Factor Graph Neural Network Meets Max-Sum: A Real-Time Route Planning Algorithm for Massive-Scale Trips

Yixuan Li[1], Wanyuan Wang[1]*, Weiyi Xu[1], Yanchen Deng[2], Weiwei Wu[1].

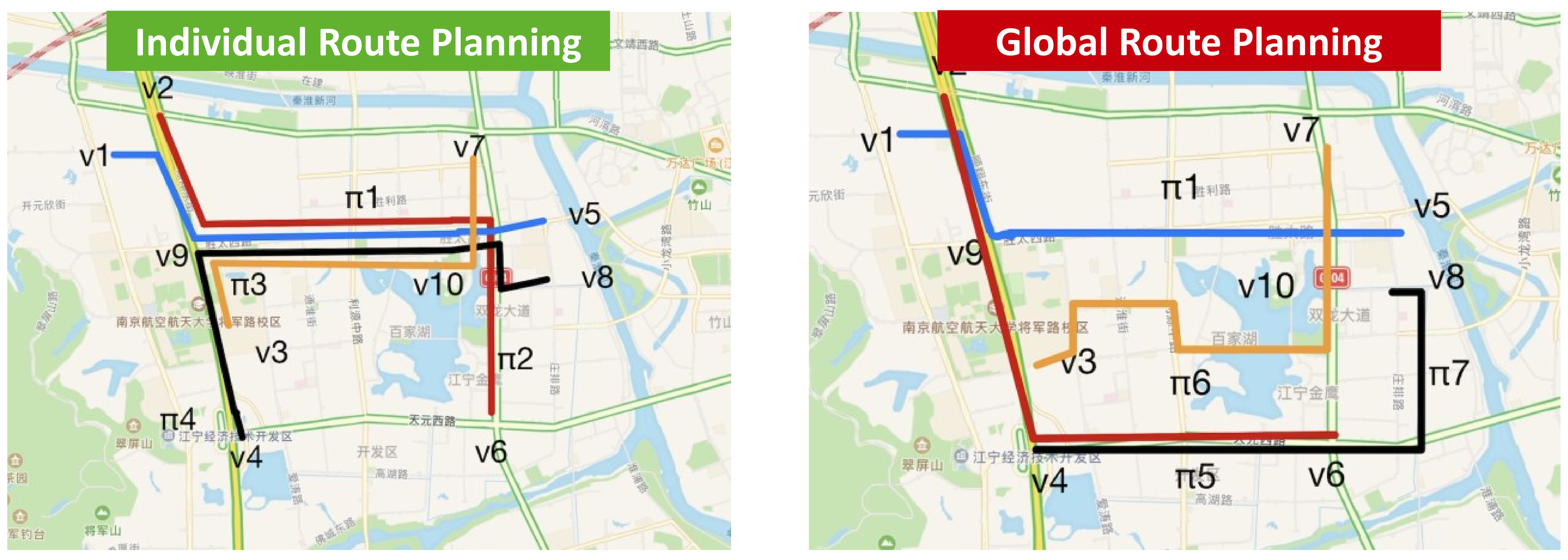
[1] School of Computer Science and Engineering, Southeast University, Nanjing, China

[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore

Email: {yixuanli, wywang, weiweiwu}@seu.edu.cn, wxu79631@gmail.com, ycdeng@ntu.edu.sg

AAMAS 2024 AUCKLAND NZ

Backgrounds



- **Individual route planning** only considers **current** traffic condition, ignoring other queries.
- **Global route planning** considers the **potential** vehicles generated by other user queries.
- The goal of Global Route Planning is to **minimize the total travel time** for all user queries.

Problem Definition:

- The travelling time $t(e)$ on each road e:

$$t(e) = t_{min}(e) \times (1 + \alpha \times f_e)$$

$t(e)$ is affected by f_e, the number of vehicles on this road.

- **The goal of individual** route planning is to minimise the travel time for **one** user π:

$$T(\pi) = \sum_{e(v_i, v_j) \in \pi} f(e(v_i, v_j))$$

- **The goal of Global** Route Planning is to minimise the travel time for **all the users** Π:

$$GT(\Pi) = \sum_{i=1}^{|\Pi|} T(\pi_i)$$

Existing Methods:

- **Exact algorithm** bears **exponential** time complexity and could not be applied to **real-world** scenarios.
- **Heuristics Methods**: greedy algorithms [1],[2], Monte Carlo tree search [3] etc.

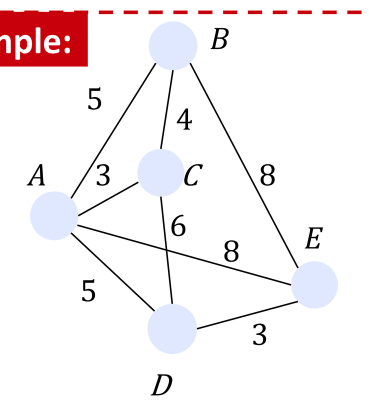
Drawbacks: low **efficiency** and **accuracy**

Our Contributions:

- **Graph Model to solve GRP:** Route-Query Factor Graph
- **Hybrid pruning technique** for Max-Sum
- **End to end framework:** Route-Query Factor Graph Neural Network

Problem Formulation - Route-Query Factor Graph

Example:



$q_1: A \to D$

$q_2: A \to B$

$q_3: A \to C$

$q_4: B \to E$

Road Map **Queries**

Minimizing total travel time for 4 users

Motivation: From the whole to the parts: Decompose the total travel time by **road**.

Decision variable: candidate routes

To simplify the problem, the candidates are the **shortest k** ones

queries $Q = \{q_1, \dots, q_n\}$

Variables $X = \{x_1, \dots, x_n\}$

Domains $D = \{D_1, \dots, D_n\}$

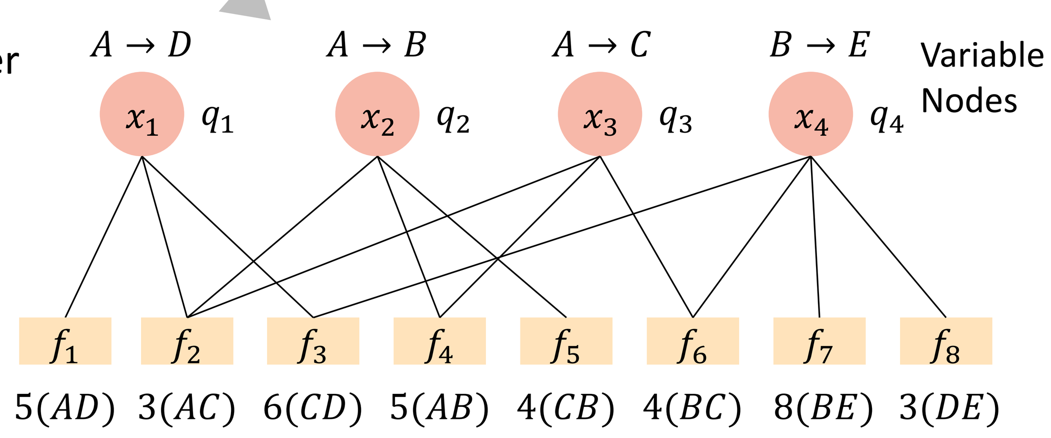
Top-k

k=2

Objective function: sum of the user travel times in all candidate roads:

$$F = f_1, \dots, f_8$$

$$f(e) = t_{min}(e) \times (1 + \alpha \times f_e)$$



Max-sum with Hybrid Pruning

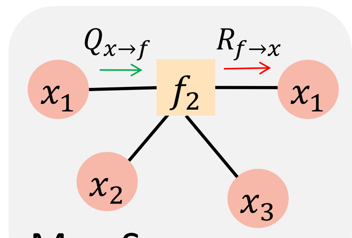
This model can be solved by message passing algorithm: Max-Sum [4]

Advantages:

Parallel computation: each node as a computational unit

Distributed computing to rapidly solve and protect privacy

Controllable: can stop after any iteration and return results



Process of Max-Sum

1. Query Message: variable nodes → function nodes (which road to take)

$$Q_{x_i \to f}^{k}(x_i) = \sum_{f' \in N(x_i) \setminus f} R_{f' \to x_i}^{k-1}(x_i) + \alpha_i$$

2. Response Message: function → variable nodes (maximum possible cost caused by query nodes)

$$R_{f \to x_i}^{k}(x_i) = \boxed{\min_{N(f) \setminus x_i}} \left(\sigma(N(f)) + \sum_{x_i' \in N(f) \setminus x_i} Q_{x_i' \to f}^{k}(x_i')\right)$$

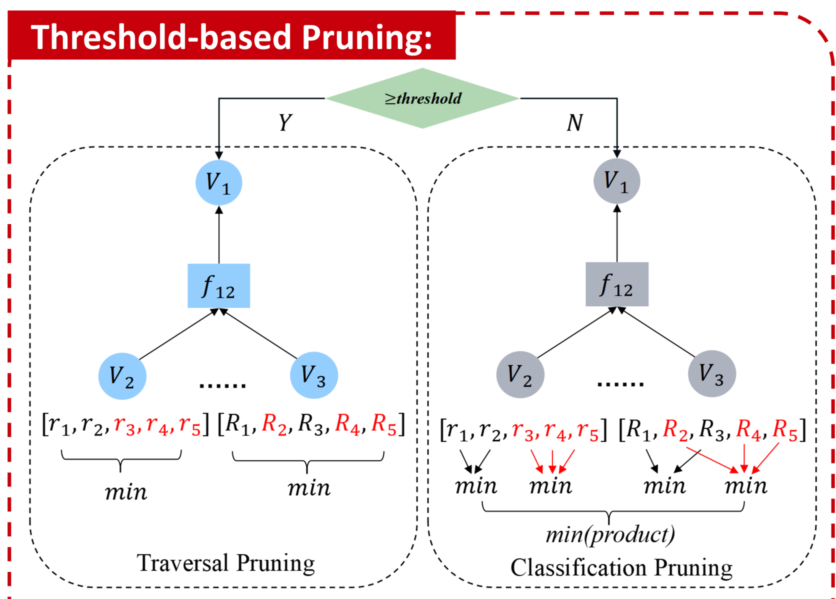
3. Decision: The variable nodes use the information to make decision

$$\tilde{d}_i = \underset{x_i \in D_i}{\operatorname{argmin}} \sum_{f' \in N(x_i)} R_{f' \to x_i}^{k}(x_i)$$

Exponential complexity:

Response message need to consider all the possible combinations of neighbors, for N neighbors, K candidates, the computational complexity is $O(K^N)$

Threshold-based Pruning:



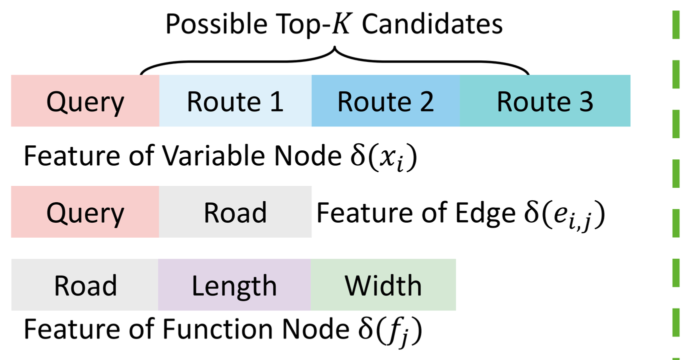
Classification Pruning: Queries can be divided into two categories: **passing through** the road or not. So finding the optimal value **respectively** can reduce computational complexity to $O(2^N)$.

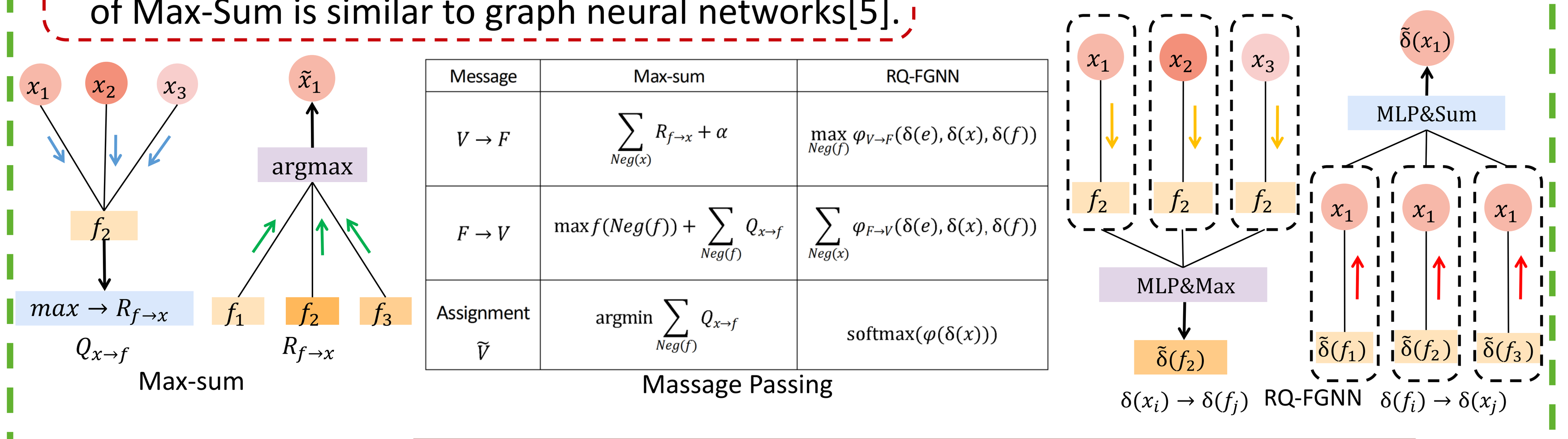
Traversal pruning: greedy method for busy traffic, complexity of $O(KN)$.

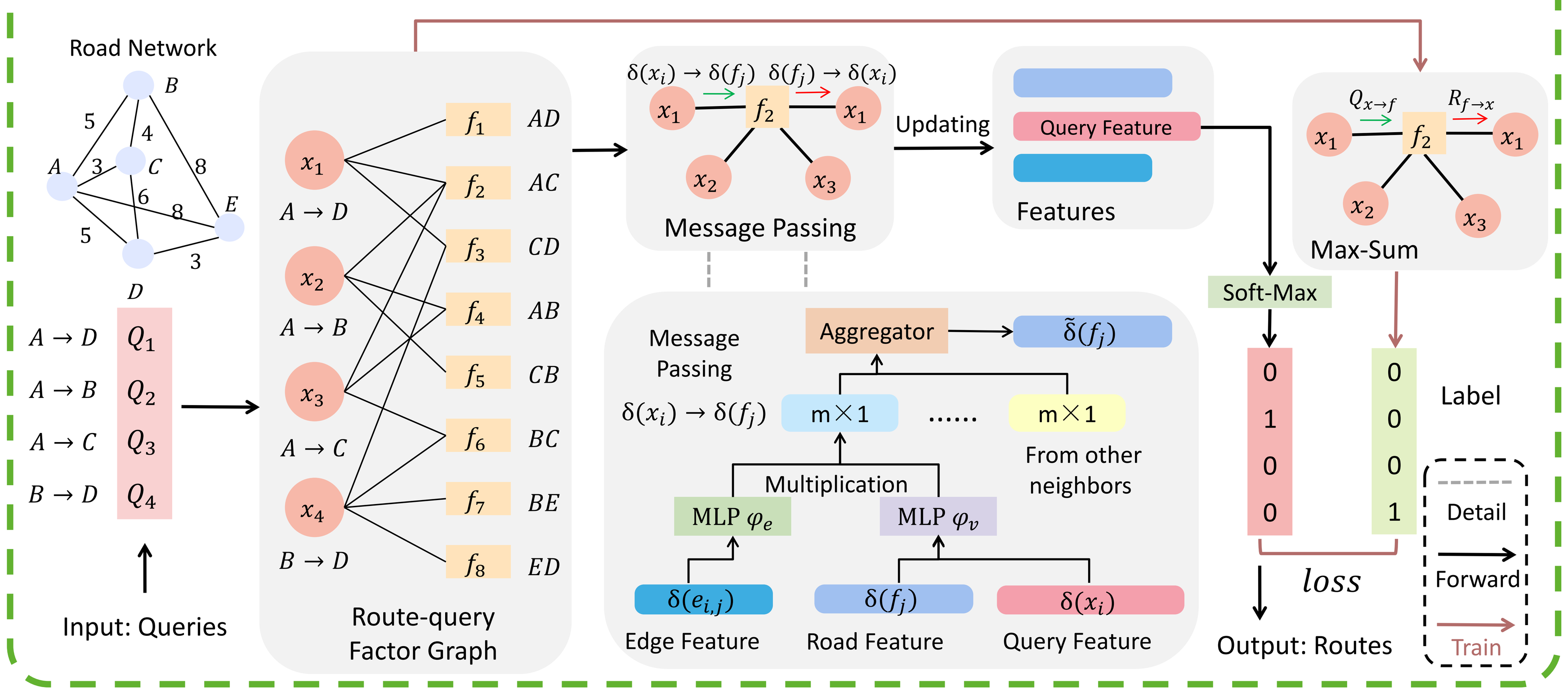
End to end Framework

Motivation

- **Historical Information:** many closed-related GRP instances must be solved repeatedly.
- **Similar Patterns:** the same road network and the set of candidate paths for each query is invariant.
- **Computation Process:** the computation process of Max-Sum is similar to graph neural networks[5].







Experiments

Experimental Settings

Datasets:

TG: Real-world San Joaquin County Road Network [6], **SG:** Synthetic road network

Metrics:

Global travel time: $GT(\Pi)$

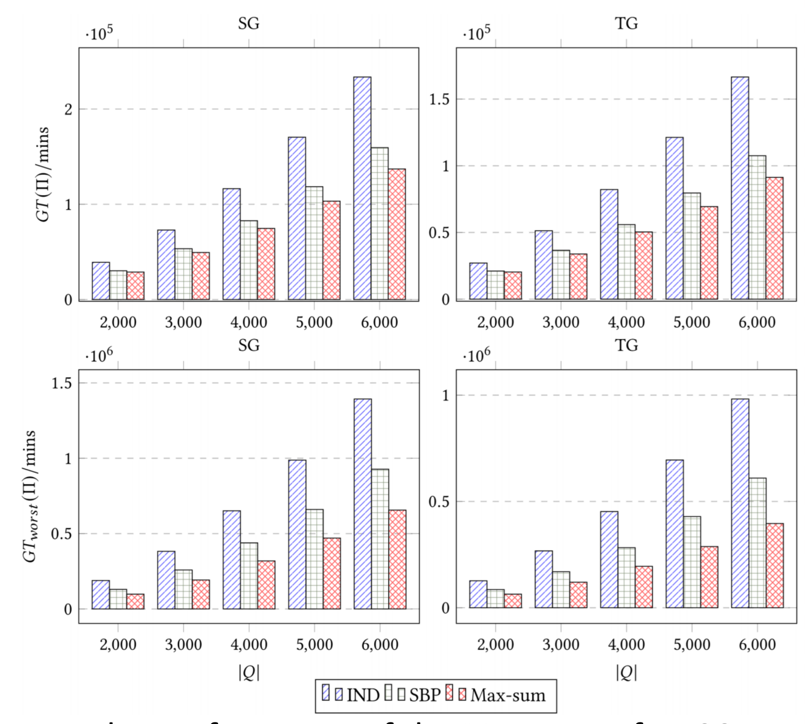
Worst-case travel time: $GT_{worst}(\Pi)$ as an upper bound of the total travel time for all vehicles caused by queries.

Baselines:

IND: Individual-based search algorithm. [7]

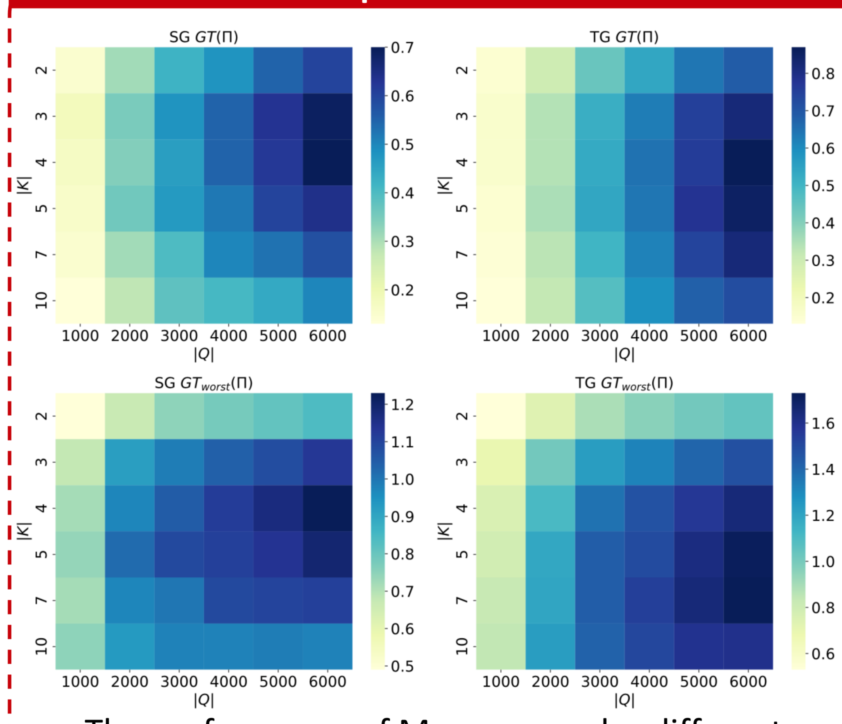
SBP: Self-aware batch process, the SOTA. [2]

Experiment 1



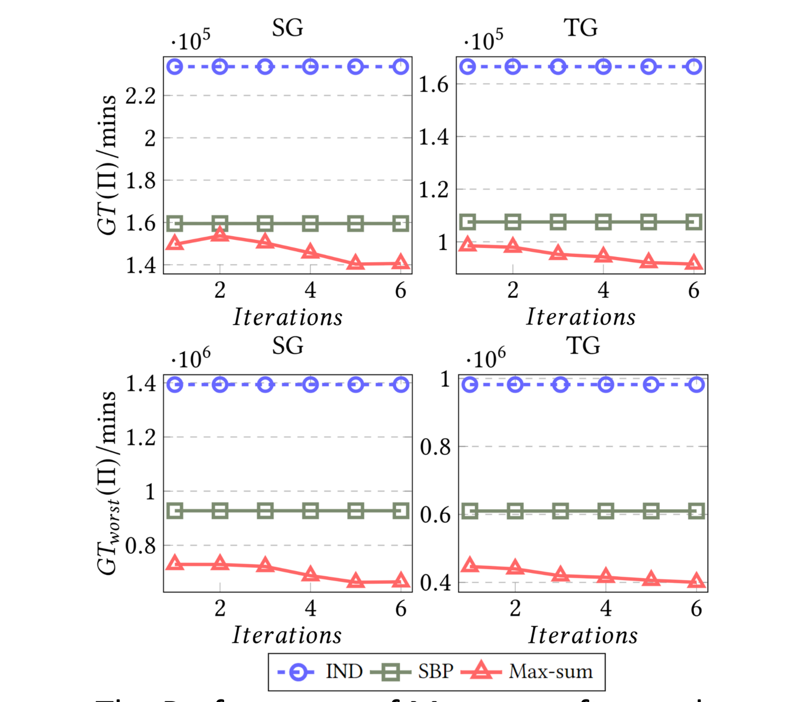
The performance of the Max-sum after 20 iterations **under different numbers of queries.**

Experiment 3



The performance of Max-sum under different numbers of candidate routes and queries.

Experiment 2



The Performance of Max-sum after each iteration when query count is 6000.

Experiment 4

Table 1: Effect of training set size

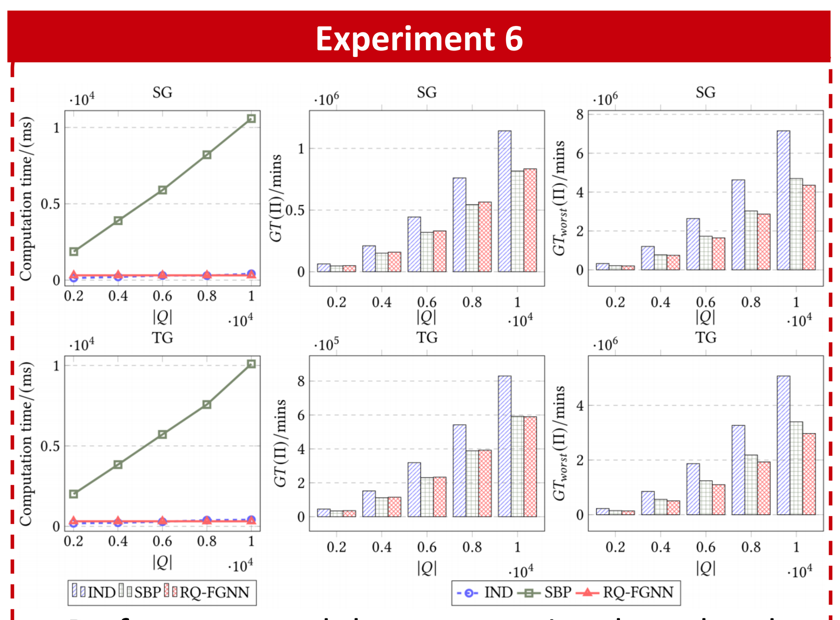
Maps	*Methods*	*200*	*400*	*600*	*800*	*1000*
		Number of samples				
SG	GNN	61.35%	73.20%	80.56%	79.29%	82.70%
	RQ-FGNN	64.19%	77.25%	85.47%	85.72%	88.76%
TG	GNN	63.74%	74.39%	84.45%	85.13%	85.78%
	RQ-FGNN	67.02%	78.91%	89.02%	90.20%	92.53%
Average Gain		**3.06%**	**4.29%**	**4.74%**	**5.75%**	**6.41%**

Experiment 5

Table 2: Performance of models under different K

Maps	*Methods*	*2*	*3*	*4*	*5*	*6*
		Value of K				
SG	GNN	85.24%	82.70%	79.71%	77.34%	71.39%
	RQ-FGNN	89.32%	88.76%	85.12%	82.41%	77.82%
TG	GNN	87.05%	85.78%	80.43%	76.58%	73.31%
	RQ-FGNN	92.11%	92.53%	87.62%	83.36%	78.14%
Average Gain		**4.57%**	**6.41%**	**6.30%**	**5.93%**	**5.63%**

Experiment 6



Performance and the computational overhead of methods under different numbers of queries

[1] Ke Li et al. Towards Alleviating Traffic Congestion: Optimal Route Planning for Massive-Scale Trips. IJCAI'20.
[2] Ke Li et al. Traffic Congestion Alleviation over Dynamic Road Networks: Continuous Optimal Route Combination for Trip Query Streams. IJCAI'21.
[3] Guiyang Luo et al. AlphaRoute: Large-Scale Coordinated Route Planning via Monte Carlo Tree Search. AAAI'23.
[4] Liel Cohen et al. Governing convergence of Max-sum on DCOPs through damping and splitting. Artificial Intelligence 279, 103212, 2020.
[5] Zhen Zhang et al. Factor Graph Neural Networks. NeurIPS'20.
[6] https://users.cs.utah.edu/ lifeifei/SpatialDataset.htm.
[7] Jiajie Xu et al. Traffic aware route planning in dynamic road networks. DASFAA'12.