

Learning to Delegate for Large-scale Vehicle Routing

Sirui Li, Zhongxia Yan, Cathy Wu Advances in Neural Information Processing Systems 34 (NeurIPS 2021)

理論談話会#08

B4一山遼

概要

- たくさんの場所を回る配送ルート(CVRP)の問題
- これまでは強いソルバーを使っても都市の数が増えると時間がかかっていた
- 機械学習を使ったアプローチはあったが速さと性能のトレードオフだった
- 学習して委任する
- 大きな問題を分割し、分割された問題を解くのは既存のサブソルバーに委任する、どの問題を解かせたら一番性能が良くなるのかという判断を機械学習でやる

新規性有用性信頼性

- 新規性
 - Transformerの使い方
 - いままでは強化学習のアプローチが大半だった
 - 機械学習で直接解くのではないのが斬新
- 有用性
 - スケーラビリティ
 - 大規模VRPでも高速かつ解の質がよい
- 信頼性
 - 広域な実験
 - 人工的な分布から現実世界の分布まで
 - コードの公開
 - GitHubで公開されている



目次

1&2. Introduction, CVRP

3. Related Work

4. An Iterative Framework for VRPs

5. Learning to Delegate

6. Experiments and Analysis

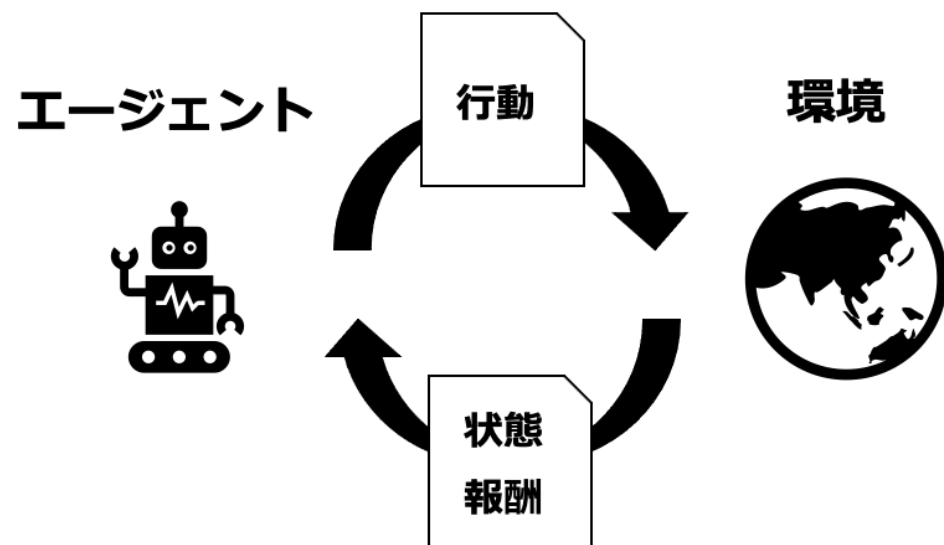
7. Conclusion



研究背景

VRP においては、車両の経路を最適化するための行動を学習するという観点から、強化学習が適用されることが多かった

強化学習的アプローチ



ルートを試行錯誤して、どのような行動をすればコストを下げられるかを学んでいくやり方。

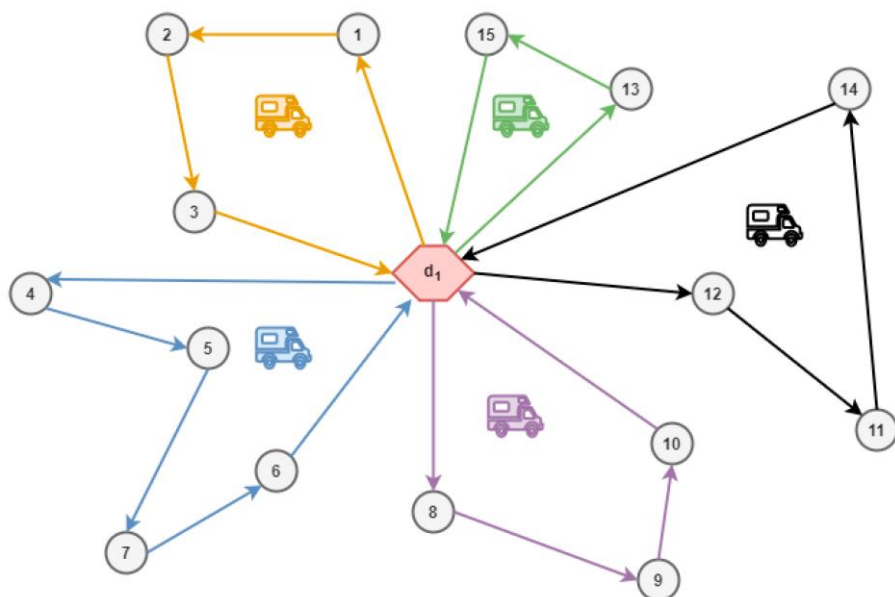
他の強化学習的アプローチは囲碁・将棋AIや掃除ロボットなど

1&2. Introduction, CVRP

導入とCVRPについて

VRPについて

VRPについて



概要

地理空間上に分布するノード（都市、客等）に対し、単一ないしは複数のデポ（車庫）から出発し、すべてのノードを訪問した後に車庫に戻る経路のコストを最小化させる

現実世界での応用

ゴミ収集
宅配便の集荷、配送
スクールバス
オンデマンドバス

課題

NP困難であるため、ノードの数が増えると計算量が爆発的に増加し、求解が困難になる。

CVRPについて

CVRPとはCapacitated Vehicle Routing Problemsの略であり、車両に容量制約がついたVRPである。

定式化

目的関数

$$\min \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}$$

- $V = \{0, 1, 2, \dots, n\}$: ノード集合 (0はデポ、1~nは顧客)
- c_{ij} : ノード i から j への移動コスト (距離や時間など)
- d_i : 顧客 i の需要量 ($d_0 = 0$)
- Q : 各車両の最大積載量
- K : 利用可能な車両数
- $x_{ij} \in \{0, 1\}, i \neq j$: 経路において i から j に移動するかどうか
- u_i : ノード i に到達したときの累積需要量 (積載量管理用)

制約条件

1. ノードには一度だけ訪問

$$\sum_{j=0}^n x_{ij} = 1 \quad \sum_{j=0}^n x_{ji} = 1 \quad \forall i, j = 1, 2, \dots, n$$

2. デポから出発・帰還

$$\sum_{j=1}^n x_{0j} \leq K, \quad \sum_{i=1}^n x_{i0} \leq K$$

3. サブツアー除去制約 (積載条件も)

$$u_i - u_j + Qx_{ij} \leq Q - d_j \quad \forall i \neq j, \quad i, j \in \{1, 2, \dots, n\}$$
$$d_i \leq u_i \leq Q \quad \forall i = 1, 2, \dots, n$$
$$u_0 = 0$$

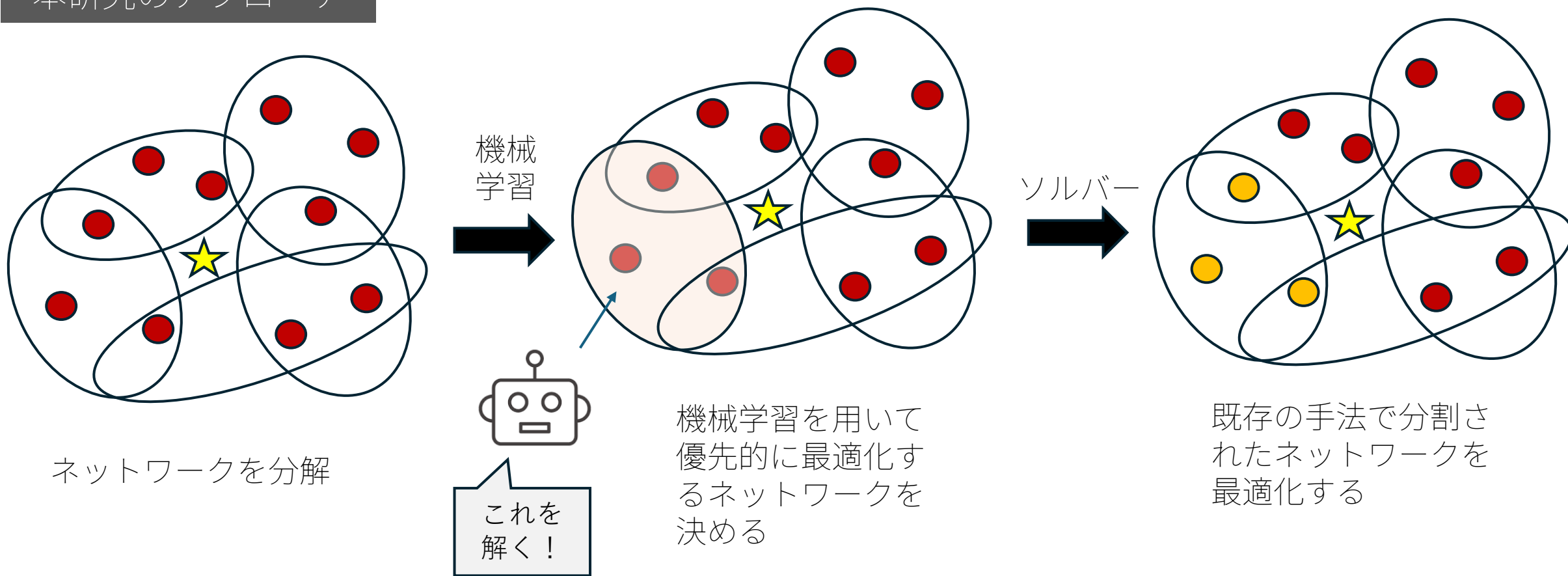
既存の機械学習を使ったVRPへのアプローチ

機械学習を使ったVRPへのアプローチはあったが……

- 都市の数が100以下でないと解の質が保障されていない
- 都市の数が200以上の研究がそもそもほとんどない
- 強化学習や深層強化学習の手法が取られていた

Learning-to-delegate ~分割して委ねる~

本研究のアプローチ



3. Related Work

関連研究

Classical methods

VRPソルバー

LKH-3

- 様々なVRPへの汎用性
- エッジを入れ替えて新しいルートを作る
 - この操作を繰り返す

HGS

- CVRPに特化している
- ハイブリッド遺伝的アルゴリズムとローカルリサーチで最適化をする
- 1000までは素晴らしい精度

最適化手法

POPMUSIC

- Partial OPTimization Metaheuristic Under Specialの略
 - “OP”timizationは無理がないか…？
- 問題の分割→部分問題の選択→部分領域に局所探索アルゴリズムを適用→全体解への統合
- 本研究の手法はPOPMUSICからインスピレーションを受けた

Deep learning method

1) 構築手法

自己回帰型のモデルが直接解を出力する
Pointer Networkなど

2) 改善手法

解に対して反復的にローカルな更新を行う



$N \leq 100$ においてLKH-3を上回った！！



大規模VRPでは軌道の収集が非常に高コストになってしまう…

Scaling up learning methods

- NP困難グラフ問題の学習済みネットワークのスケーリングの研究が進められている
- 既往研究はいくつかあるものの、本研究ではVRPの制約と構造にとって自然なスキームを提示する
- 大規模なVRPを分解するために学習を組み込む試みはあるが、本研究より効果的なものはなかった。

4. An Iterative Framework for VRPs

VRPへの反復フレームワーク

発想

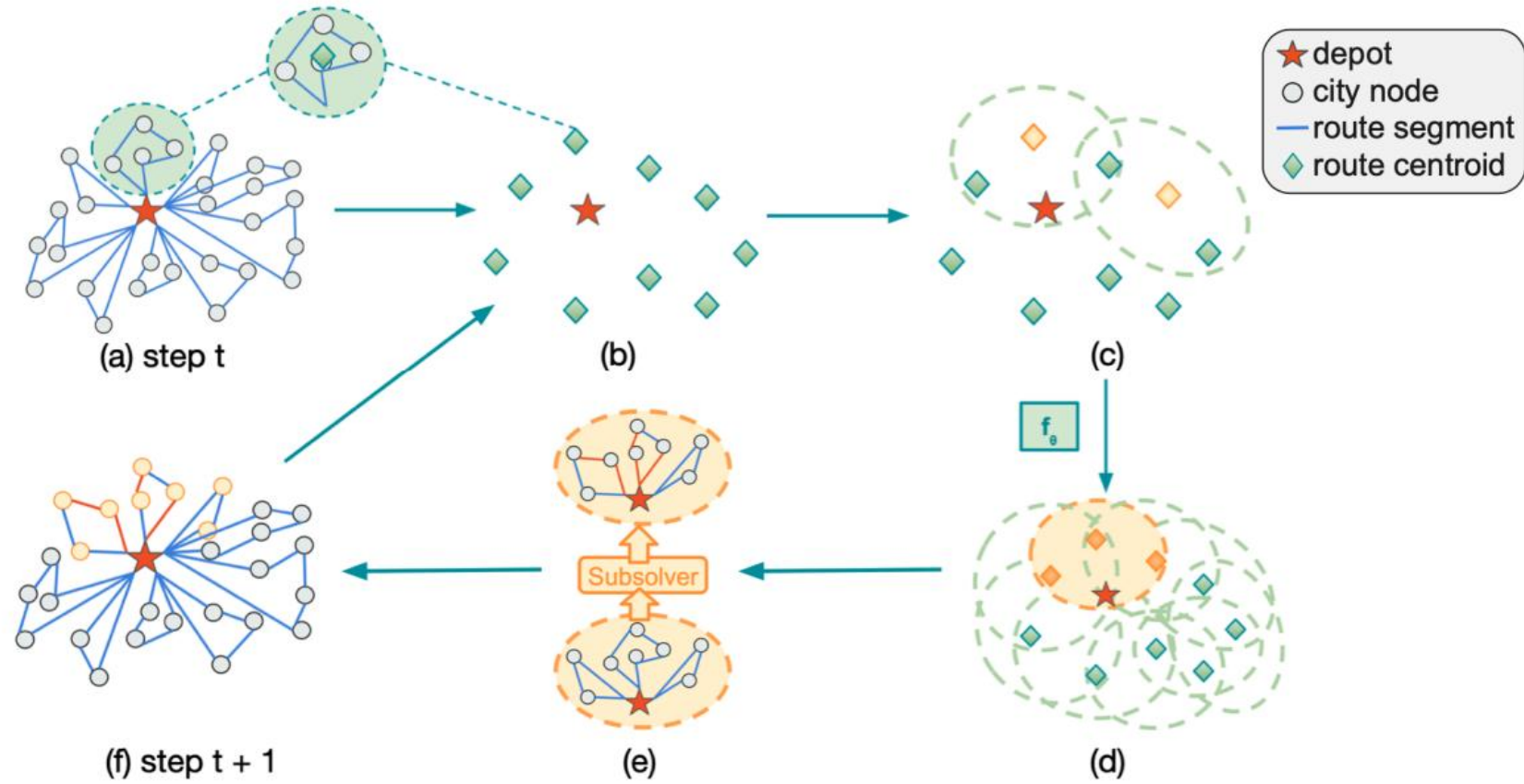
現状

CVRPはNP困難だが典型的なCVRPであれば実用的なソルバーが使える

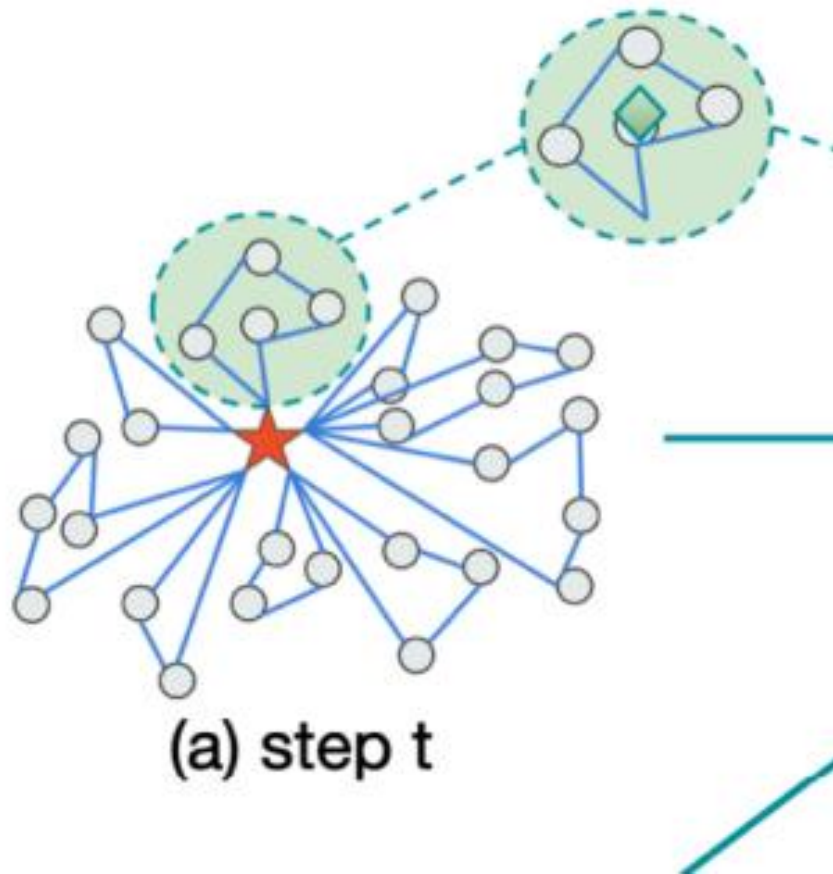
仮定

大きいCVRPは小さいCVRPの連続として効率的に解けるのではないか

Algorithm



Algorithm



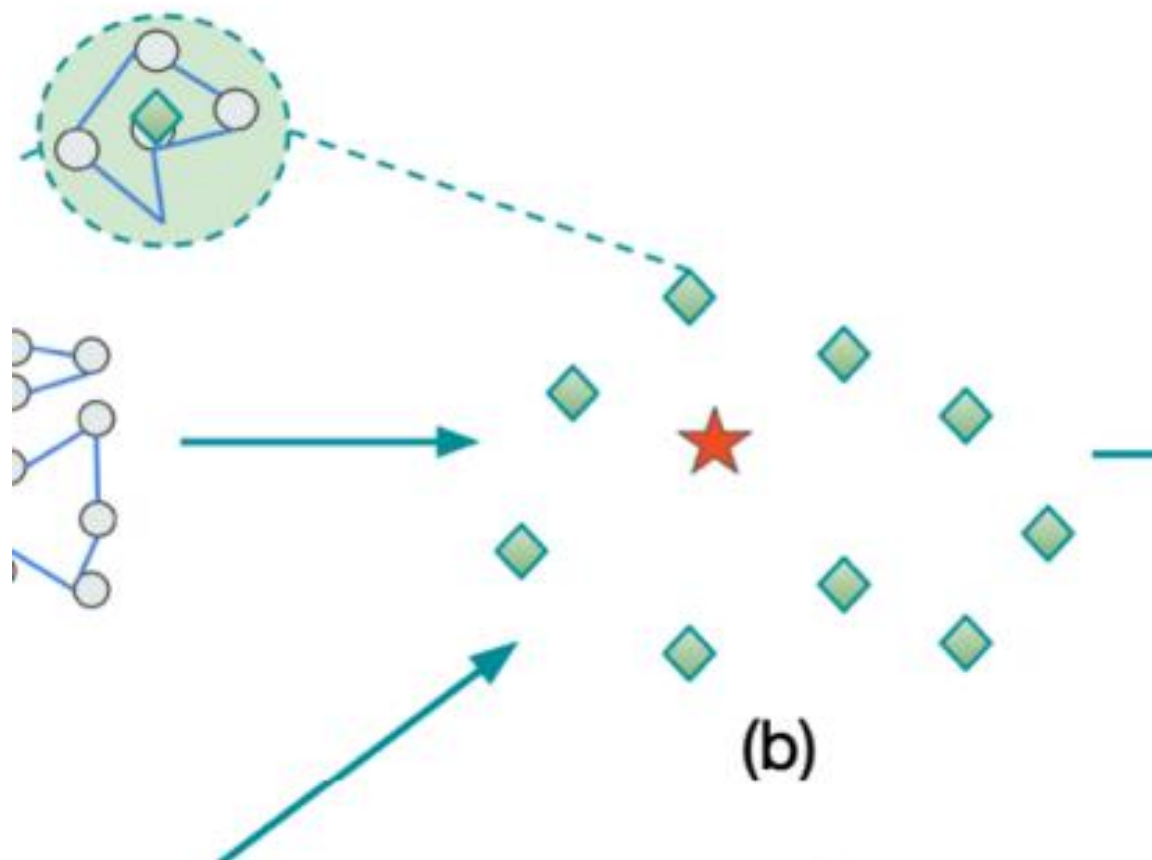
タイムステップの最初の状態
この状態のルートを初期解 X とする

○ ノード

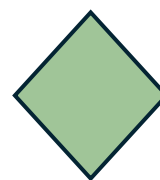
— エッジ

★ デポ

Algorithm

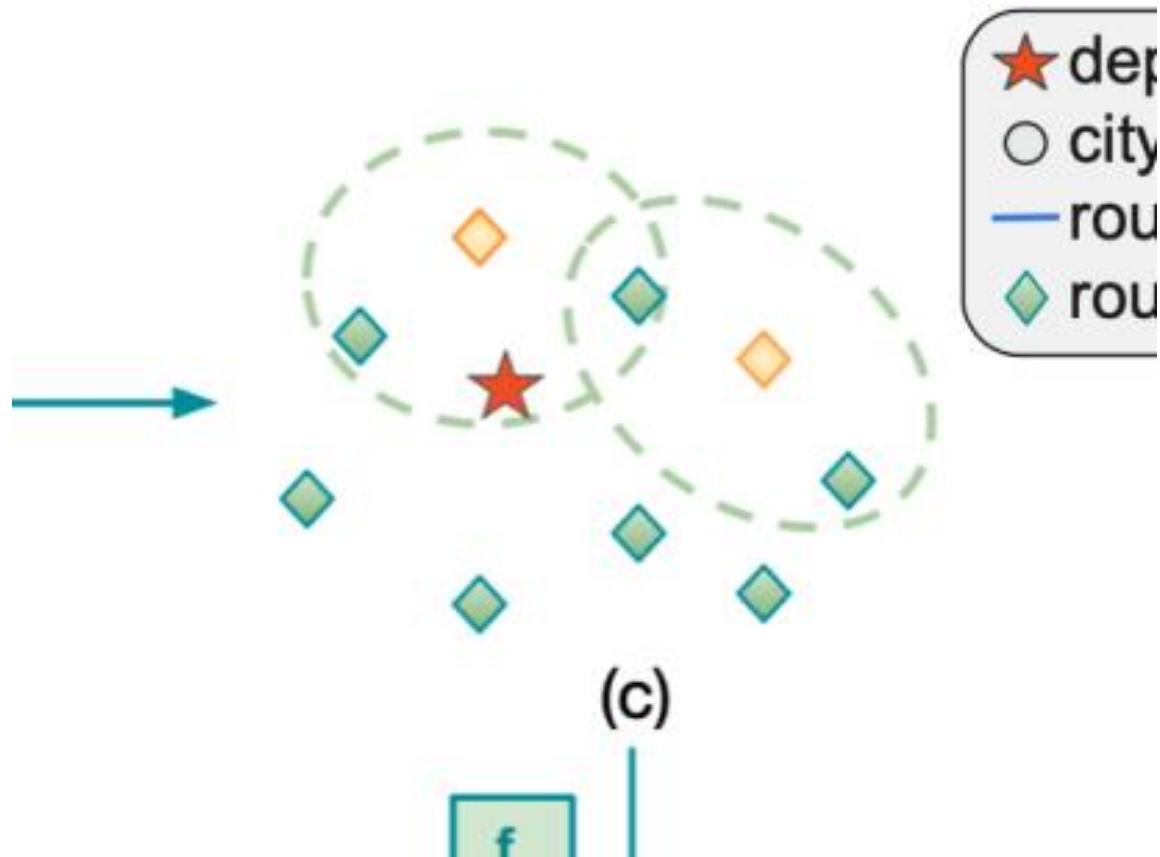


各ルートに含まれる都市の重心をとり、それぞれのルートをまとめる



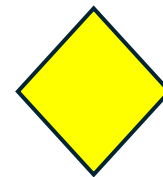
ループに含まれる都市の重心

Algorithm



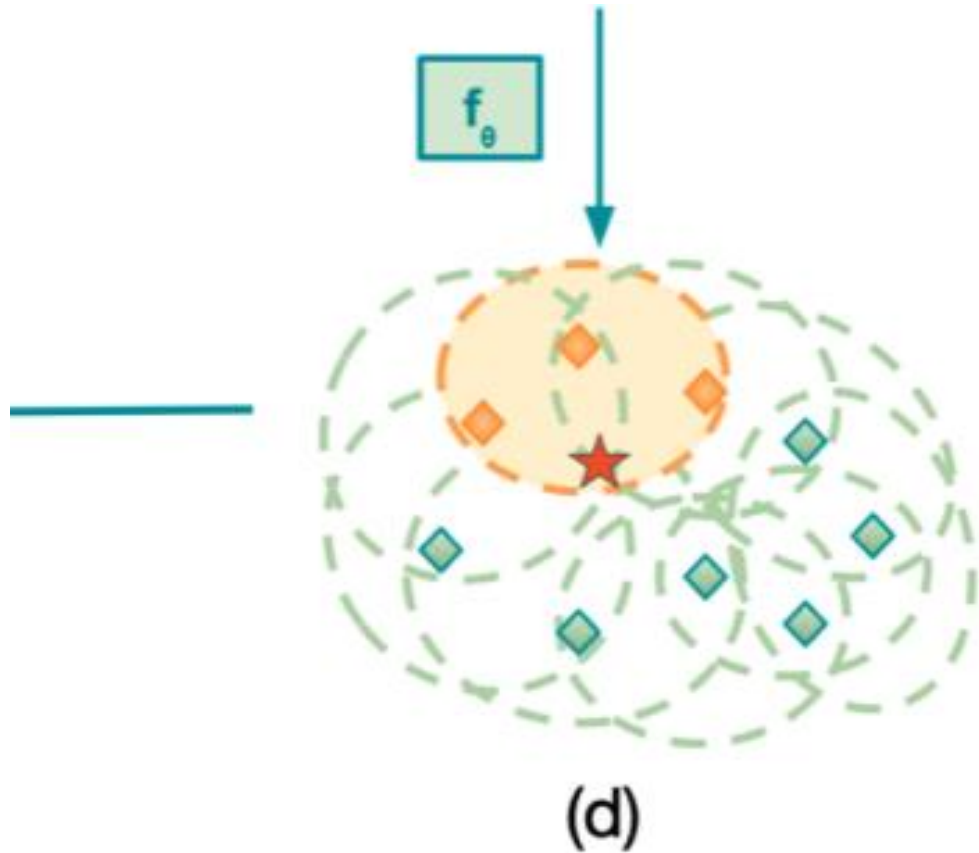
重心の数 k を決め、中心とする重心から近い k 個の重心をくくった部分問題を作成する。

左図は $k=3$ としたときの例



部分問題の中心となる重心

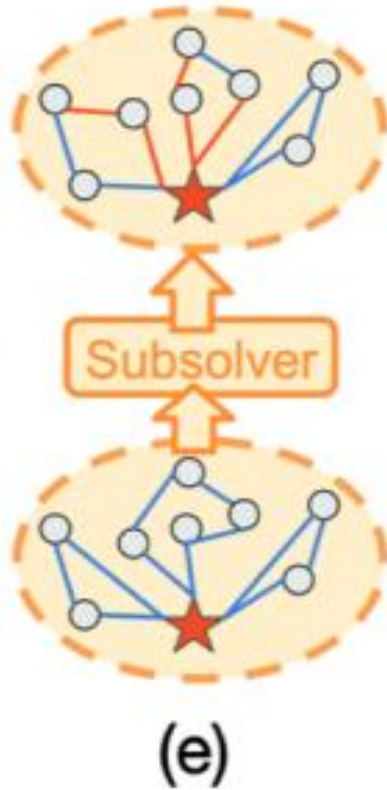
Algorithm



部分問題セクターにより、優先的に
解く部分問題を決める

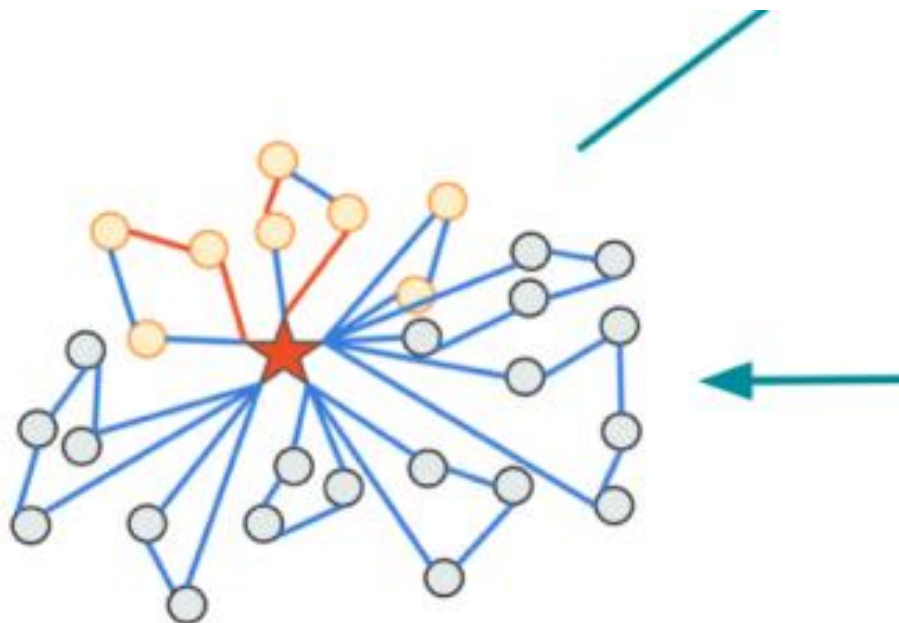
f_{θ} 部分問題セクター

Algorithm



先ほど選択した部分問題をサブソルバーで解く

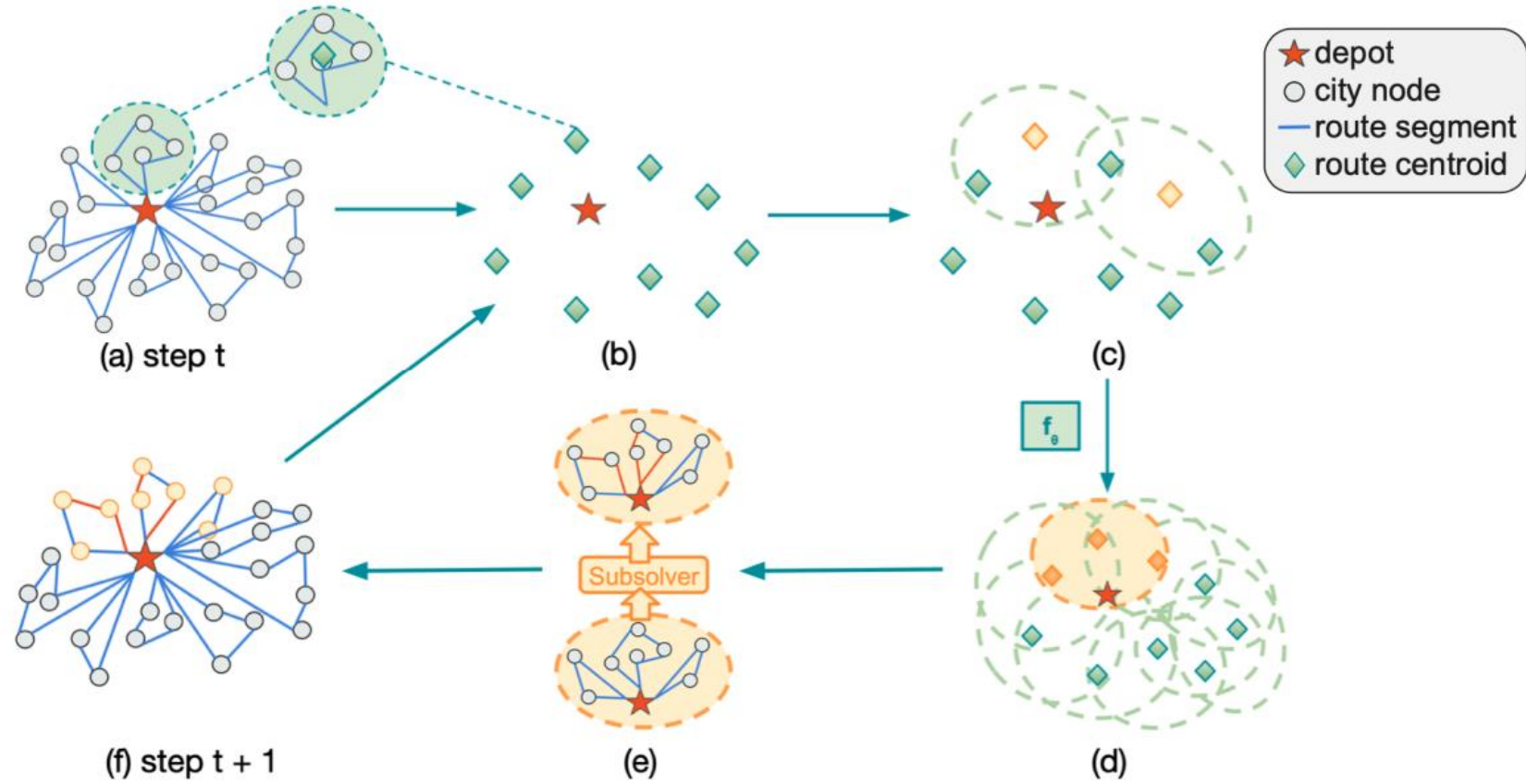
Algorithm



全体のルートと統合して新しいステップの解を形成する

(f) step $t + 1$

Algorithm



Algorithm

文字に起こすところな感じ

for Step $t = 1$ to T :

$S_{k,local} \leftarrow \text{Construct Subproblem}(P, X, k)$

$S \leftarrow f_{\theta}(S_{k,local})$

$X'_S \leftarrow \text{Subsolver}(S)$

$X \leftarrow X'_S \cup X_{P \setminus S}$

end for

インプット

P : 実際の問題

X : 初期解

f_{θ} : 部分問題選択関数

S : 部分問題

k : 部分問題のサイズ

Algorithm

文字に起こすところな感じ

for Step $t = 1$ to T :

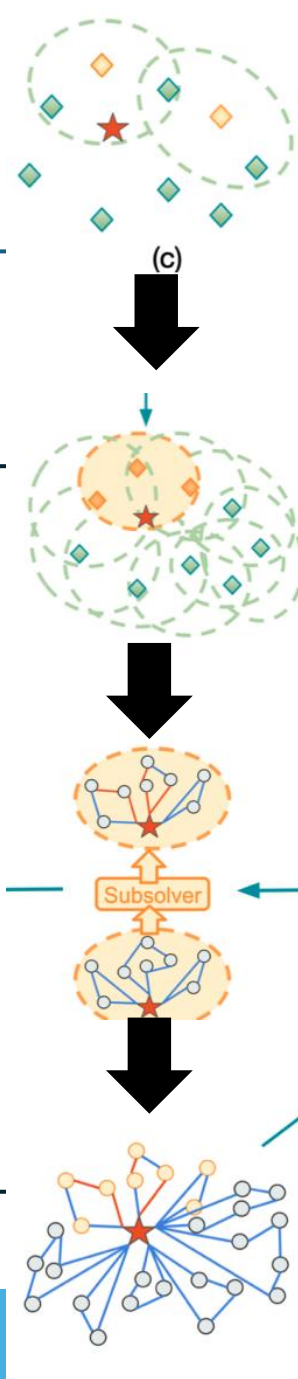
$S_{k,local} \leftarrow \text{Construct Subproblem}(P, X, k)$

$S \leftarrow f_{\theta}(S_{k,local})$

$X'_S \leftarrow \text{Subsolver}(S)$

$X \leftarrow X'_S \cup X_{P \setminus S}$

end for



インプット

P : 実際の問題

X : 初期解

f_{θ} : 部分問題選択関数

S : 部分問題

k : 部分問題のサイズ

The Restricted Subproblem Selection Space

部分問題の作成において制限をつけないと計算量が大変なことになってしまう……

ルート数を R とすると

制限なし
オーダー $O(2^R)$

R を含むか含まないかの
2択なので

個数制限
オーダー $O(R^k)$

k 個のルートを含むとその
組み合わせは ${}_RC_k$
なので

個数 & 空間制限
オーダー $O(N)$

中心となるルートを決めて
そのルートから近いもの
 k 個を選ぶので

The Restricted Subproblem Selection Space

部分問題の作成において制限をつけないと計算量が大変なことになってしまう……

ルート数を R とすると

制限なし	個数制限	個数 & 空間制限
オーダー $O(2^R)$	オーダー $O(R^k)$	オーダー $O(N)$

多くの組み合わせ最適化問題において近接する要素によって強く影響を受けるという事実より
個数 & 空間制限を採用する

5. Learning to Delegate

部分問題選択モデルの説明

Improvement as the criteria for subproblem selection

評価式

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

S	部分問題
$c(A)$	解Aにおける総コスト
X_S	部分問題Sの初期解
X'_S	部分問題SをLKH-3で解いた際の新しい解

解の更新について

1. $\delta(S) \geq 0$ のときに新しい解を採用する
2. 局所最適を避けるために特に部分問題選択でたくさんのステップを踏んだ後は $\delta(S) \leq 0$ でも新しい解を採用する
3. 適切なマスクを用いることで、解の更新が見られない部分問題が再び選ばれることを避けている

Subproblem selection

戦略

最も即時的な向上が見られる部分問題を選ぶ！

正当性1

サブソルバーは低レベルの操作を繰り返すので、問題の選択という高レベルの操作は高い即時効用を得られやすい

正当性2

VRPに関する強化学習の多くで高速収束のため低い割引率を使い、将来の報酬よりも直近の改善を重視している

さらには今回のモデルでは複数ステップにわたる改善効果も考慮できるので、長期的な影響も考慮して部分問題を選ぶこともできる

Ground-truth labels

選択空間が限られているので、各ステップで可能な部分問題を列挙し、サブソルバーをまわすことはできるがそれではあまりにも時間がかかりすぎる……



部分問題セクターがテスト時に最良の即時的な向上を予測できたらよさそう

Selection strategies: regression vs classification

回帰

VS

分類

連続した数値による出力

Transformer

- 気温の予測
 - 家の価格
- など

カテゴリ（ラベル）による出力

GAT

- 画像認識
 - 迷惑メール振り分け
- など

Selection strategies: regression vs classification

回帰

VS

分類

連続した数値による出力

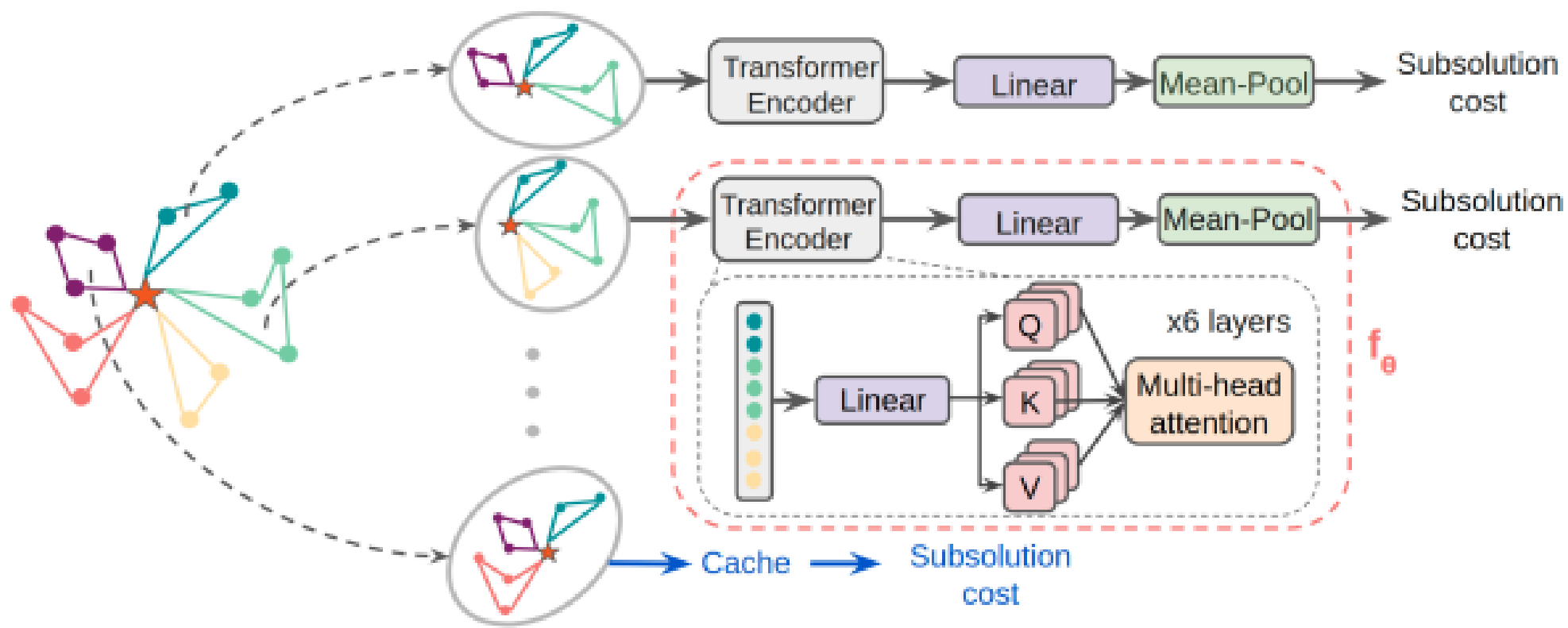
- 気温の予測

回帰で訓練された f_{θ} により $c(X'_s)$ を予測でき、 $c(X_s) - f_{\theta}(S)$ の最大値を計算して部分問題を選択する

も出力

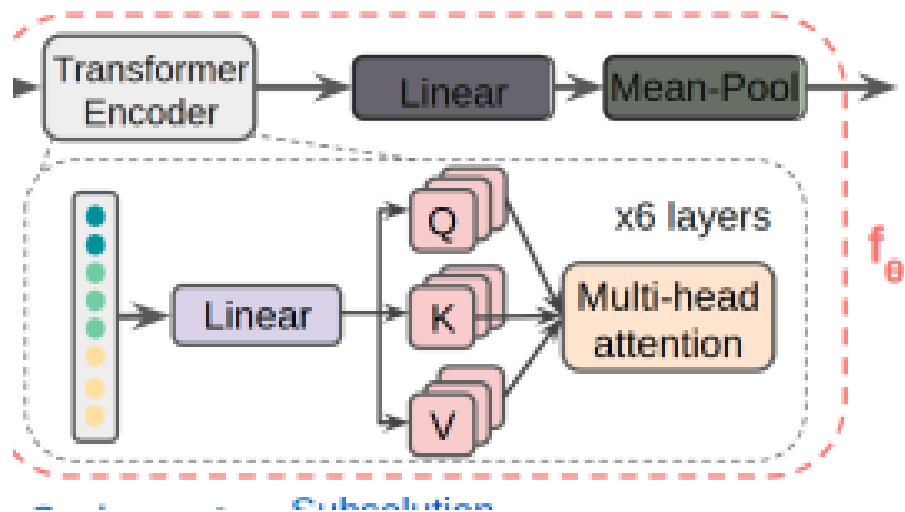
Network architecture

f_{θ} (部分問題セレクター) をトランスフォーマーエンコーダー構造で定義した



Network architecture

f_{θ} （部分問題セレクトター）をトランスフォーマーエンコーダー構造で定義した



ルートに関するデータ（位置、需要等）を線形化して、特徴量を抽出する。

Q：中心となるルートの特徴

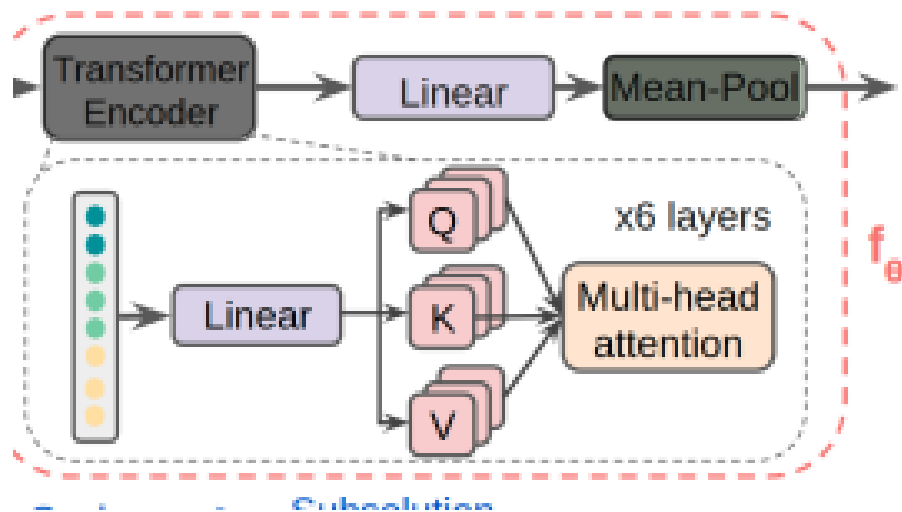
K：部分問題のほかのルートの特徴

V：実際に得る情報

QKVが複数あることでモデルの表現力が高くなっている

Network architecture

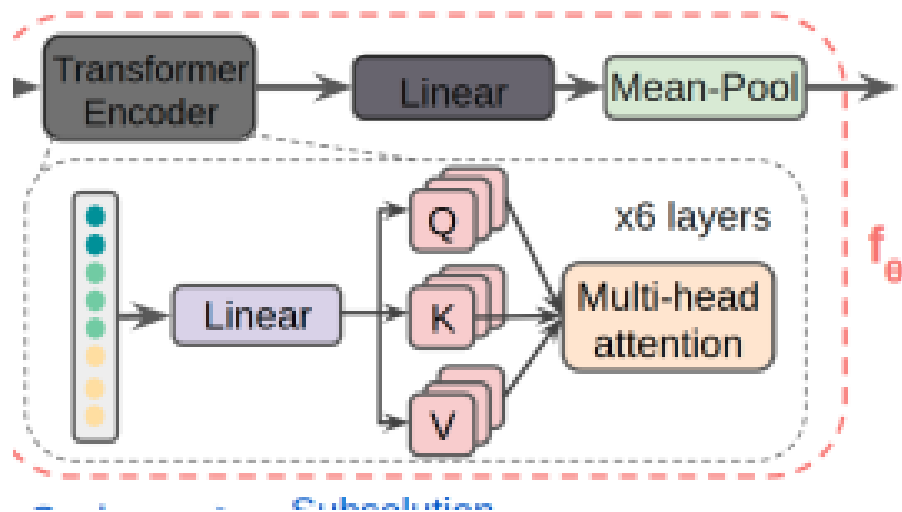
f_{θ} (部分問題セレクトター) をトランスフォーマーエンコーダー構造で定義した



Transformerの出力に対して、全都市共通の線形変換をする。
次の平均プーリングに向けて次元整形する。

Network architecture

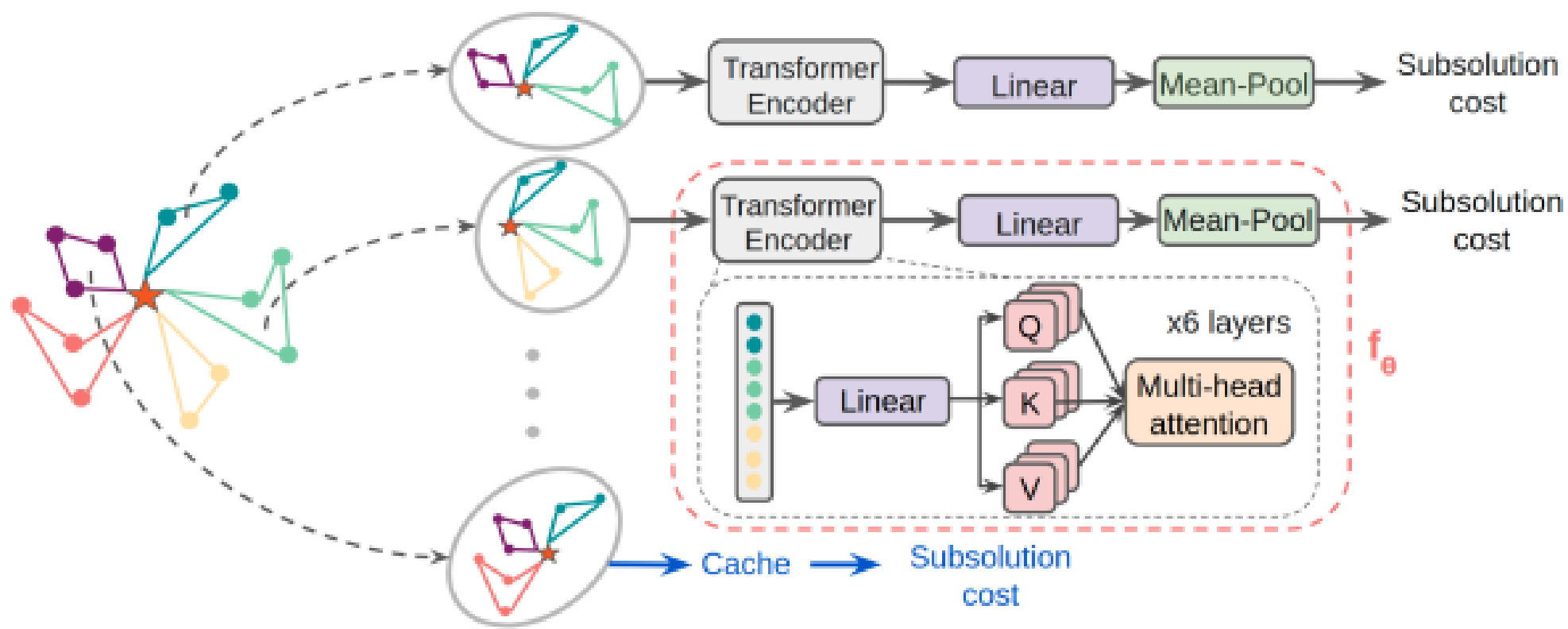
f_{θ} (部分問題セレクトター) をトランスフォーマーエンコーダー構造で定義した



都市ベクトルを平均化し、スカラー値に集約
このスカラーが予測値 $f_{\theta}(S)$ となる

Network architecture

f_{θ} (部分問題セレクター) をトランスフォーマーエンコーダー構造で定義した



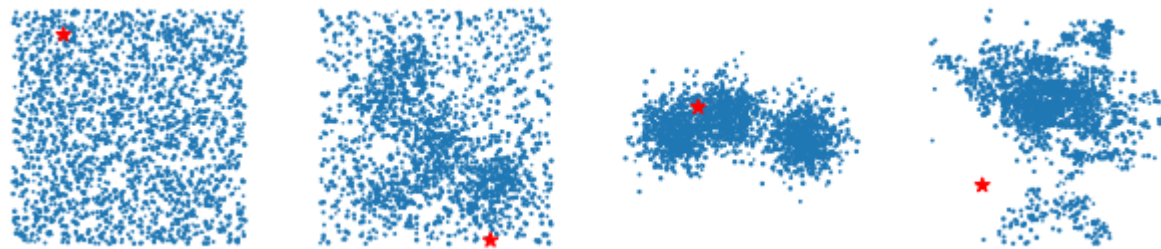
Loss function

$$L(\theta; S) = \begin{cases} \frac{1}{2}(f_{\theta}(S) - c(X'_S))^2, & \text{if } |f_{\theta}(S) - c(X'_S)| \leq 1 \\ |f_{\theta}(S) - c(X'_S)| - \frac{1}{2}, & \text{otherwise} \end{cases}$$

6. Experiments and Analysis

実験と分析

6.0 Distribution



左から一様分布、ミックス、クラスター、現実世界の分布

以下の質問に答えるために幅広い実験をした

1. 一様分布：ベースラインとどのような比較ができるか。時間、解の質
2. クラスター分布：クラスター型に分布した都市ではどのようにふるまうか
3. 我々のモデルは巨大都市や現実世界においても一般化できるか
4. 我々の手法はほかのVRPにも使えるか

6.1 Setup

まずVRPの分布を訓練データ2000、検証データ40、テストデータ40に分ける

初期解の作り方

- 空間をデポから放射状に36度ずつ10個に分ける
- 各部分集合においてLKH-3を100ステップ実行する

6.1 Setup

訓練データの作り方

- 訓練用セットの各インスタンス $\mathbf{P}$ に対して初期解 $\mathbf{X}$ からスタートし、各部分問題 $\mathbf{S} \in \mathbf{S}_{k,local}$ に対してサブソルバーを実行し、部分解 $\mathbf{X}_S$ と改善度 $\delta(\mathbf{S})$ を計算する
- これを反復フレームワークのステップ数30回分やる
- 例えばCVRPの一様分布だと $k=10$ で $\mathbf{S}_{k,local}$ を作り、LKH-3を500ステップ繰り返す
- 都市数 $N=2000$ と $N=3000$ では訓練データを作らない
 - 重すぎるのと $N=500,1000$ で十分な効果を得られたため

6.1 Setup

- $N \in \{500, 1000, 2000\}$ のモデルを組み合わせて単一モデルを学習させ、サイズごとに別々のモデルを作ることを避けた。
 - 汎化性能を上げるため
- 検証セットのパフォーマンスに基づいて、手作業で最適なハイパーパラメータを選択する。テストセットでの最終結果は、異なるランダムな初期値で5回実行した平均と標準誤差として記録する。

Baseline

基準	LKH-3	メタヒューリスティック、VRPへの汎用性
比較1	OR Tools	Googleによる組み合わせ最適化ライブラリ
比較3	AM	Transformerをベースにした深層強化学習
比較4	NeuRewriter	逐次的編集を行う深層強化学習

Baseline

部分問題セクターの性能を評価するため、*hand-crafted heuristics*という反復フレームワークを採用した3つの追加のベースラインを設計した。

1. Random

部分問題Sを一様確率で選ぶ

2. Count-based

それまでの選択であまり選ばれていない都市ノードを含む部分問題を選択する

3. Max Min Distance

前に選択された部分問題の最も近い重心からの距離が最大の部分問題を選択する。

Metrics

1. 手法Xからの向上

- 手法Yのトータルの改善度から手法Xのトータルの改善度を引いたもの

2. 手法Xからのスピードアップ

- 手法Xがある一定の解の品質になるまでにかかった時間を手法Yがある一定の解の品質になるまでにかかった時間から引く

95%から100%の解品質にするには不釣り合いな時間がかかるので、95%を要求する解の精度とする。

6.2 Uniform CVRP Distribution

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)	61.70	76s	119.55	3.0min	233.86	6.8min

LKH-3(95%)は95%の解精度

LKH-3(30k)は収束近くまで30kのステップを実行した

- Oursはどの問題サイズに対してもベストなパフォーマンスを示した。
 - LKH-3に対して7~15倍短い
 - 長い時間かけるとより良い精度になる
- 我々の部分問題セクターでは部分問題の解の変更されたところのみを評価すればよいため、計算時間は問題のサイズNにほとんど依存しない。

6.2 Uniform CVRP Distribution

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)	61.70	76s	119.55	3.0min	233.86	6.8min

LKH-3(95%)は95%の解精度

LKH-3(30k)は収束近くまで30kのステップを実行した

- LKH-3 for 30kはほかのヒューリスティックスやベースラインより精度がよかった。
 - 大きいサイズでも解の精度はよかったが時間がかかり長くなった。
 - これまでの学習ベースの方法は高速だが、解の質はかなり悪くなる

6.2 Uniform CVRP Distribution

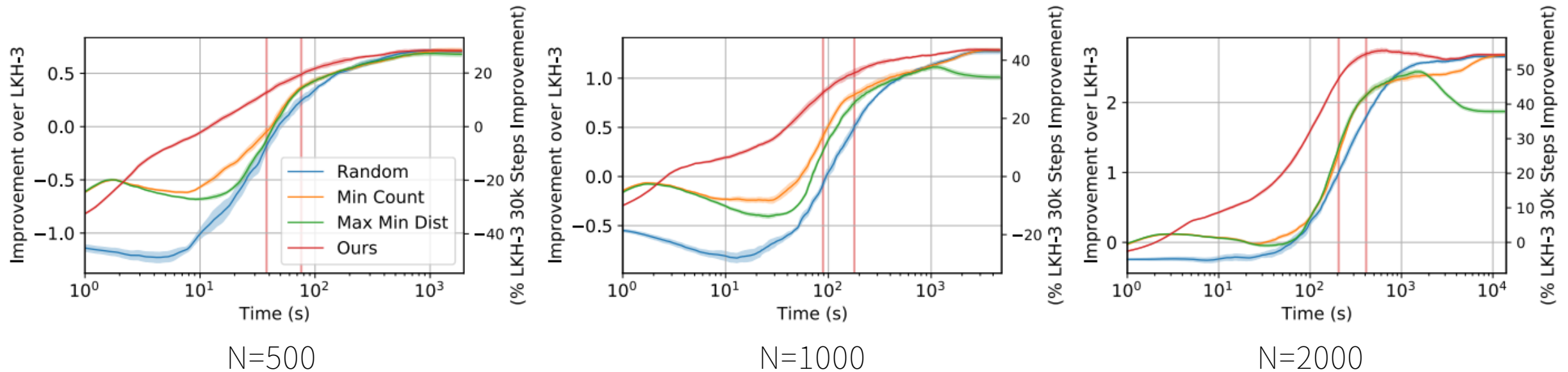
Method	N = 500		N = 1000		N = 2000	
	Cost	Time	Cost	Time	Cost	Time
LKH-3 (95%)	62.00	4.4min	120.02	18min	234.89	52min
LKH-3 (30k)	61.87	30min	119.88	77min	234.65	149min
OR Tools	65.59	15min	126.52	15min	244.65	15min
AM sampling	69.08	4.70s	151.01	17.40s	356.69	32.29s
AM greedy	68.58	25ms	142.84	56ms	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)	61.70	76s	119.55	3.0min	233.86	6.8min

- 我々のヒューリスティックベースラインであるRandom、Count-based、Max Min Distanceも、部分問題セレクトがなくてもLKH-3(95%)と比べて5~10倍のスピードアップを達成できた

LKH-3(95%)は95%の解精度

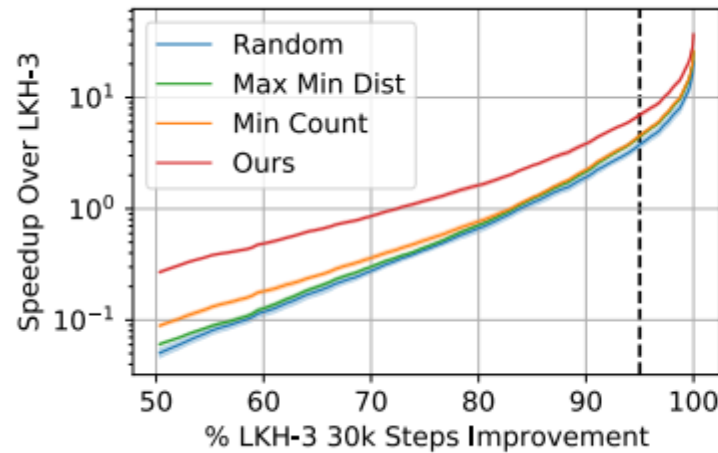
LKH-3(30k)は収束近くまで30kのステップを実行した

6.2 Uniform CVRP Distribution

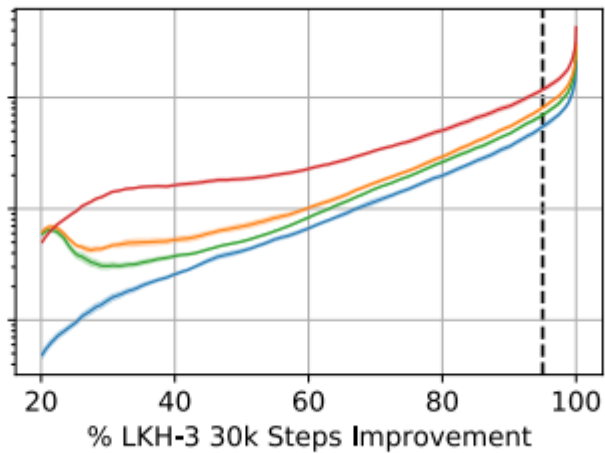


- 妥当な時間では我々の手法が最も解の質が良い
- かなりの時間が経つとどの手法も同じような解の質に落ち着く
 - それはすべての部分問題を選びきってしまうから

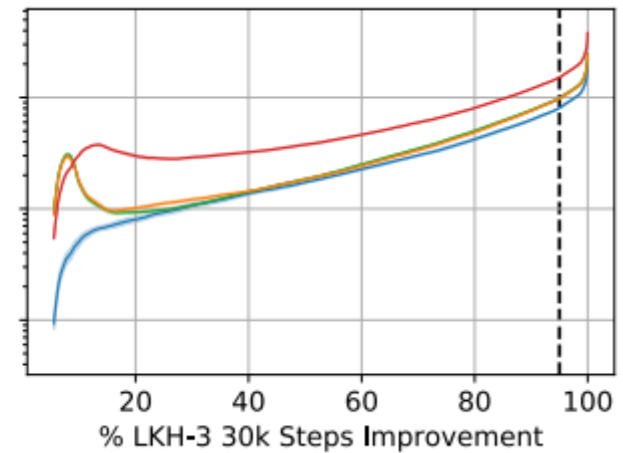
6.2 Uniform CVRP Distribution



N=500



N=1000



N=2000

- スピードアップは解の質が低いレベルでもしばしば顕著であり、解の質が高くなるほど向上する。
 - LKH-3は高い解の質からより高い解の質にするのには時間がかかる

6.3 Clusterd and Mixed CVRP Distributions

ミックス

50%の都市を一様分布から抽出、残りをクラスター分布の中心付近から抽出

N (都市数)
500, 1000, 2000

×

n_c
(クラスター中心)
3, 5, 7

×

m
クラスター
or ミックス

これを単一のモデルとして訓練。 N, n_c, m の組み合わせごとに10インスタンスの検証セットとテストセット

6.3 Clusterd and Mixed CVRP Distributions

Table 2: Speedup for $N = 2000$ clustered and mixed CVRPs at 95% LKH-3 30k solution quality.

Setting		$n_c = 3$	$n_c = 5$	$n_c = 7$
Clustered	Ours	26x	18x	25x
	Random	11x	7.5x	9.0x
Mixed	Ours	13x	14x	14x
	Random	6.6x	6.4x	7.6x

N=2000でクラスター分布とミックス分布で試した結果

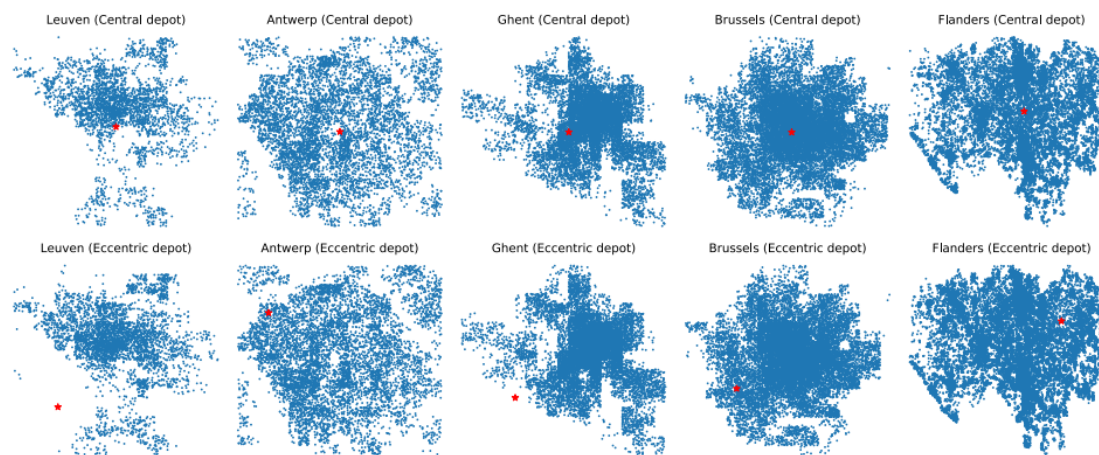
- Randomより2倍以上速い
- ミックスよりクラスターのほうが威力を発揮する。

6.4 Out-of-distribution Generalization

部分問題セクターが学習時には未知の $N=3000$ の一様分布や現実世界のCVRP分布にどのように一般化されるか調べる

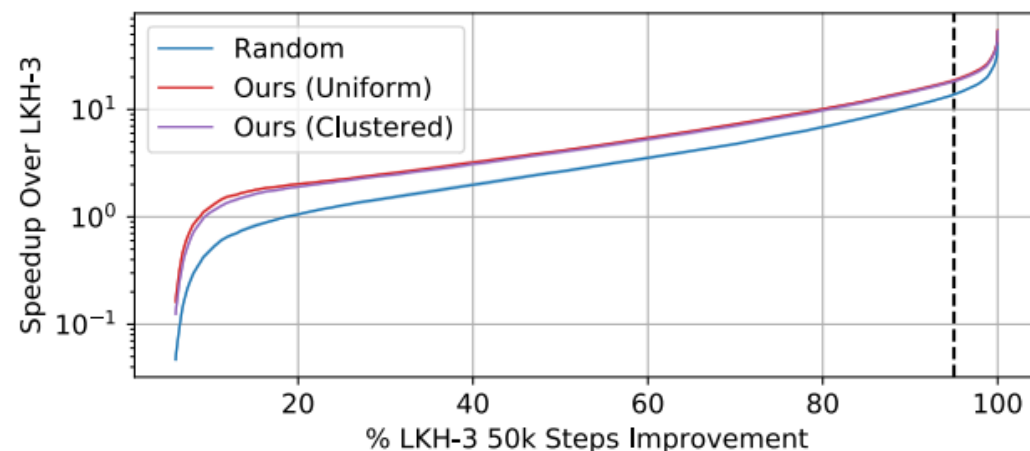
現実世界のCVRP
インスタンスの作成

- ベルギーの5都市から持ってくる
 - $N=3000 \sim 30000$
 - Leuven, Antwerp, Ghent, Brussels, Flanders
 - デポが中心にあるものとはずれにあるものの2種類
- 都市をランダムに選び $N=2000$ とする
- 需要は $N=2000$ に合うように再生成する



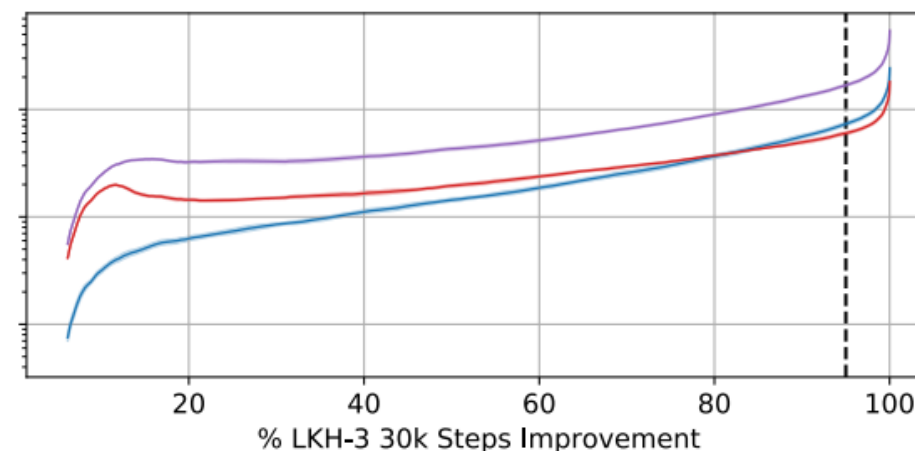
6.4 Out-of-distribution Generalization

一様分布とクラスター分布で学習した部分問題セレクタを新しいデータ分布に適用させた。



N=3000一様分布

N=3000に移行した場合、一様分布で学習してもクラスター分布で学習しても同じような性能



N=2000現実世界の分布

しかし、クラスターデータで学習した場合は現実世界の分布によく汎化される。
一様分布で学習した場合実世界には合わず、ランダムベースラインよりも速度が悪化した。

6.5 Other VRP Variants

- CVRPTWとVRPMPDについて考える
 - CVRPTWは各都市を訪問する時間的制約
 - VRPMPDは容量の制約に加えて集配と運搬の制約
- CVRPのように我々の反復フレームワークでかなりのスピードアップが見られたが学習により更なるスピードアップが見られる
- CVRPTW
 - Our 8.2x random 5.9x
- VRPMPD
 - Our 31x random 20x
- CVRPTWの時間窓制約が都市訪問の順序に厳密な順序付けを課し、部分問題セレクトの難易度を高めているのではないか

6.6 A State-of-the-Art CVRP Subsolver: HGS

- HGSは最先端のCVRPにフォーカスしたソルバーである
- $N=2000$ では我々の手法はHGSより103倍速かった。
 - ランダムは77倍
 - LKH-3は $N=2000$ で15倍だった
- $N=3000$ では198倍
 - ランダムは152倍
- HGSは500から1000都市の中規模問題用に設計され、キャリブレーションされているため、大規模VRPのサブソルバーとしてよりよく機能したと考えられる。

7. Conclusion

結論

Conclusion

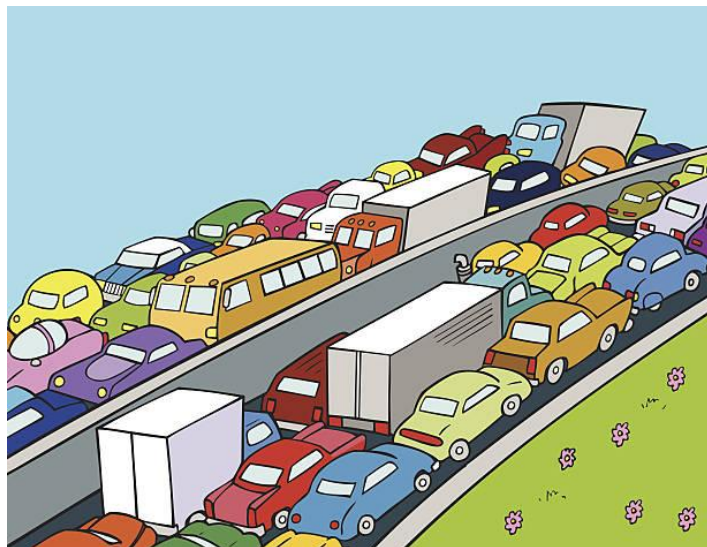
- 空間局所性を用いることで部分問題セクターが現実的に使えるようになった
- 提案手法は $N=3000$ までのVRPソルバーを高速化し、必要な計算時間を一桁削っている
- 提案手法は学習をしない手法（ソルバー）より1.5~2倍速く解くことができ、この結果は様々なVRP分布、変種、ソルバーに対しても同様であった

Conclusion

- これまでの機械学習は強化学習に依存していたが、この研究は、中程度の品質のラベルのみを用いた教師あり学習で問題の選択を行うことが高速化につながることを示した。
- 組み合わせ最適化に中程度の品質のラベルを効果的に活用するほかの方法を探ることが考えられる。
- このフレームワークは機械学習とORの両コミュニティにおいてスケールアップするための強力な技術として役立つだろう。

Negative Social Impact

大規模なVRPのより効率的なソリューションの実現は、配送車両のトラフィックを増加させたり、実店舗型小売店との競争を激化させるサービスの集中化といった負の外部性をもたらす可能性がある。



VS

amazon

所感

- 良かった点
 - 今までの強化学習的アプローチにとらわれず、改善すべきルートを予測するというやり方により高速化を実現できていた点
 - 様々なソルバーや分布による比較やほかの制約付きVRPでの検証が行われていた点
- 微妙だった点
 - Transformerの説明があまりなかったため、ブラックボックス感があった
 - Learning to delegate（学習して委ねる）という概念がくどかった
 - 発想としては割とわかりやすいのでは？と思ってしまった
 - 全体最適を求めるのに即時的向上を目指すことへの正当性が微妙？

所感

- 感想

- Transformerの使い方として最も改善したほうが良いルートを予測するというのがおもしろかった。
- 微妙な点としてTransformerの中身がわからないことを挙げたが、逆に言えば中身がわからなくてもやっていることをなんとなく理解できる論文だったのでわかりやすくはあった。
- やっていること自体はそんなに難しくないで思いついてしまえば意外とできるのでは？と思った
 - 思いつくのが難しいのだろうか……
- 最後の社会への悪影響は入れないといけないのでしょうか…？

Appendix

回帰でなく分類で行った場合
TransformerではなくGATを使用

Appendix

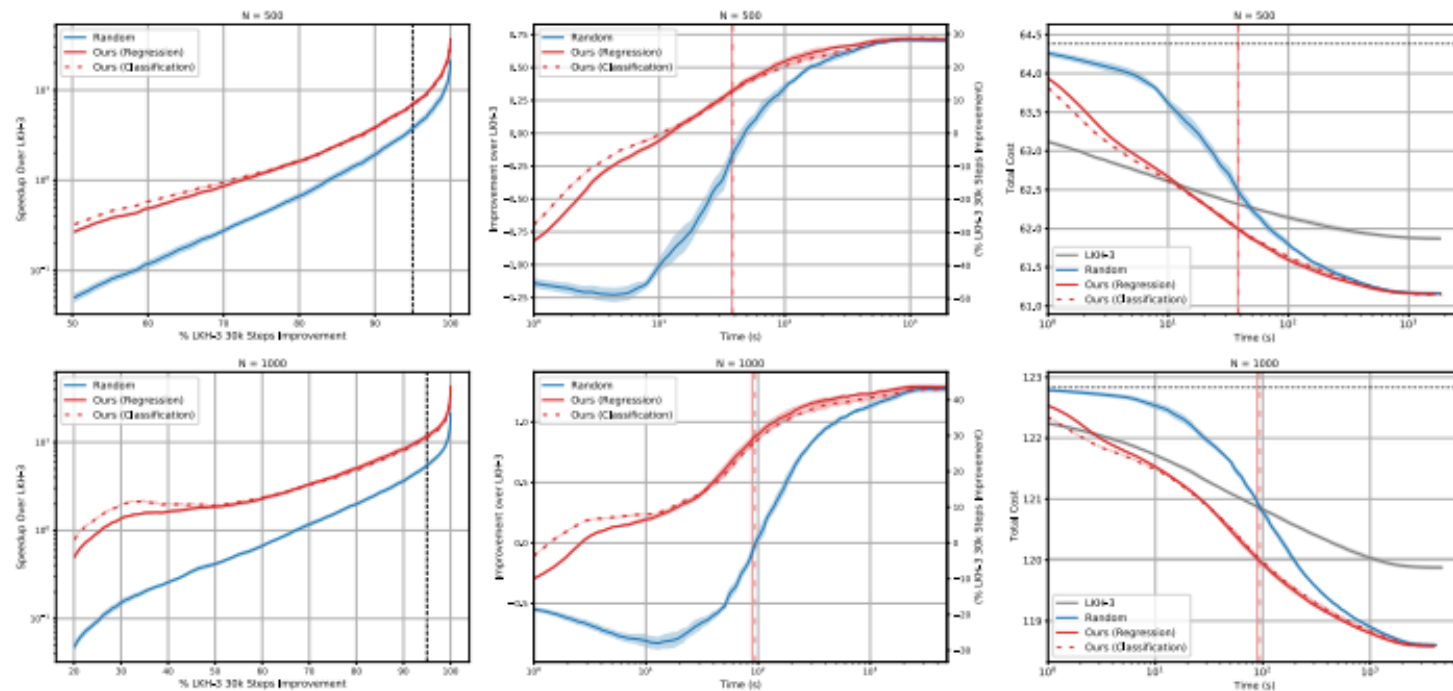
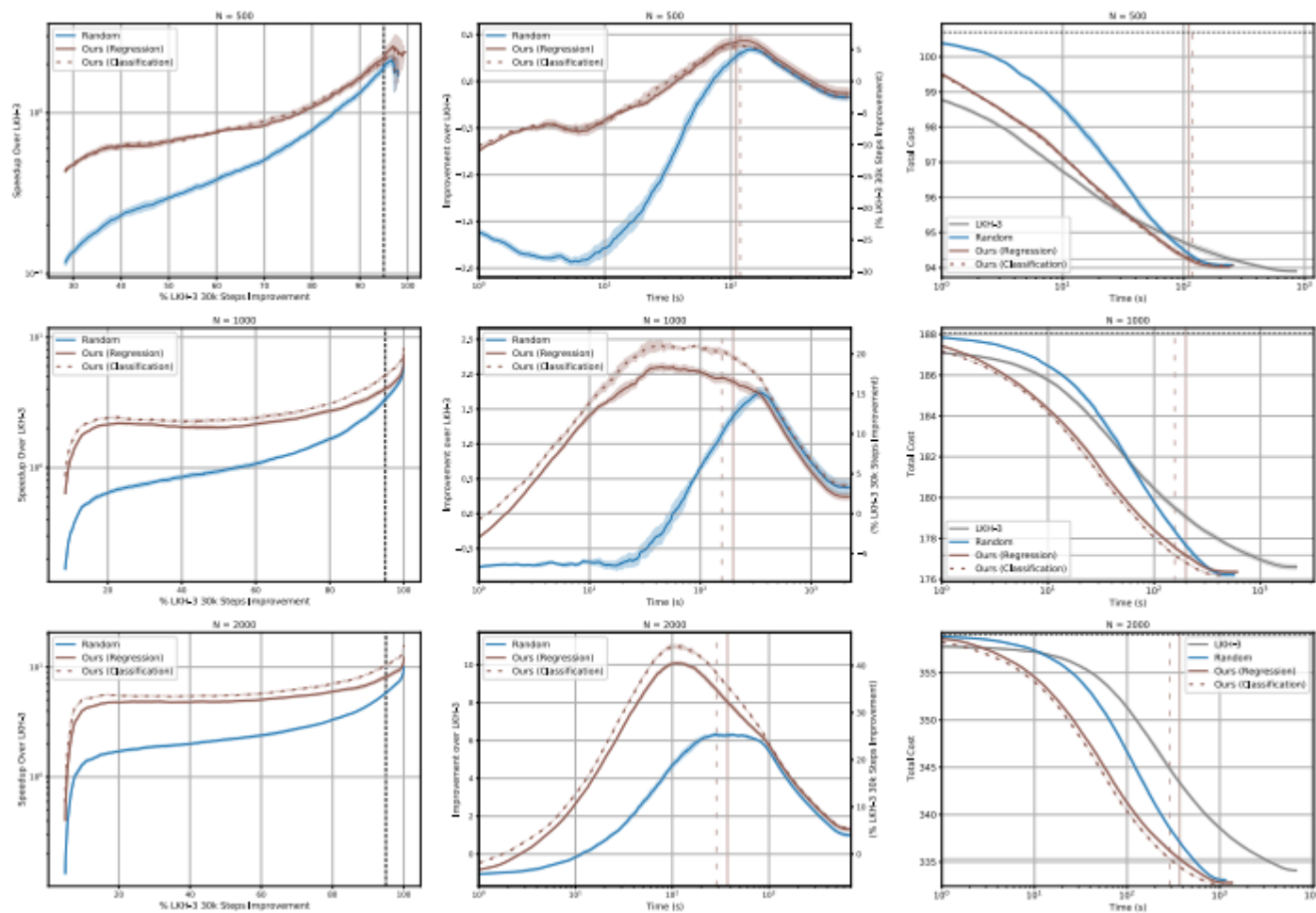


Figure 13: **Uniform CVRP distribution, regression vs classification.** Graphs of speedup (left column, higher is better), improvement (middle column, higher is better), and total cost (right column, lower is better). The right vertical axis in each improvement graph (middle column) indicates the improvement over LKH-3 as a percentage of the total improvement that LKH-3 attains over 30k steps. We use loosely dashed lines to indicate classification results. The horizontal dotted black line in the total cost graph (right) indicates the initial solution quality for all methods. The vertical dashed black line in the speedup graph (left) and the horizontal LKH-3-colored line in the total cost graph (right) indicate the 95% LKH-3 solution quality. The styled vertical lines in the improvement graph (middle) and the total cost graph (right) indicate computation times required for the corresponding learning-based methods to reach the aforementioned solution quality.

Appendix

- $N = 500, 1000$ の Uniform CVRP においては、GAT（分類）は Transformer（回帰）とほぼ同等の性能。
- 速度（speedup）、改善度（improvement）、コスト（total cost）の全てで大差なし。
- 回帰モデルのほうがマルチスケール対応可能（ $N=500, 1000, 2000$ に共通）で、学習も軽い ため本論文ではそちらを主に採用。

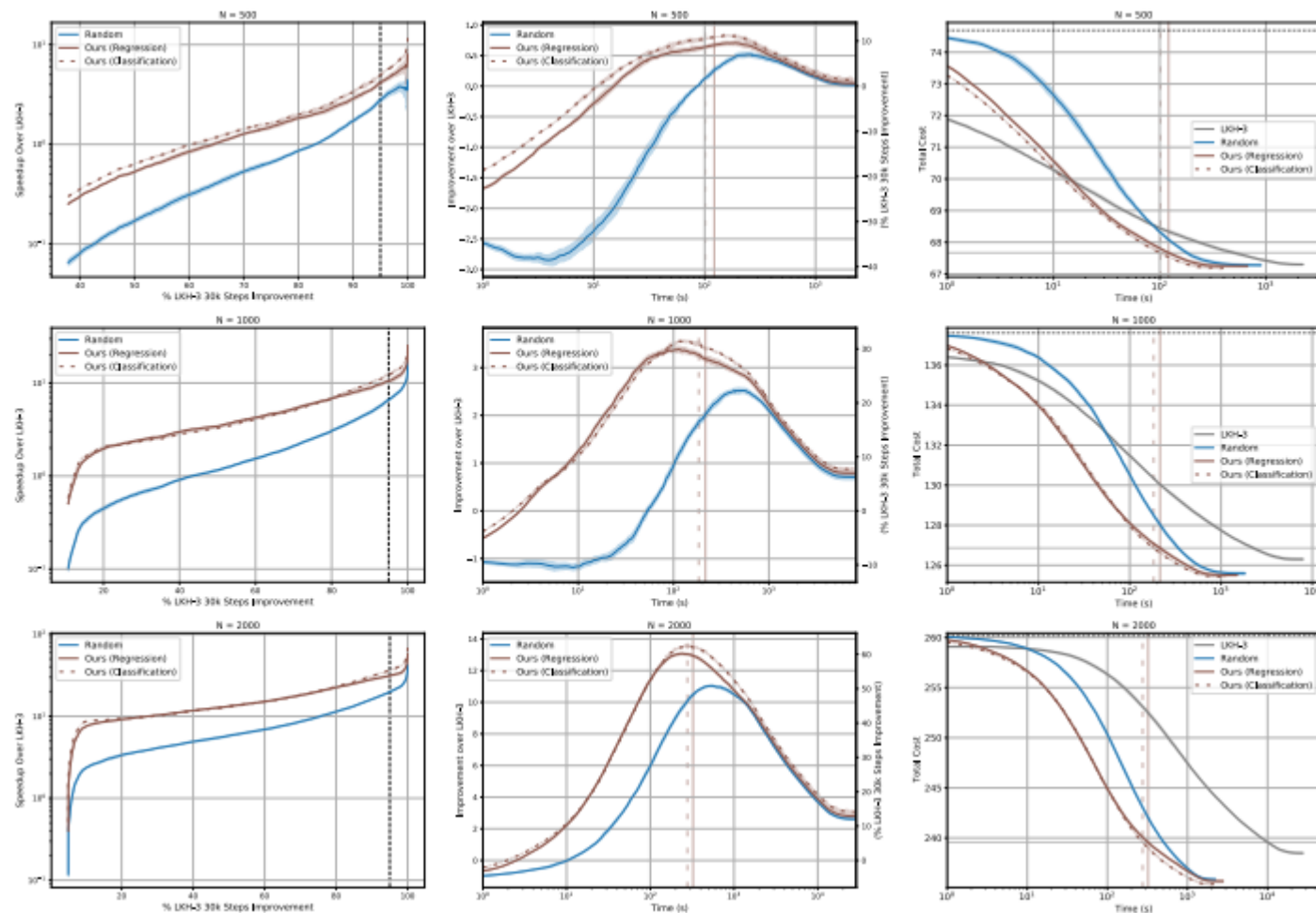
Appendix



Appendix

Figure 14: **Uniform CVRPTW distributions.** Graphs of speedup (left column, higher is better), improvement (middle column, higher is better), and total cost (right column, lower is better). The right vertical axis in each improvement graph (middle column) indicates the improvement over LKH-3 as a percentage of the total improvement that LKH-3 attains over 30k steps. We use loosely dashed lines to indicate classification results. The horizontal dotted black line in the total cost graph (right) indicates the initial solution quality for all methods. The vertical dashed black line in the speedup graph (left) and the horizontal LKH-3-colored line in the total cost graph (right) indicate the 95% LKH-3 solution quality. The styled vertical lines in the improvement graph (middle) and the total cost graph (right) indicate computation times required for the corresponding learning-based methods to reach the aforementioned solution quality.

Appendix



Appendix

Figure 15: **Uniform VRPMPD distributions.** Graphs of speedup (left column, higher is better), improvement (middle column, higher is better), and total cost (right column, lower is better). The right vertical axis in each improvement graph (middle column) indicates the improvement over LKH-3 as a percentage of the total improvement that LKH-3 attains over 30k steps. We use loosely dashed lines to indicate classification results. The horizontal dotted black line in the total cost graph (right) indicates the initial solution quality for all methods. The vertical dashed black line in the speedup graph (left) and the horizontal LKH-3-colored line in the total cost graph (right) indicate the 95% LKH-3 solution quality. The styled vertical lines in the improvement graph (middle) and the total cost graph (right) indicate computation times required for the corresponding learning-based methods to reach the aforementioned solution quality.

Appendix

- CVRPTW（時間窓付きCVRP）および VRPMPD（配送+集荷）のような複雑な制約付きVRPにおいては、GAT（分類）の方が回帰モデルより若干良い性能を示した。
- 特に $N=2000$ においては、GAT の改善度が高く、回帰モデルよりも低コスト解を達成しているケースが多かった。