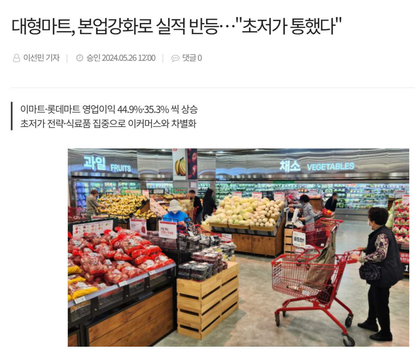
ShoppinGo: Multi-Policy NCH 기반 최적 쇼핑 경로 추천 서비스

2025-2 AJOU SOFTCON

팀 명 광교산 경로당 **지도교수** 김승운 교수님

팀 원 윤승규, 황규헌, 김민석, 박진호

개발동기 및 목적



Variables	Coefficient	Standard error	t-Value
Constant	. 53	.10	5.55**
Confusion of the store layout	.17	.02	6.93**
Spatial density	.18	.03	5.47**
Presence of salespersons	.22	.10	2.12*
Waiting time	.09	.03	3.27**
Choice overload	.08	.03	2.95**
Crowding	.13	.03	3.65**
R^2	.29		
Adjusted R^2	.28		

Notes: * $p < .05$, ** $p < .01$

- 대형마트(이마트, 롯데마트 등)는 '본업 강화' 및 '초저가 식료품' 전략을 통해 이커머스와의 경쟁 속에서도 **영업이익이 30~40%대 급증**하는 등 뚜렷한 실적 반등 존재
- 쇼핑 스트레스에 대한 연구 결과, 쇼핑 스트레스 유발 변수 중 **'매장 레이아웃에 대한 혼란 (Confusion of the store layout)'**은 (t-Value 6.93, $p < .01$)로 매우 높은 유의수준을 보이며 고객의 스트레스에 큰 영향을 미치는 핵심 원인임이 입증됨

> **쇼핑 효율성을 증진 시키는**
> **소비자 맞춤형 오프라인 쇼핑 안내 서비스 개발**

활용방안 및 기대효과

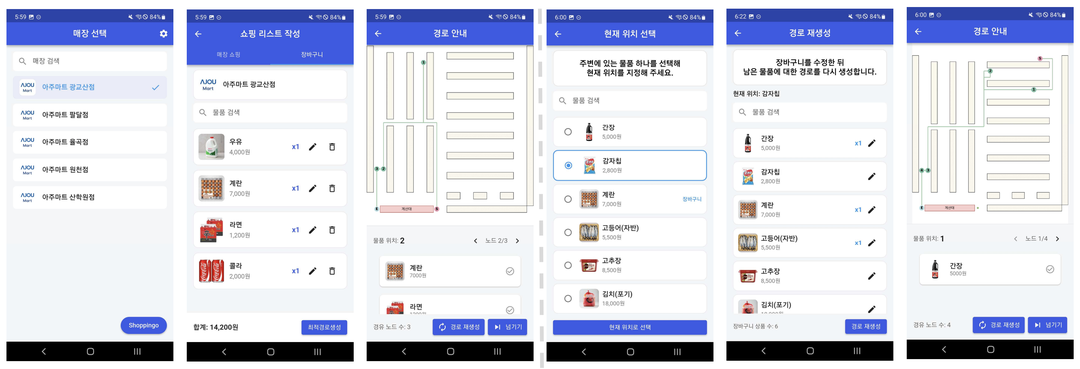
활용 방안	
실시간 최적 경로 제공	AI가 최적의 이동 경로를 **즉시 계산** → 사용자에게 내비게이션 제공
쇼핑 시간의 단축	불필요하고 체력 소모적인 동선을 제거 → 전체 쇼핑 시간을 단축
편리한 쇼핑 경험 제공	길을 헤매거나 상품 위치를 찾아다니는 **불편함 없는 편리한 쇼핑 지원**

ShoppinGo

쇼핑 시간 단축을 통한 **높은 편의성과 만족도 제공** + 개선된 쇼핑 경험으로 **쇼핑 스트레스 감소**

개발내용

최적 경로 생성 기능 / 경로 재생성 기능

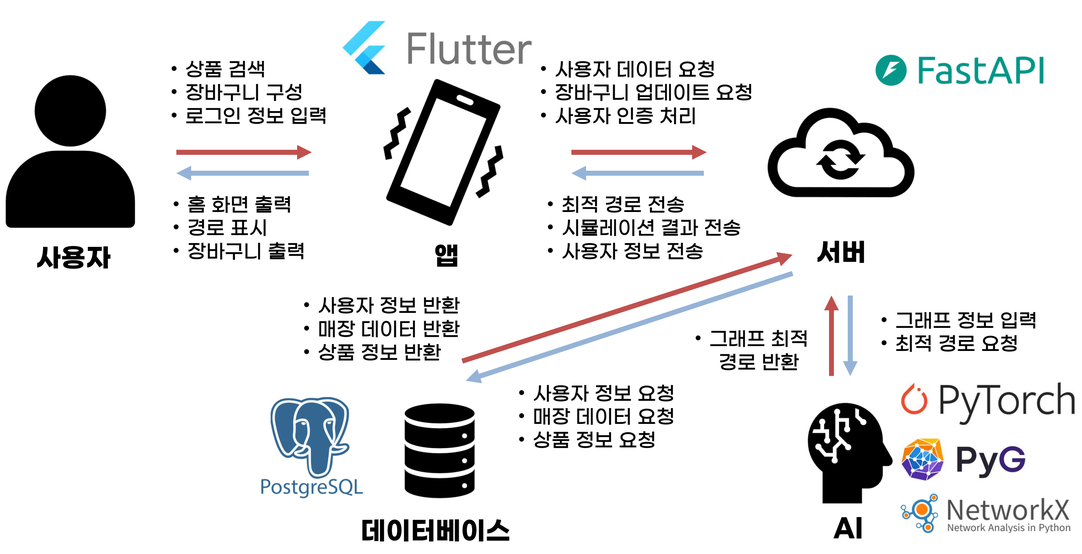


매장 선택 → 장바구니 생성 → 최적 경로 안내 → 주변 상품 선택 → 장바구니 수정 → 재생성 경로 안내

사용자 장바구니 기반 **최초 최적 경로 생성**

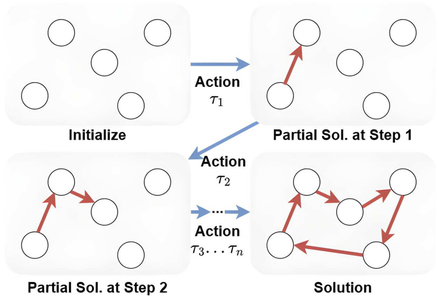
주변 물품/수정된 장바구니 기반 **최적 경로 재생성**

시스템 아키텍처



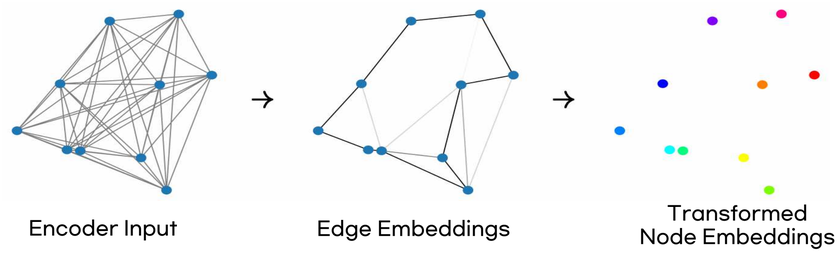
주요기술

DNN 기반 경로 탐색 (NCH: Neural Constructive Heuristic)



- **Neural Constructive Heuristic** : 신경망이 각 단계마다 결정을 내려 해를 구성해 나가는 방법
  - '경로 탐색 문제'를 해결하기 위해 **HELD (Heavy Encoder-Light Decoder) 구조의 DNN** 사용
  - **Autoregressive** → "End-to-End" 방법론
  - 각 단계 마다 DNN은 상태를 확인하고, 행동을 취함
  - Inference 속도가 매우 빠르며, 전통적인 휴리스틱 알고리즘보다 해의 품질이 더 좋다는 장점이 있음

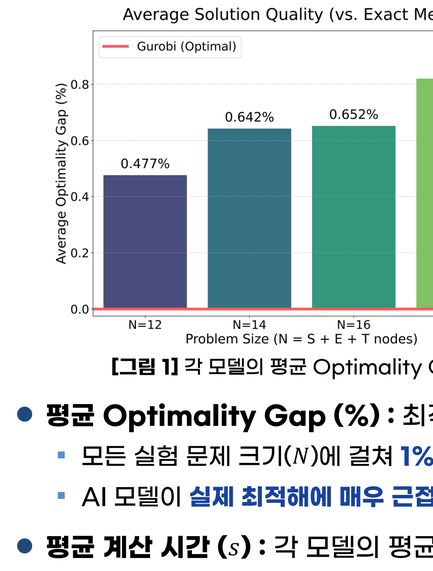
Encoder : 각 노드의 핵심 특징을 고차원 벡터(임베딩)로 변환



Encoder Input → Edge Embeddings → Transformed Node Embeddings

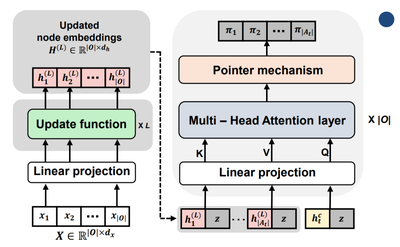
- **GREAT (Graph Edge Attention Network)** : 그래프 **'엣지'** 기반 인코더
  - 1. 인코더에게 노드와 모든 노드 간의 '최단 보행 거리' 정보가 있는 그래프를 입력
  - 2. 인코더는 엣지 간 정보 교환(Graph Attention)을 통해 **엣지 임베딩** 생성
  - 3. 생성된 엣지 임베딩을 종합하여 통해 각 노드 별 **최종 노드 임베딩** 생성

Decoder : 노드 임베딩 기반 '다음 방문 노드'를 순차적 선택



- **Ponterformer** : **멀티-포인터 쿼리** 기반 디코더
  - 1. 매 스텝 당 [①그래프 전체, ②최종 목적지, ③현재 위치, ④방문 경로]로 구성된 **'멀티-포인터 쿼리' 생성**
  - 2. 멀티-포인터 쿼리를 통해 **각 노드 별 적합도 계산**
  - 3. **물리적 거리를 고려**하여 적합도 점수 조정 (→ logits)
  - 4. 유효하지 않은 경로 마스킹 후, 유효한 노드들의 최종 logits를 softmax를 통해 **다음 노드의 확률 분포 생성**

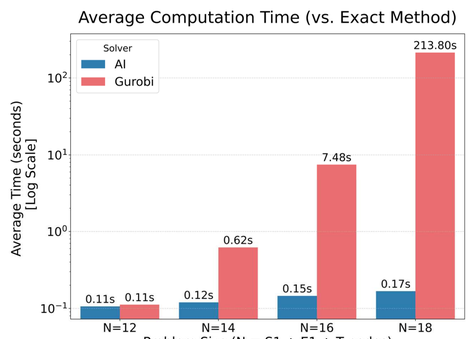
Multi-Policy & Self Imitation Learning (SIL)



- **Multi-Policy** : 샘플링된 **잠재변수(z)**를 디코더의 입력 임베딩에 Concat하여, 다양한 행동 정책을 표현
  - 하나의 문제 인스턴스에 대해 여러 정책을 병렬적으로 탐색하여 다양한 해를 도출 → **탐색 능력 강화 및 해 집합의 다양성 확보**
  - 각 z는 **특정 sub-distribution 문제에 특화**, 탐색 성능 향상
  - 하나의 DNN으로 여러 정책을 표현 → **계산 효율성 극대화**
- **Self Imitation Learning** : 모델이 **잘했던 행동을 모방**하도록 학습하는 방식
  - 모델 **스스로 정답지(pseudo-label)를 만들고**, 이를 **모방 학습**하는 모델의 성능을 지속적으로 자가 진화시키는 강화 학습(RL) 방법론 중 하나
  - 두 가지를 조합하여 **계산 속도와 최종 해의 품질을 모두 향상** → 복잡한 최적화 문제 해결 유리

결과 및 분석

[그림 1] 각 모델의 평균 Optimality Gap



[그림 2] 각 모델의 평균 실행 시간

- **평균 Optimality Gap (%) :** 최적해 대비 AI 모델 해의 근접도
  - 모든 실험 문제 크기(N)에 걸쳐 **1% 미만의 우수한 Average Optimality Gap**을 유지
  - AI 모델이 **실제 최적해에 매우 근접한 고품질의 해**를 성공적으로 도출했음을 입증
- **평균 계산 시간 (s) :** 각 모델의 평균 실행 시간
  - Gurobi 모델은 문제 크기(N)가 증가할수록 **계산 시간이 기하급수적으로 증가**
  - AI 모델은 문제 크기(N)가 커지더라도 **매우 짧은 시간 내에 해를 도출**
  - → **실시간 의사 결정 환경에 훨씬 더 적합**

오픈소스 URL

https://github.com/jinhopark-dot/Shoppingo