=clustered, #Cities=17 PointerNetwork Tour Length = 597.59 NN+2opt Tour Length = 254.75 Error = 134.58% PointerNet Inference Time = 0.1426s, NN+2opt Time = 0.0205s</p>	<p>Mode=one_side, #Cities=17 PointerNetwork Tour Length = 607.75 NN+2opt Tour Length = 251.40 Error = 141.74% PointerNet Inference Time = 0.1469s, NN+2opt Time = 0.0236s</p>
--	--	---

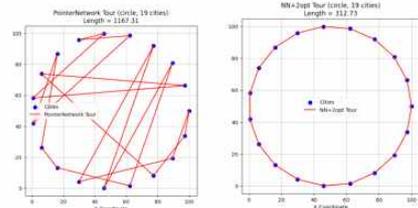


[그림 13 Pointer Network 예측 결과 - 1]

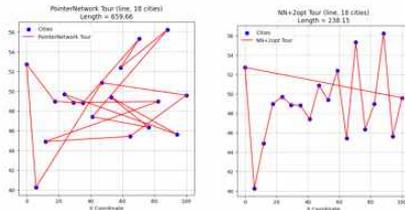
Mode=grid, #Cities=10
 PointerNetwork Tour Length = 615.74
 NN+2opt Tour Length = 374.54
 Error = 64.40%
 PointerNet Inference Time = 0.0097s, NN+2opt Time = 0.0021s



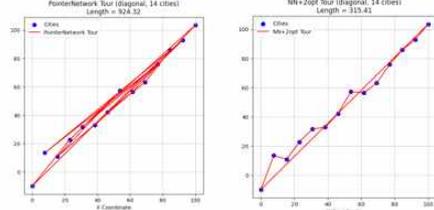
Mode=circle, #Cities=19
 PointerNetwork Tour Length = 1167.31
 NN+2opt Tour Length = 312.73
 Error = 273.26%
 PointerNet Inference Time = 0.1576s, NN+2opt Time = 0.0103s



Mode=line, #Cities=18
 PointerNetwork Tour Length = 659.66
 NN+2opt Tour Length = 238.15
 Error = 176.99%
 PointerNet Inference Time = 0.1588s, NN+2opt Time = 0.0087s



Mode=diagonal, #Cities=14
 PointerNetwork Tour Length = 924.32
 NN+2opt Tour Length = 315.41
 Error = 193.05%
 PointerNet Inference Time = 0.1273s, NN+2opt Time = 0.0040s



[그림 14 Pointer Network 예측 결과 - 2]

4. 결론 및 향후 연구 방향

4.1 연구 결론

- URM 연구를 통해, 기존 경로 할당 방식의 구조적 취약점을 개선할 수 있는 경로 변경 알고리즘을 적용함으로써 선석·야드 배정 효율성을 높일 수 있음을 보였다. 이를 통해 선박의 대기 시간을 단축하고 항만 운영의 전반적인 효율을 개선할 가능성을 확인하였다.
- TSP Estimation 연구에서는 머신러닝 기법을 통해 경로 길이를 예측하는 다양한 모델을 비교하였으며, 각각의 학습 구조 및 성능 특성을 정리할 수 있었다.
 - FNN은 간단한 구조임에도 불구하고 양호한 예측 성능을 보였으나, 해를 직접 제시하지 못한다는 제약이 있다.
 - GNN은 그래프 구조 활용으로 더 높은 확장성 및 정확도를 기대할 수 있으나, 아직 소규모 문제에서의 시간 효율성은 개선이 필요하다.
 - Pointer Network는 강화학습 기반으로 경로 자체를 생성할 수 있는 잠재력이 있지만, Reward 수렴 문제가 보완 과제로 남았다.

4.2 향후 연구 방향

1) TSP Estimation

- 더 큰 데이터셋 및 더 다양한 TSP 인스턴스(무작위 분포, 클러스터 분포, 실제 지형 기반 데이터 등)를 활용하여 모델의 일반화 성능을 심층적으로 검증한다.
- Pointer Network와 같은 강화학습 모델의 보상 함수 설계 및 학습 안정성을 개선하여, 단순 경로 길이 추정에 그치지 않고, 실제 최적 경로에 가까운 해를 직접 도출할 수 있는 방안을 모색한다.

5. 후기

이번 인턴십을 통해, 머신러닝 관점에서 조합 최적화 문제를 다루는 전반적인 과정을 체득할 수 있었다. 특히 TSP Estimation 실험을 통해 서로 다른 모델들의 구조적 특징과 성능 지표를 비교하며, 머신러닝 모델 설계에서 고려해야 할 요소(데이터 전처리, 손실 함수 정의, 학습 안정성 등)를 폭넓게 이해하게 되었다.

그러나 단기간(6주) 내에 모든 실험 시나리오를 충분히 진행하기에는 제약이 있었다. 예컨대 Pointer Network의 학습 안정성을 높이기 위한 파라미터 튜닝이나 보상 함수 개선에 더 많은 시간을 투자하지 못한 점이 아쉬움으로 남았다. 향후에는 강화학습과 그래프 구조학습을 보다 정교하게 결합하여, 실제 산업적 의사결정 문제에도 즉시 적용 가능한 수준의 모델을 개발해보고 싶다.

이번 인턴십의 경험은 데이터 분석과 최적화가 결합된 하이브리드 의사결정 지원 시스템이 산업 현장에서 얼마나 큰 가치를 가질 수 있는지 실감하게 해주었다. 앞으로도 본 연구를 더욱 확장하여, 머신러닝 기반 조합 최적화 모델이 가진 잠재력을 심도 있게 탐구하고, 실제 문제 해결로 이어질 수 있도록 노력할 계획이다.