

# Learning to Delegate for Large-scale Vehicle Routing

Li, S., Yan, Z., & Wu, C. (2021). Learning to delegate for large-scale vehicle routing. *Advances in Neural Information Processing Systems*, 34, 26198-26211.

2025.04.22.

발표자 : 김성희

# 목차

---

1. 연구 소개 및 배경
2. 기존 연구의 한계점
3. Framework
4. Learning to Delegate
5. 실험 및 분석
6. 결론

# 1. 연구 배경

## 연구 배경

- 이 논문은 **VRP(Vehicle Routing Problem)**, 특히 **CVRP(용량 제한 차량 경로 문제)**를 다룸
- VRP는 택배, 음식배달, 공급망 관리 등에서 매우 널리 쓰이는 **조합 최적화 문제**
- 문제의 크기가 커질수록(예: 고객이 수천 명), **해를 구하는데 걸리는 시간이 기하급수적으로 늘어남**

## 연구 동기

- 모든 고객을 고려한 전체 최적화는 너무 무겁고 느리기 때문에, **가장 효과적으로 개선할 수 있는 ‘부분만’ 골라서 최적화**
- 전체를 다 풀 필요는 없고, **중요한 부분부터 먼저 해결**
- **중요한 부분을 딥러닝이 예측!**

## 연구 목적

- 대규모 VRP 문제를 빠르고 효율적으로 해결하는 새로운 학습 기반 프레임워크 제안
- 기존의 강력한 알고리즘(LKH-3 등)을 **서브솔버**로 사용하되, **딥러닝이 최적화할 부분(서브문제)을 선택**하여 계산 자원을 효율적으로 사용
- 다양한 문제 크기, 도시 분포, VRP 변형에도 **일반화(generalization)**가 잘 되는 구조로 구축

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론

## 2. 기존 연구의 한계점

### 전통 휴리스틱 기반 방법의 한계

대표 알고리즘	장점	한계점
<b>LKH-3</b> (Lin-Kernighan Heuristic)	- 해 품질이 매우 좋음 - 다양한 VRP 변형에도 적용 가능	- 계산 시간이 매우 길다 (예: 2000노드 CVRP에 1시간 이상 소요) - 대규모 문제에 실시간 적용 어려움
<b>HGS</b> (Hybrid Genetic Search)	- 유전 알고리즘 기반으로 빠르고 유연	- 구현 복잡도 높음 - 특정 상황에만 최적화되어 있음
<b>POPMUSIC</b> (Partial Optimization Metaheuristic)	- 부분 최적화 아이디어 기반	- 어느 부분을 고칠지에 대한 <b>전략이 고정됨 (휴리스틱에 의존)</b> - 학습이 아닌 룰 기반 선택이므로 유연성 부족

### 딥러닝 기반 최적화 기법의 한계

접근 방식	장점	한계점
<b>Pointer Network 기반 구성 모델</b> (예: Attention Model, POMO 등)	- 빠른 경로 생성 - 데이터 기반 학습 가능	- 작은 문제(100 노드 이하)에서는 좋지만 <b>문제 크기가 커지면 품질과 일반화 성능 급락</b>
<b>개선 기반 모델</b> (예: NeuRewriter)	- 기존 해를 반복적으로 수정 → 휴리스틱과 유사한 구조	- <b>연산량이 많고, RL 기반 학습이 매우 느림</b> - large-scale VRP에는 적용된 사례 거의 없음
<b>대규모 학습 접근</b> (예: Meta-Learning, Diffusion)	- 일반화 가능성 있음	- 학습/추론 비용이 매우 큼 - 학습 과정이 복잡하고 확장성이 낮음

기존 알고리즘은 정확하지만 너무 느리고, 딥러닝은 빠르지만 문제 크기에 약함  
특히, 어떤 부분을 우선적으로 최적화해야 가장 효과적인지 판단하는 기능은 없었음

### 3. An Iterative Framework for VRPs

#### 논문 핵심 컨셉

문제	→ 대규모 VRP를 한 번에 풀긴 어렵다
핵심 전략	→ 가장 개선 여지가 큰 subproblem부터 고치자
학습 모델	→ $f_\theta$ 가 각 subproblem의 예상 개선량을 예측
실행 방식	→ 매 반복마다 $f_\theta$ 가 하나의 subproblem을 선택 → LKH로 최적화
결과	→ 전체 해가 점진적으로 개선, 속도는 기존 대비 10~15배 빠름

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

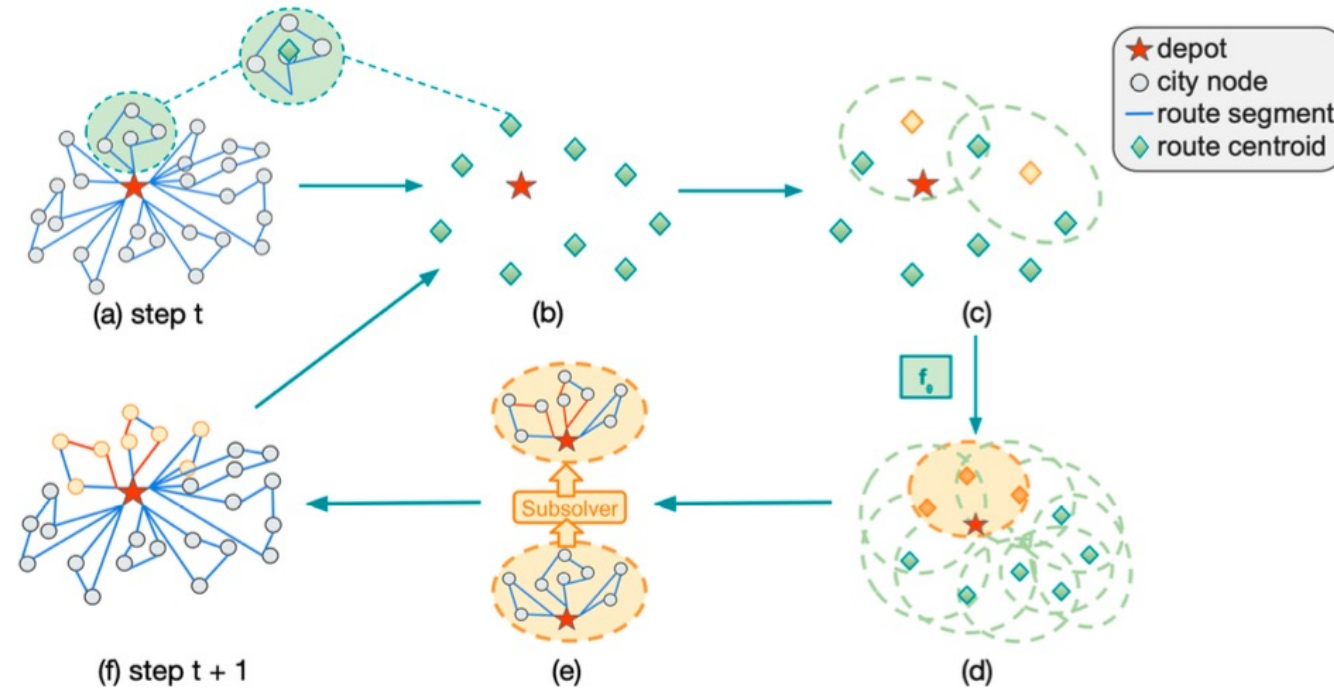
4. Learning to Delegate

5. 실험 및 분석

6. 결론

### 3. An Iterative Framework for VRPs

#### 반복 프레임워크



- (a) 전체 경로 X (각 원은 도시, 파란 선은 경로, 빨간 별은 depot)
- (b) 각 경로의 중심점을 계산 (중심좌표 구하기)
- (c) 중심점이 가까운 k개의 경로를 묶어 subproblem 후보 생성
- (d) 학습된  $f_\theta$ 가 가장 효과 클 subproblem S를 선택 (노란 원)
- (e) 선택된 S를 서브솔버로 풀어 새로운 경로  $X'_S$  생성
- (f) X의 해당 부분만  $X'_S$ 로 교체하고 다음 반복으로 진행

- 1. 연구 소개 및 배경
- 2. 기존 연구의 한계점
- 3. Framework
- 4. Learning to Delegate
- 5. 실험 및 분석
- 6. 결론

### 3. An Iterative Framework for VRPs

#### Learning to Delegate

---

**Algorithm 1:** Learning to Delegate

---

**Input:** Problem instance  $P$ , initialized solution  $X$ , subproblem selector  $f_\theta$ , subsolver  $\text{Subsolver}$ , number of steps  $T$ , parameter  $k$  denoting the size of subproblems

```
1 for Step  $t = 1: T$  do  
2    $\mathcal{S}_{k,\text{local}} \leftarrow \text{ConstructSubproblems}(P, X, k)$   
3    $S \leftarrow f_\theta(\mathcal{S}_{k,\text{local}})$   
4    $X'_S \leftarrow \text{Subsolver}(S)$   
5    $X \leftarrow X'_S \cup X_{P \setminus S}$   
6 end for
```

---

- $\mathcal{S}_{k,\text{local}}$  : 각 route  $r$ 에서  $k$ 개만 뽑음  $\rightarrow$  총 후보 수 =  $R$ 개
- VRP 같은 지리 기반 문제는 지역적 특성(locality)이 있음
- 실제로 서로 멀리 있는 도시들은 같은 경로에 포함될 확률이 낮음  
 $\rightarrow$  굳이 멀리 떨어진 것들을 묶어 풀 필요가 없음

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론

## 4. Learning to Delegate

### Subproblem 선택 기준 - 개선량(Improvement)

- Subproblem  $S$ 를 선택했을 때, 실제로 얼마나 개선이 일어나는지를 기준으로 평가
- 따라서, Subproblem  $S$ 의 개선 정도를 수치화하는 기준이 필요함
- 개선량 정의  $\rightarrow$  거리(비용)가 얼마나 줄었는지를 수치로 표현
- 기존 해:  $X_S$ , 서브솔버 결과:  $X'_S$

$$\delta(S) = c(X_S) - c(X'_S)$$

### 개선이 없을 경우의 처리 - hill-climbing + masking

- 실험 결과, 이전 해  $X_S$ 를 그대로 서브솔버에 주면  
 $\rightarrow$  로컬 최적에 갇혀 개선이 거의 없음 ( $\delta(S) \leq 0$ )
- 해결 방법:
  - $X_S$ 는 서브솔버에 제공하지 않고 새롭게 구성
  - $\delta(S) \leq 0$ 일 경우  $\rightarrow X'_S$ 는 버리고 기존 해  $X_S$  유지
  - 동일한  $S$ 를 다시 고르지 않도록 masking 처리

[초기 해]

- Route 1:  $A \rightarrow B \rightarrow C \rightarrow A$
- Route 2:  $A \rightarrow D \rightarrow E \rightarrow A$

[서브솔버에 넘긴 후 결과]

- Route 1':  $A \rightarrow D \rightarrow C \rightarrow A$
- Route 2':  $A \rightarrow B \rightarrow E \rightarrow A$

✓ 기존 경로 구조와 무관하게 새로 구성

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론



## 4. Learning to Delegate

### Subproblem 선택 기준 공식

- 이 논문의 목표는, 지금 당장 가장 개선 효과가 클 subproblem을 선택하는 것
- 강력한 서브솔버(LKH)는 내부적으로 다양한 로우레벨 최적화를 수행하기 때문에 subproblem만 잘 골라도 전체 해의 품질이 크게 향상됨

$$\arg \max_S c(X_S) - f_{\theta}(S)$$

### Ground-truth labels

- 학습을 위해 필요한 정답(레이블)  $\delta(S)$ 를 어떻게 만드나?
- 학습 데이터 생성 시, subproblem  $S$ 마다 실제로 LKH-3을 실행
- 그 결과로 실제 개선량  $\delta(S)$ 를 얻어 회귀 모델의 정답으로 사용

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

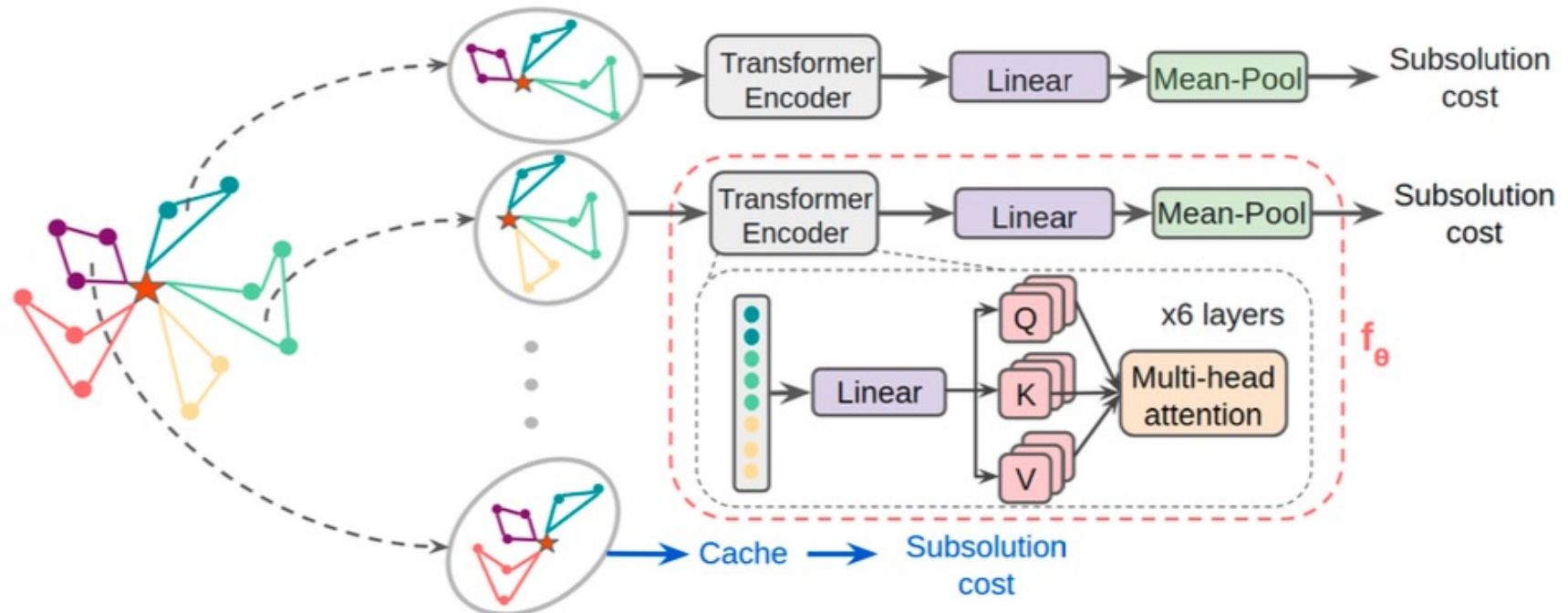
5. 실험 및 분석

6. 결론

## 4. Learning to Delegate

### Network architecture

- Transformer Encoder 기반 회귀 모델
- 입력: 도시의 상대 좌표  $(x, y)$ , 수요  $(d)$
- 출력: 스칼라 값 (예측된 개선 후 비용)
- 손실함수: Huber Loss (안정성 확보)



## 4. Learning to Delegate

### Offline Phase: 학습 (모델 훈련, 사전 준비 단계)

VRP 문제 생성 ( $N=500, 1000, 2000$ )

초기 해  $X_0$  생성  
(도시 분할 후 LKH-3로 각 부분을 짧게 최적화)

Subproblem 후보 생성  
(각 경로 중심점 기준, 가까운  $k$ 개 route 묶음  $\rightarrow S \in S_{k,local}$ )

LKH-3 실행  $\rightarrow$  개선량 계산  
(각  $S$ 에 대해  $X_S, X'_S, \delta(S) = c(X_S) - c(X'_S)$ )

학습 데이터 구성  
(입력: 도시 위치+수요 / 출력: 개선 후 비용  $c(X'_S)$ )

$f_\theta$  학습 (Transformer, Huber Loss)

모델 저장 (정제된  $f_\theta$  완성)

### Online Phase: 추론 및 반복 실행 (실제 문제 적용)

신규 VRP 인스턴스 입력 ( $N=500 \sim 3000$ )

초기 해  $X_0$  빠르게 생성

$T$ 번 반복

후보  $S \in S_{k,local}$  생성

$f_\theta(S)$ 로 개선량 예측

가장 좋은  $S$  선택

선택된  $S$ 만 LKH 실행  $\rightarrow X'_S$  생성

전체 해 업데이트:  $X \leftarrow X'_S \cup X_{P \setminus S}$

최종 해 완성

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

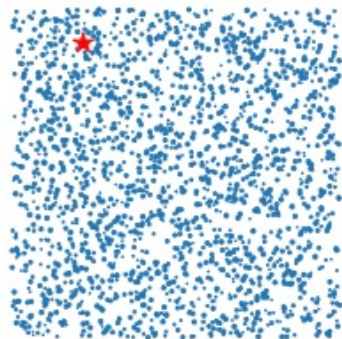
5. 실험 및 분석

6. 결론

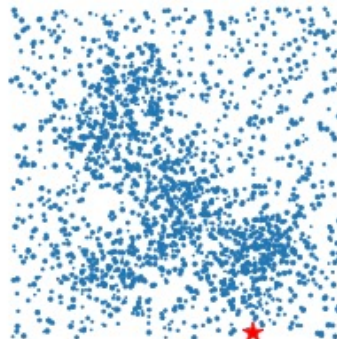
## 5. 실험 및 분석

다양한 조건에서 얼마나 잘 작동하는지 다각도로 검증

Uniform distribution



Clustered distributions



Real-world



Figure 3: **Instances from  $N = 2000$  CVRP distributions.** From left to right: instance from uniform, mixed ( $n_c = 7$  cluster centers), clustered ( $n_c = 3$  cluster centers), and real-world. The red star is the location of the depot, while blue dots are cities sized proportional to demand.

1. Uniform distribution : 2차원 평면에 **균일한 분포**에서도 기존 대비 성능 우위를 보이는지 확인
2. Clustered distributions : 모델이 **지역적 집중도(locality)**를 잘 활용하는지 검증
3. Out-of-distribution : 모델의 **범용성(generalization)**을 검증하는 실험
4. VRP variants(문제 유형 확장) : 제안한 프레임워크의 **확장성 & 실용성**을 검증
5. VRP solvers : 다른 최적화 도구(알고리즘)와도 **유연하게 연동**되는지 실험

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론

## 5. 실험 및 분석

### Uniform 분포에서의 실험 결과

Method	N = 500		N = 1000		N = 2000	
	Cost	Time	Cost	Time	Cost	Time
LKH-3 (95%)	62.00	4.4min	120.02	18min	234.89	52min
LKH-3 (30k)	61.87	30min	119.88	77min	234.65	149min
OR Tools	65.59	15min	126.52	15min	244.65	15min
AM sampling	69.08	4.70s	151.01	17.40s	356.69	32.29s
AM greedy	68.58	25ms	142.84	56ms	307.86	147ms
NeuRewriter	73.60	58s	136.29	2.3min	257.61	8.1min
Random	61.99	71s (3.8x)	120.02	3.2min (5.5x)	234.88	6.4min (8.0x)
Count-based	61.99	59s (4.5x)	120.02	2.1min (8.2x)	234.88	5.3min (10x)
Max Min	61.99	59s (4.5x)	120.02	2.5min (7.0x)	234.89	5.2min (10x)
Ours (Short)	61.99	38s (7.0x)	119.87	1.5min (12x)	234.89	3.4min (15x)
Ours (Long)	<b>61.70</b>	76s	<b>119.55</b>	3.0min	<b>233.86</b>	6.8min

→ Ours (Short)는 LKH-3 (95%) 수준의 해를 낸다 (때로는 더 좋음)

→ LKH-3보다 7~15배 빠름

→ Ours (Long)은 LKH-3 (30k)보다 더 좋은 해도 가능

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

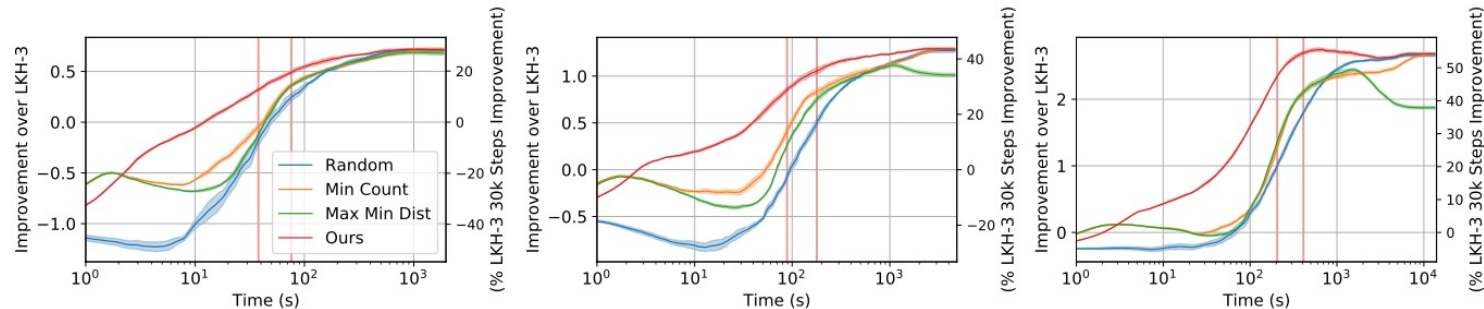
4. Learning to Delegate

5. 실험 및 분석

6. 결론

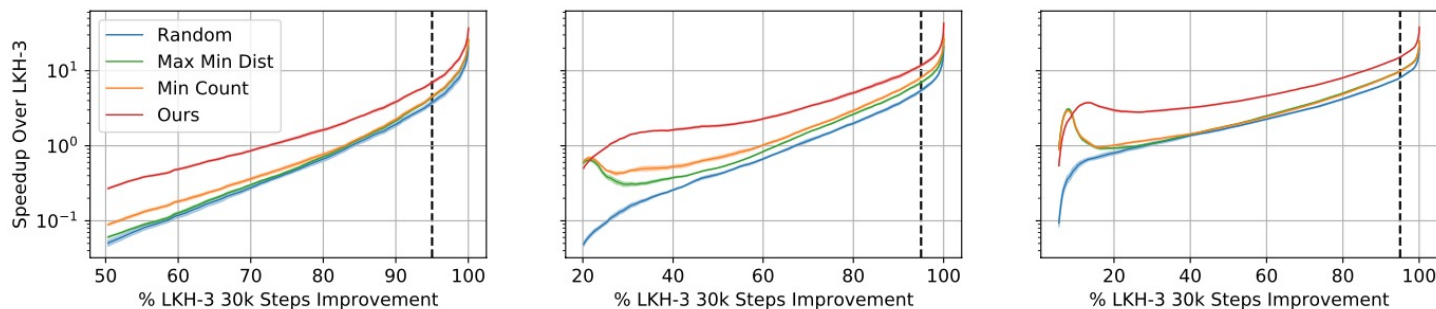
## 5. 실험 및 분석

시간 vs 개선량 : 같은 시간에 누가 더 많이 개선했나?



→ Ours는 짧은 시간 안에 빠르게 좋은 해에 도달하며,  
전통적인 전체 최적화 방식보다 훨씬 빠른 수렴 속도를 보임

품질 vs 시간 : 같은 품질을 누가 더 빨리 도달하나?



→ Ours는 동일한 품질의 해를 기존 LKH-3보다 훨씬 빠르게 달성할 수 있으며,  
특히 문제 크기가 커질수록 speedup 효과가 더 커짐

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

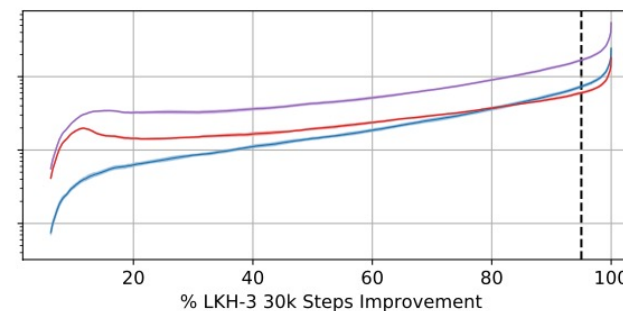
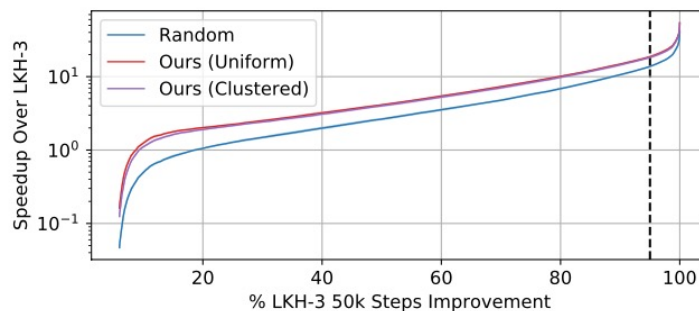
5. 실험 및 분석

6. 결론



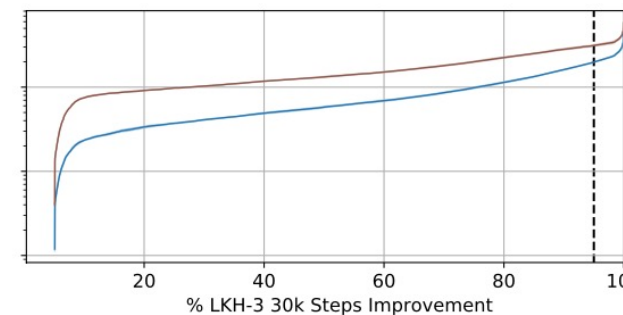
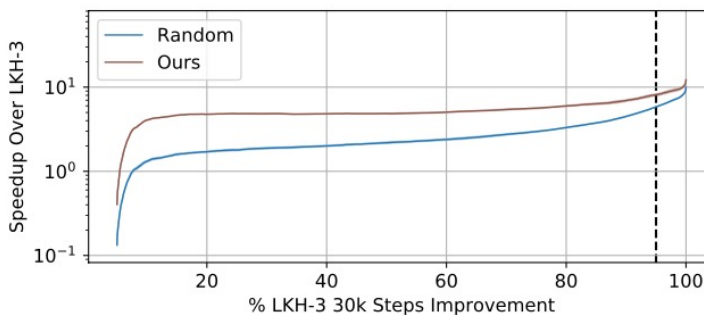
## 5. 실험 및 분석

### 학습에 사용하지 않은 분포(Out-of-distribution)에서 범용성 평가



- 두 모델 모두 finetuning 없이도 높은 speedup 달성
- Ours (Clustered)는 clustered 구조에 가까운 real-world에 더 잘 작동함
- 실제 지도 기반 VRP에서도 학습된 모델이 강한 일반화 성능을 보이며 빠르게 좋은 해에 도달

### CVRP 이외의 VRP 변형 문제에도 모델이 확장 가능한지 일반화 평가



- 우리 모델은 CVRP 외에도 변형된 VRP 문제에서도 LKH-3보다 빠르게 목표 품질에 도달

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

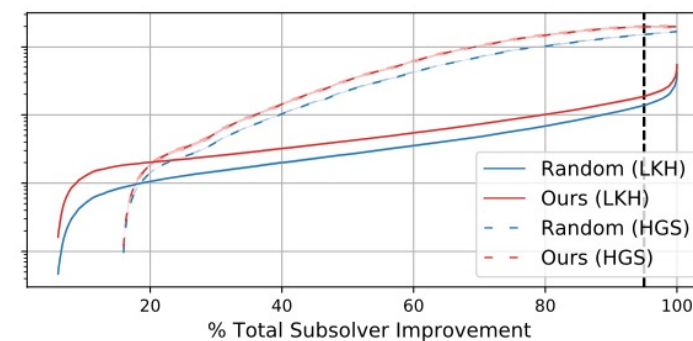
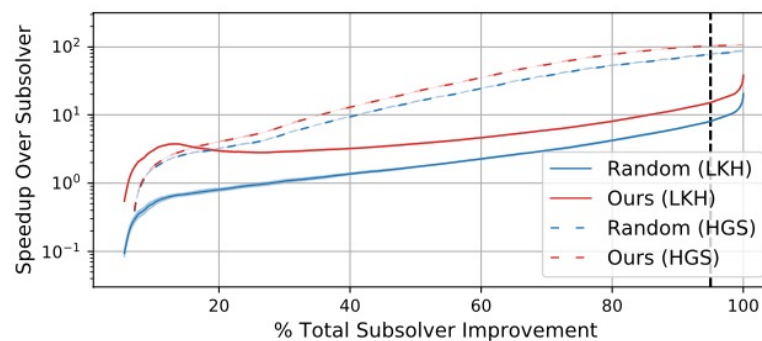
4. Learning to Delegate

5. 실험 및 분석

6. 결론

## 5. 실험 및 분석

### 제안한 프레임워크가 LKH-3뿐 아니라 HGS와도 잘 결합되는지 평가



- Ours + HGS가 더 빠른 경우도 존재 → 프레임워크가 subsolver에 독립적이고 유연함을 보여줌
- 특히 문제 크기가 클수록 speedup 차이 명확해짐



## 6. 결론

### 주요 연구 성과

- 제안한 학습 기반 프레임워크는 대규모 VRP 문제에서 subproblem을 효율적으로 선택해 기존 최적화 알고리즘을 빠르게 보조하는 구조
- 공간적 지역성(spatial locality)을 활용해 subproblem 선택 공간을 줄이고 학습 가능하게 함
- 최대 N=3000 문제까지, 기존 솔버 대비 1.5~2배 이상의 speedup + 높은 해 품질 확보
- 강화학습 없이도, 적당한 품질의 label로 지도학습만으로 높은 성능 가능함을 입증
- 다양한 VRP 분포, 문제 변형, 서브솔버에도 일반화 가능 → 조합최적화 전반에 확장 가능성

### 향후 연구 방향

- 공간 지역성이 있는 다른 조합최적화 문제에도 적용 가능

1. 연구 소개 및 배경

2. 기존 연구의 한계점

3. Framework

4. Learning to Delegate

5. 실험 및 분석

6. 결론

**Q&A**