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

东南大学 SOUTHEAST UNIVERSITY

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Speaker: Yixuan Li

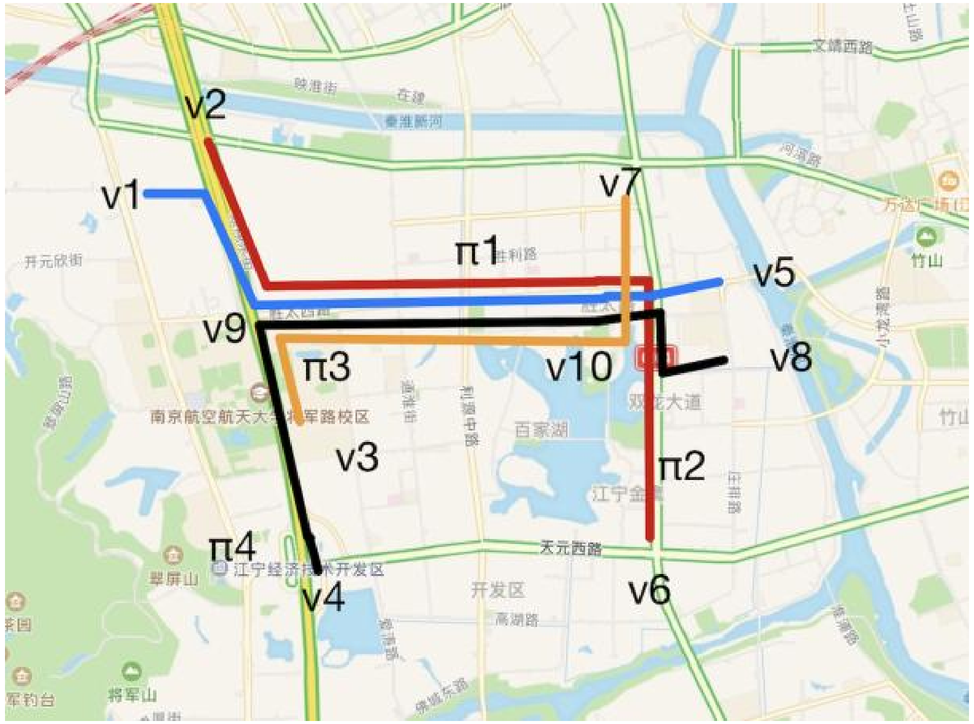
Supervisor: Wanyuan Wang

8 May 2024, Auckland, New Zealand

AAMAS 2024 AUCKLAND NZ

Backgrounds

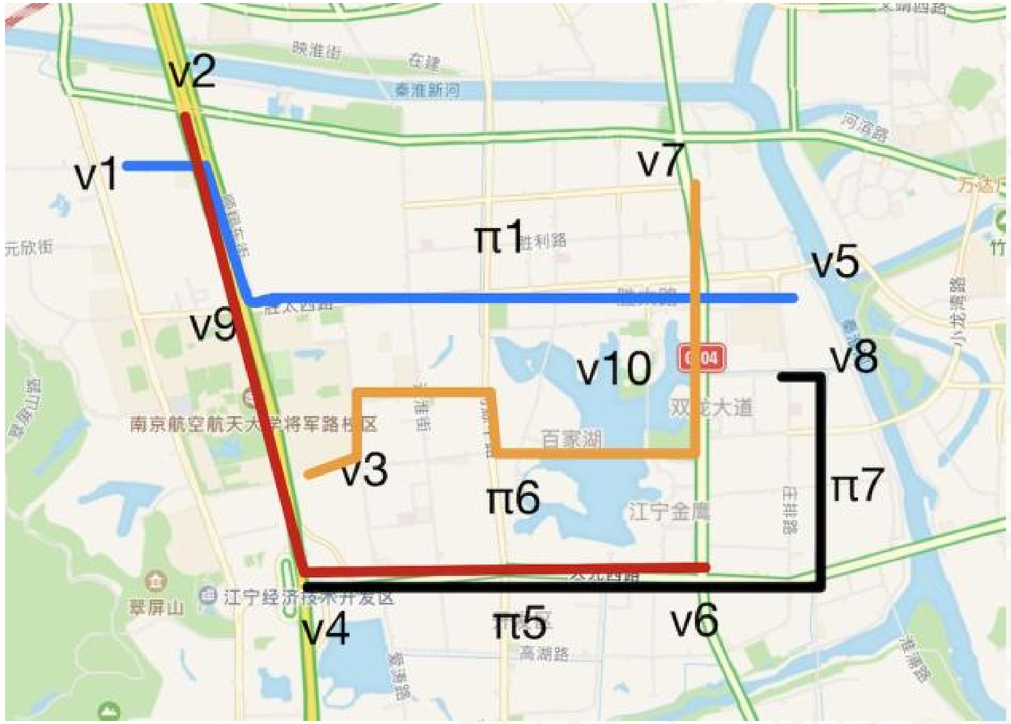
Individual Route Planning



- Individual route planning only considers the **current** traffic condition, ignoring other queries.
- Individual planning will result in four paths going through the same road and causing **congestion**.

VS

Global Route Planning



- Global route planning considers the **potential** vehicles generated by other user queries.
- Global route planning can effectively avoid potential congestion

- In large-scale scenarios, there may be **hundreds of user queries** arriving at the same time every second.
- The goal of Global Route Planning is to **minimise the total travel time** for all the user queries.

Global Route Planning

Problem Definition:

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

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

$t_{min}(e)$ is the travel time without vehicles

f_e is 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** $\boldsymbol{\Pi}$:

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

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: however, current methods often fail to balance **efficiency** and **accuracy**

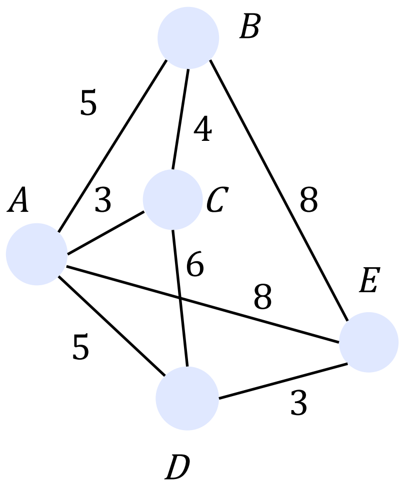
Our Methods:

- **Fast:** returning high-quality approximate solutions in a controllable time
- **Distributed:** decomposing global problems into parts
- **Faster:** solving in milliseconds with neural networks

[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.

Contribution 1: Problem Formulation

Example:



Road Map

$q_1: A \rightarrow D$

$q_2: A \rightarrow B$

$q_3: A \rightarrow C$

$q_4: B \rightarrow E$

Queries

Minimizing total travel time for 4 users

Top-k

k=2

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

$A \rightarrow D, A \rightarrow C \rightarrow D$

$A \rightarrow B, A \rightarrow C \rightarrow B$

$A \rightarrow C, A \rightarrow B \rightarrow C$

$B \rightarrow E, B \rightarrow C \rightarrow D \rightarrow E$

Decision variable: candidate routes

To simplify the problem, the candidate routes are the **shortest k** ones

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

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

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

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

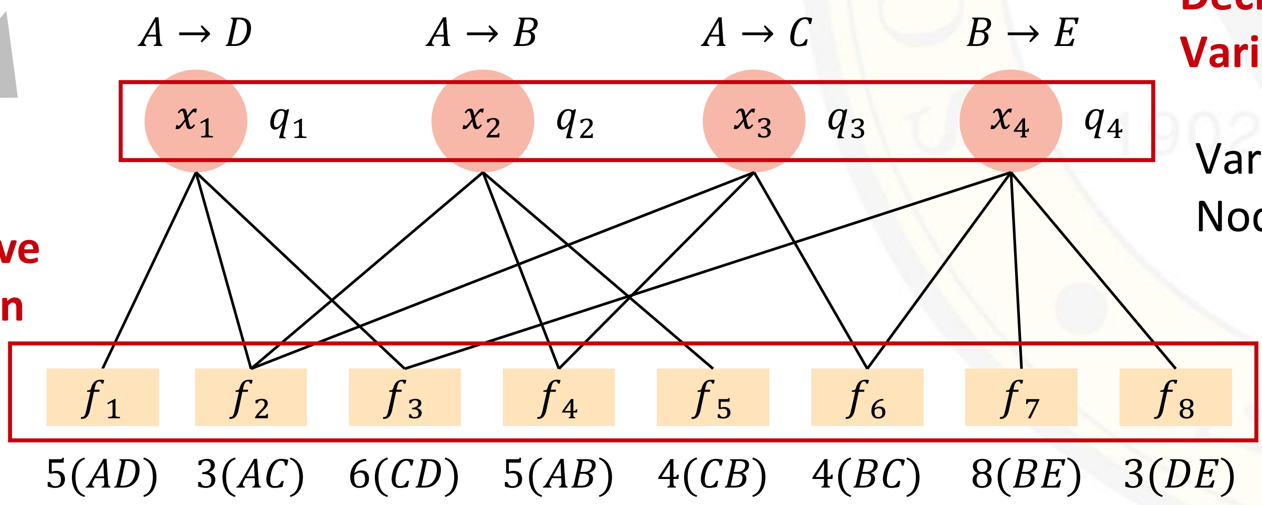
$$F = f_1, ..., f_8$$

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

Distributed Constrained Optimization Problem (DCOP)

Decision Variable

Objective Function



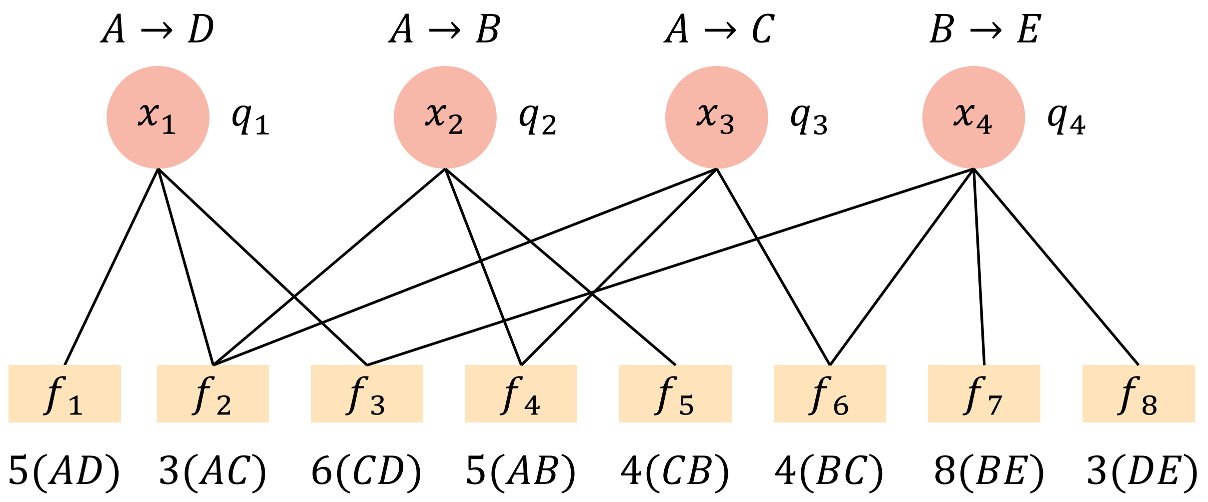
Variable Nodes

Function Nodes

Route-query Factor Graph

Max-Sum Algorithm

Route-query Factor Graph



Solving by message passing algorithm: Max-Sum [1]
Adapted from the belief propagation algorithm: Max-Product

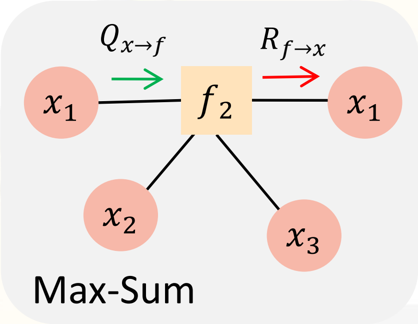
Advantages:
Parallel computation: each node as a computational unit
Distributed computing to rapidly solve and protect privacy
Controllable: can stop after any iteration and return results

[1] Liel Cohen et al. Governing convergence of Max-sum on DCOPs through damping and splitting. Artificial Intelligence 279, 103212, 2020.

Max-Sum Algorithm

Based on the **inference** and **message passing**

The Process of Max-Sum:



1. **Query:** variable nodes send messages to the function nodes (telling which road to take)
2. **Response:** function nodes send response messages to the variable nodes (calculating the maximum possible cost caused by query nodes)
3. **Decision:** The variable nodes use the information to make decision (choosing their value that minimises their travel time)

Contribution 2: Threshold-based Pruning for Max-Sum

Process of Max-Sum

1. **Query Message:**

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

2. **Response Message:**

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

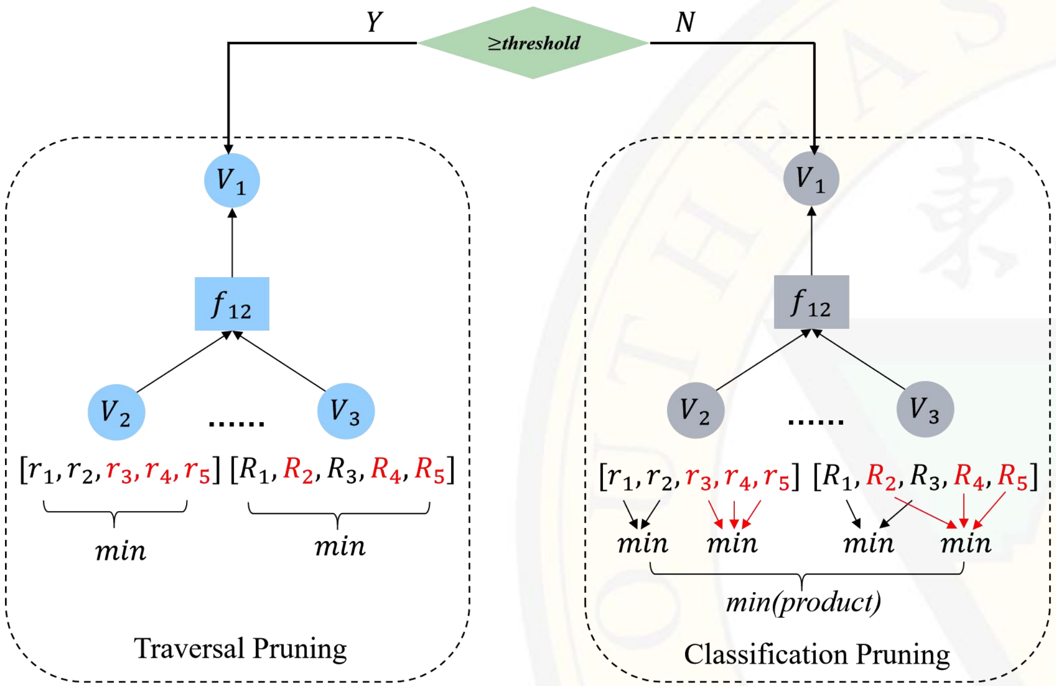
3. **Decision:**

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

Exponential complexity:

The response message need to consider all the possible combinations of neighbours, if the number of query nodes adjacent to the road node is N, and there are K candidates for each query, then the computational complexity is $O(K^N)$

Threshold-based Pruning:



Classification Pruning: The query can be divided into two categories: **passing through** the road or not

Finding the optimal value from the two types **respectively**.

Computational complexity: $O(2^N)$ without loss of accuracy

Traversal pruning: greedy method, complexity of $O(KN)$

Experiments

Experimental Settings

Datasets:

TG: Real-world San Joaquin County Road Network[1], with scale reduction to 2000 vertices and 4958 edges.

SG: Synthetic road network with 2000 vertices and 3771 edges.

Metrics:

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

Worst-case travel time: $GT_{worst}(\Pi)$ as an auxiliary metric. Removes uncertainty in arrival times and represents the upper bound of the total travel time for all vehicles caused by queries.

Baselines:

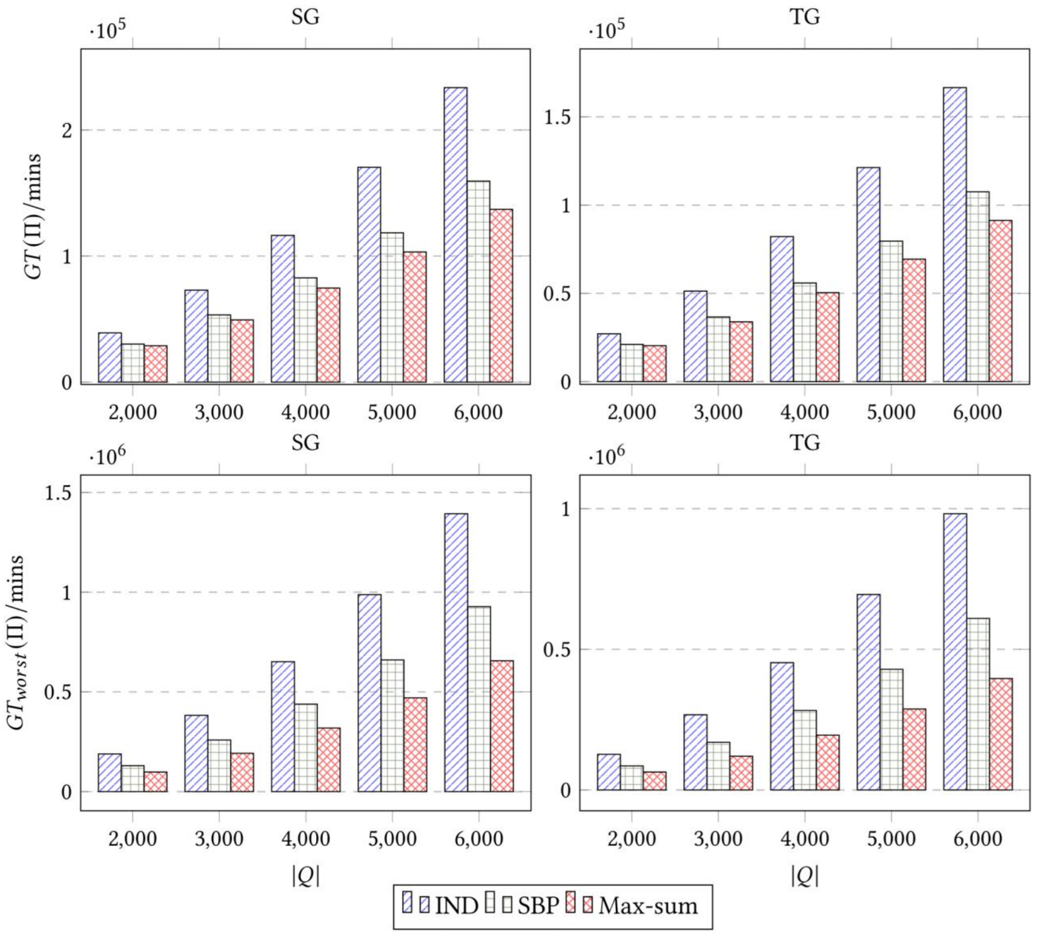
IND: Individual-based search algorithm. [1]

SBP: Self-aware batch process, the state-of-the-art method. [2] Consists of an initial greedy-based search with refining steps.

Experiment 1:

How does the Max-sum algorithm **perform** with different numbers of users?

Experiment 1



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

[1] https://users.cs.utah.edu/ lifeifei/SpatialDataset.htm

[1] Jiajie Xu et al. Traffic aware route planning in dynamic road networks. DASFAA'12

[2] Ke Li et al. Traffic Congestion Alleviation over Dynamic Road Networks: Continuous Optimal Route Combination for Trip Query Streams. IJCAI'21

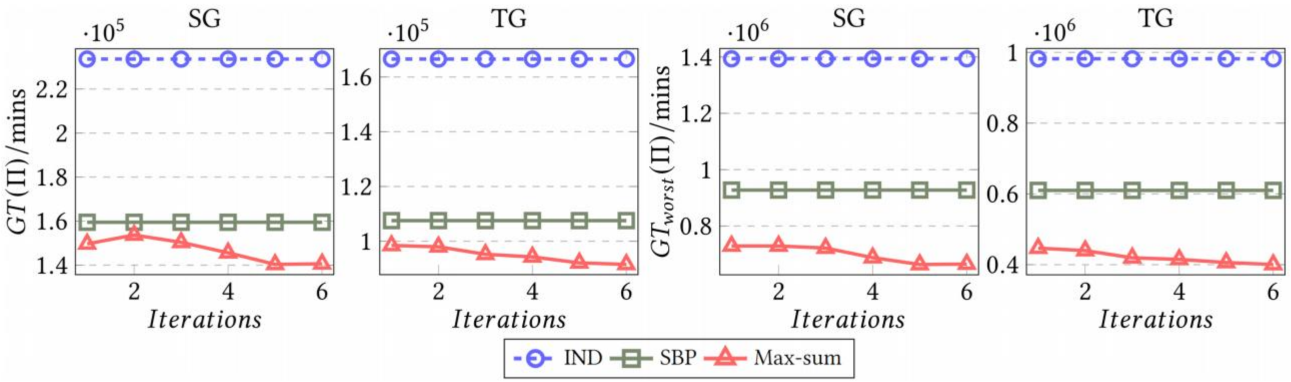
Experiments

Experimental Settings

Experiment 2:
How **efficient** is the max-sum algorithm compared to the baseline approach?

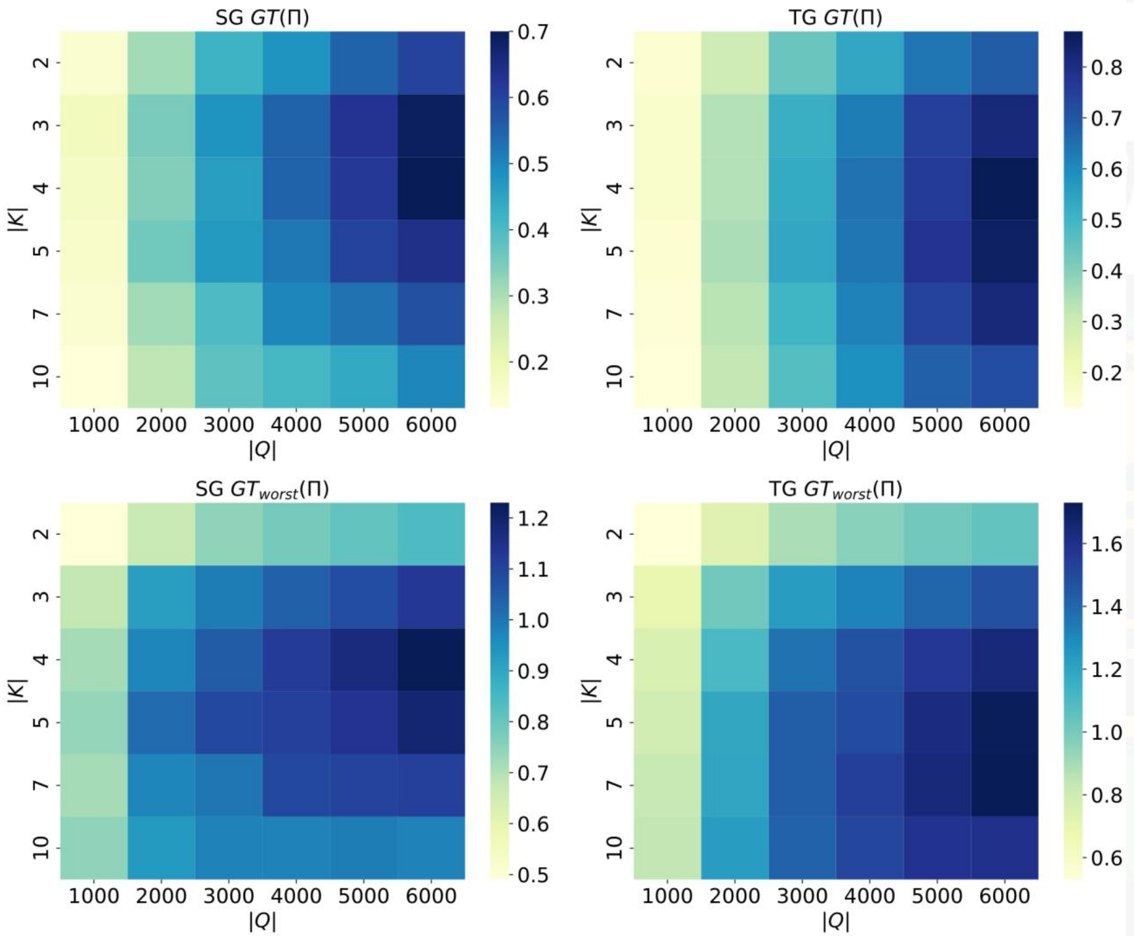
Experiment 3:
How does the max-sum algorithm perform with different **hyperparameters** K and different numbers of users?

Experiment 2



The Performance of Max-sum after each iteration when query count is 6000. (without parallel or distributed computing)

Experiment 3



The performance of Max-sum under different numbers of candidate routes and queries after 20 iterations.

Even Faster? Neural Networks!

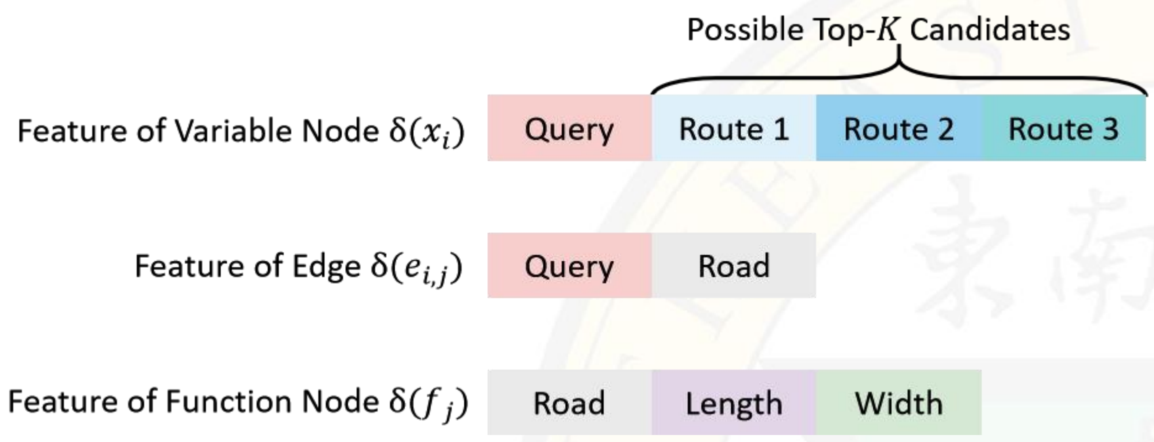
Motivation

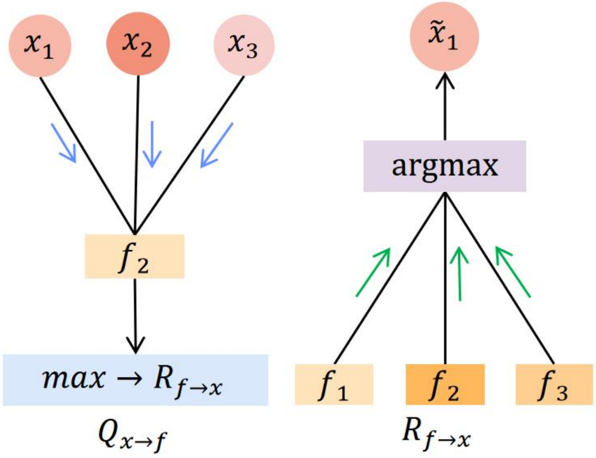
Historical Information: many closed-related GRP instances must be solved repeatedly.

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 [1].

The idea of learning the input-output mapping of the CO problem is promising for both optimality and scalability.

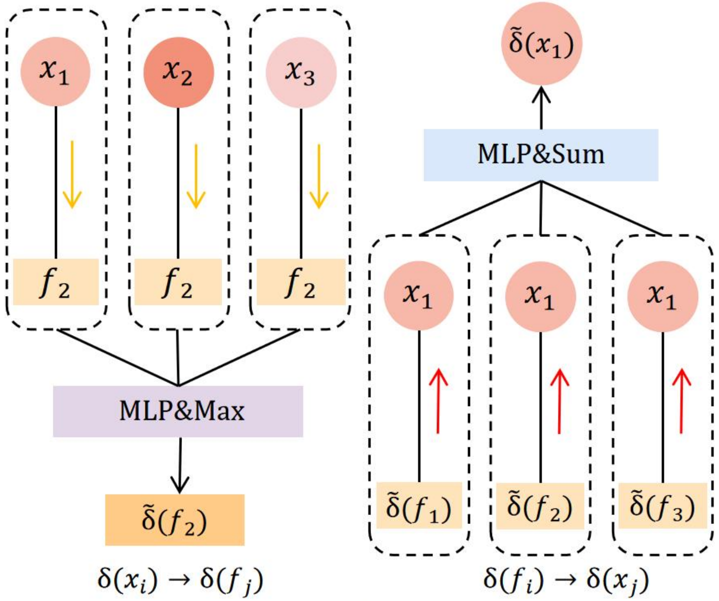




Max-sum

Message	Max-sum	RQ-FGNN
$V \rightarrow F$	$\sum_{Neg(x)} R_{f\rightarrow x} + \alpha$	$\max_{Neg(f)} \varphi_{V\rightarrow F}(\delta(e), \delta(x), \delta(f))$
$F \rightarrow V$	$\max f(Neg(f)) + \sum_{Neg(f)} Q_{x\rightarrow f}$	$\sum_{Neg(x)} \varphi_{F\rightarrow V}(\delta(e), \delta(x), \delta(f))$
Assignment $\tilde{V}$	$\operatorname{argmin} \sum_{Neg(f)} Q_{x\rightarrow f}$	$\operatorname{softmax}(\varphi(\delta(x)))$

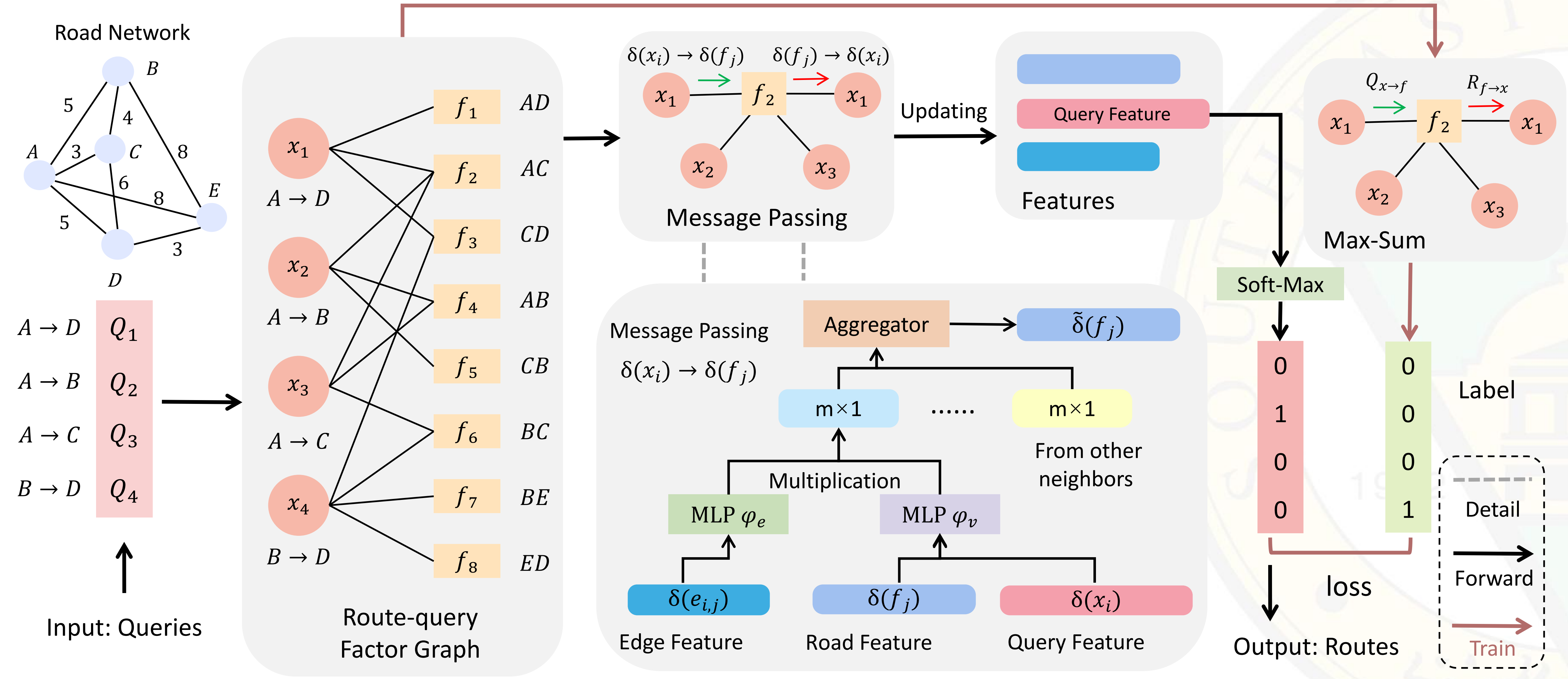
Massage Passing



RQ-FGNN

[1] Zhen Zhang, Fan Wu, and Wee Sun Lee. Factor Graph Neural Networks. NeurIPS'20.

Contribution 3: Route-Query Factor Graph Neural Network



Experiments

Experiment 4

Table 1: Effect of training set size

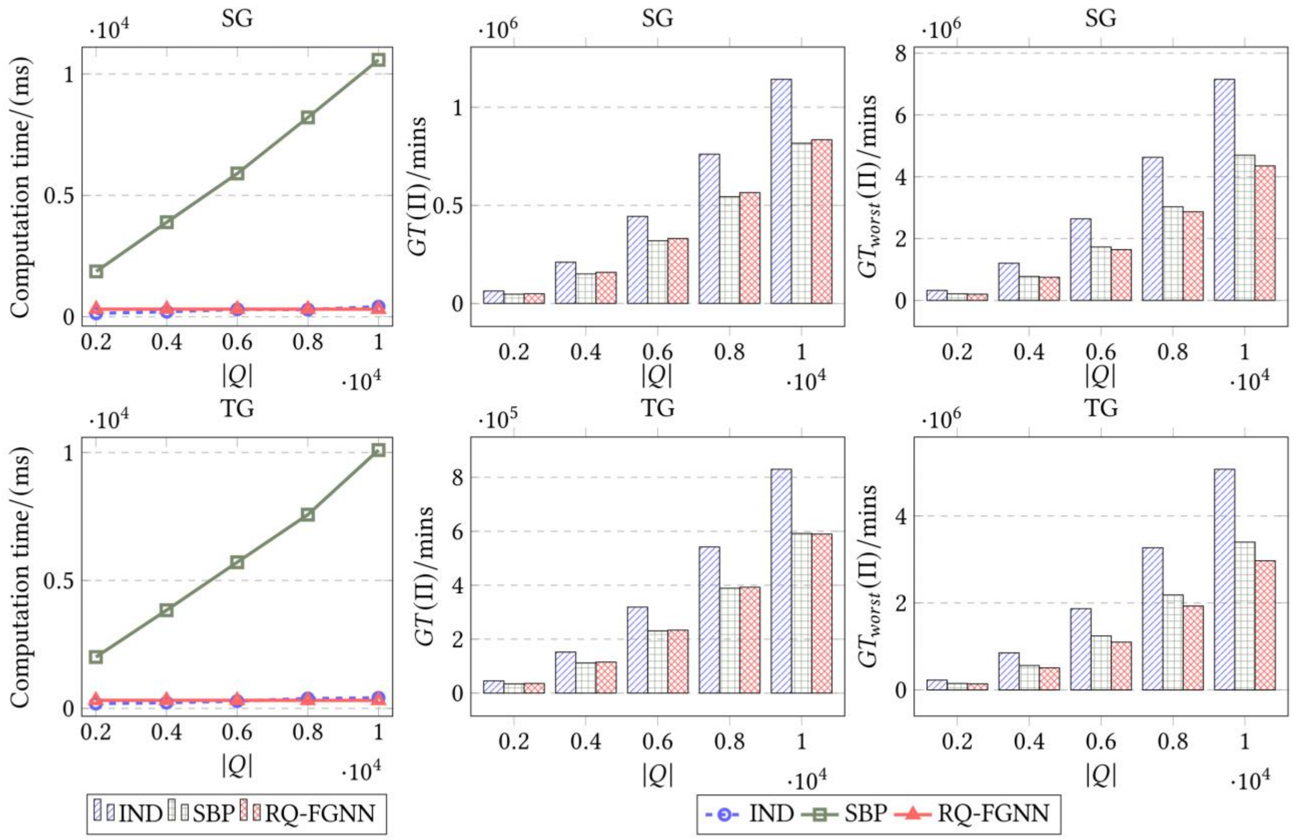
Maps	*Methods*	*Number of samples* 200	400	600	800	1000
SG	GNN	61.35%	73.20%	80.56%	79.29%	82.70%
	RQ-FGNN	64.19%	77.25%	85.47%	85.72%	88.76%
TG	GNN	63.74%	74.39%	84.45%	85.13%	85.78%
	RQ-FGNN	67.02%	78.91%	89.02%	90.20%	92.53%
Average Gain		**3.06%**	**4.29%**	**4.74%**	**5.75%**	**6.41%**

Experiment 5

Table 2: Performance of models under different K

Maps	*Methods*	*Value of K* 2	3	4	5	6
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 corresponding to different numbers of queries

Conclusion and Future Work

Conclusion

Proposed a graph model named the **route-query factor graph** for the GRP problem.

Applied a damped Max-sum method and design a **hybrid pruning** approach based on the characteristics of the GRP problem, which can return high-quality solutions.

Further devised an **end to end** message-passing route-query factor graph neural network to **parameterize** the Max-sum.

Future Work

Explore the **theoretical guarantee** of RQ-FGNN.

Migration application: accelerate integer programming (IP).

Extending Top-k candidates: variable network structure.

Distributed neural networks for faster online inference.

Contact Information

I am Yixuan, a second-year graduate student at Department of Computer Science and Engineering, Southeast University, advisored by Prof. Wanyuan Wang.

My research interest lies in Reinforcement Learning, Operation Research and Multi-Agent Systems.

Please feel free to contact me if interested in my work.

My email: Yixuanli@seu.edu.cn

THANKS！

Questions&Comments?