KDD Cup & DiDi • $ 30,000 • 944 Single-Player Team • 66 Multi-Player Team • 1220 participants

KDD CUP 2020 KDD CUP 2020: Learning to Dispatch and Reposition on a Mobility-on-Demand Platform

DiDi × KDD2020

2020-07-09 - Team Merger Deadline

2020-04-02 - Launch

2020-07-17 - Close

KDD CUP 2020 LDR
Dispatch Winner Report

Team: Polar Bear

Members: Yansheng Wang
Dingyuan Shi
Yuanfei Luo
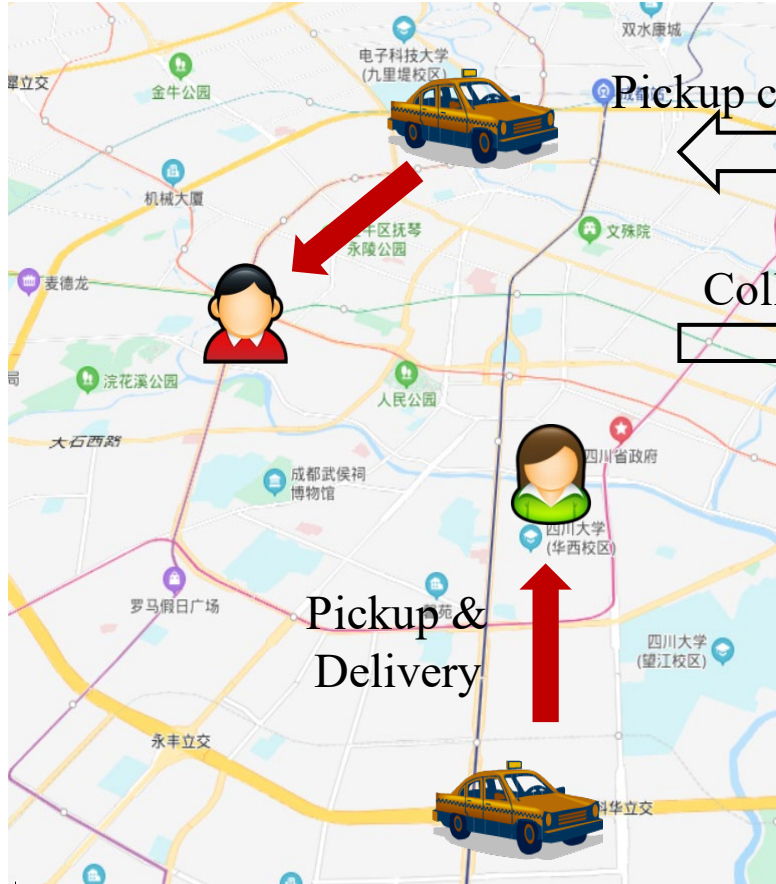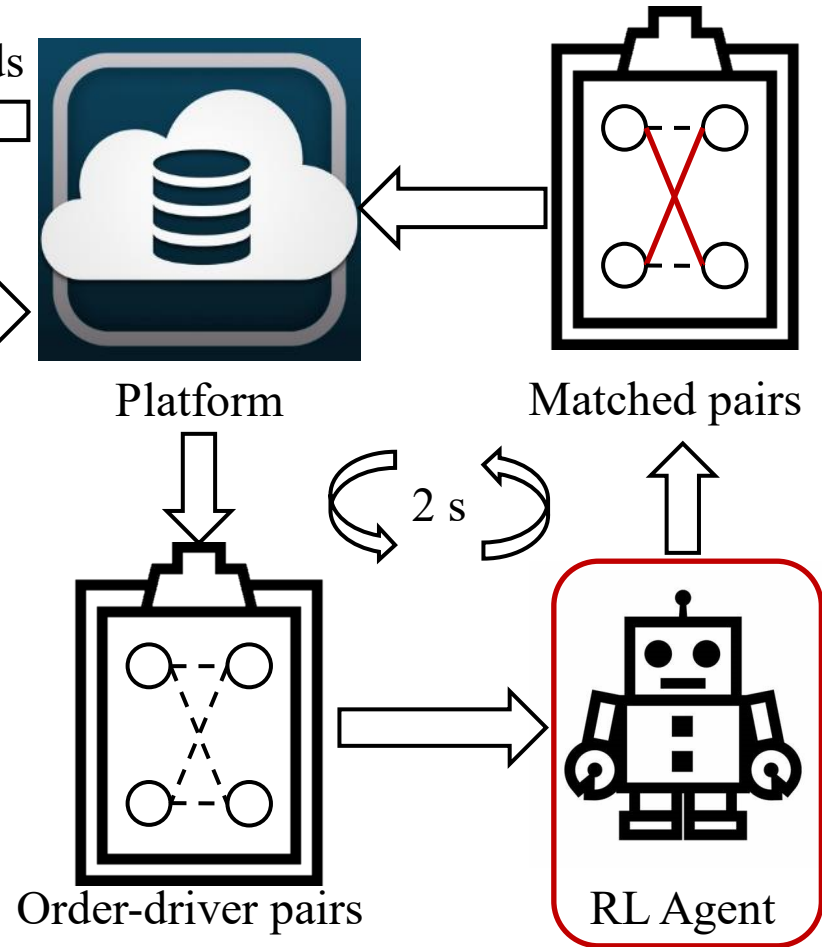Maoxiaomin Peng
Yi Xu
Yongxin Tong

Problem

Pickup commands

Collect data

Platform

Matched pairs

2 s

Pickup & Delivery

Historical orders and simulated drivers

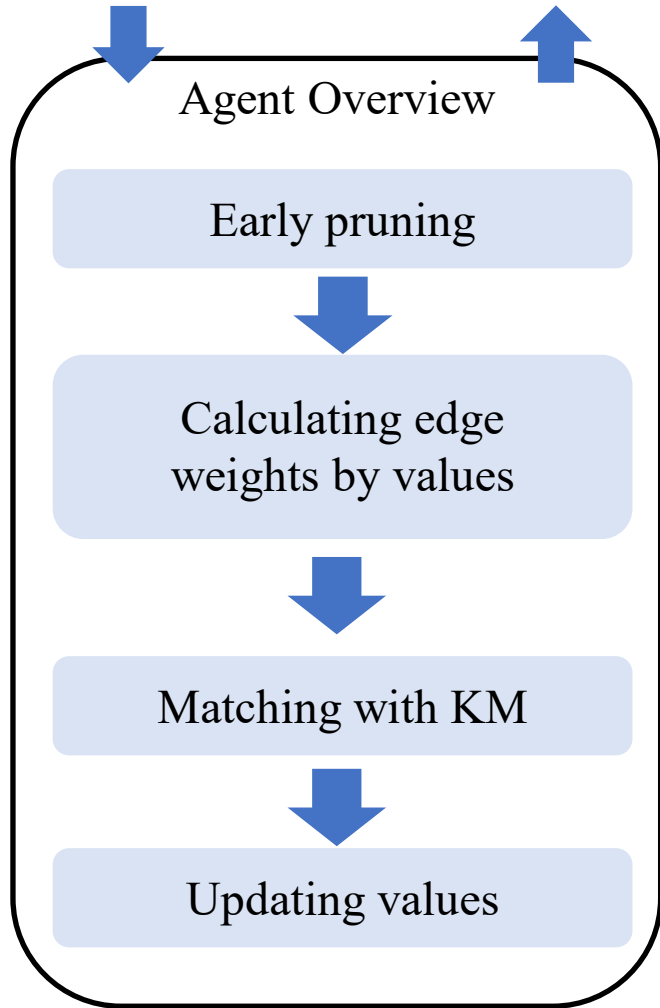Order-driver pairs

RL Agent

Objective: Maximize Mean Total Driver Income

Solution

Unmatched pairs   Matched pairs

Agent Overview

Early pruning

↓

Calculating edge weights by values

↓

Matching with KM

↓

Updating values

Solution

Unmatched pairs   Matched pairs

Agent Overview

Early pruning

Calculating edge weights by values

Matching with KM

Updating values

Top K nearest drivers

Original Graph   →   Pruned Graph

Solution

Unmatched pairs   Matched pairs

Agent Overview

Early pruning

Calculating edge weights by values

Matching with KM
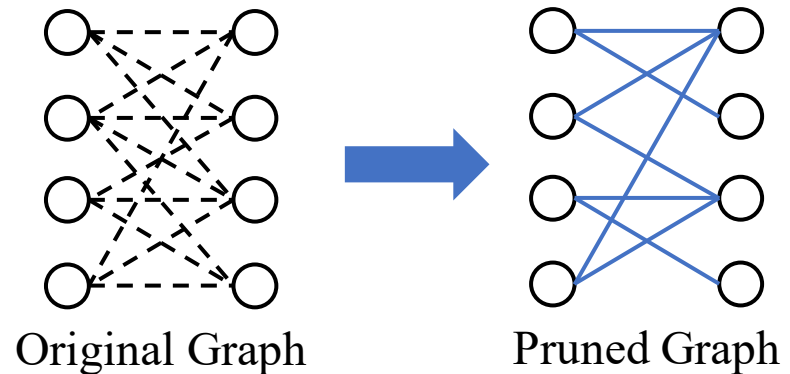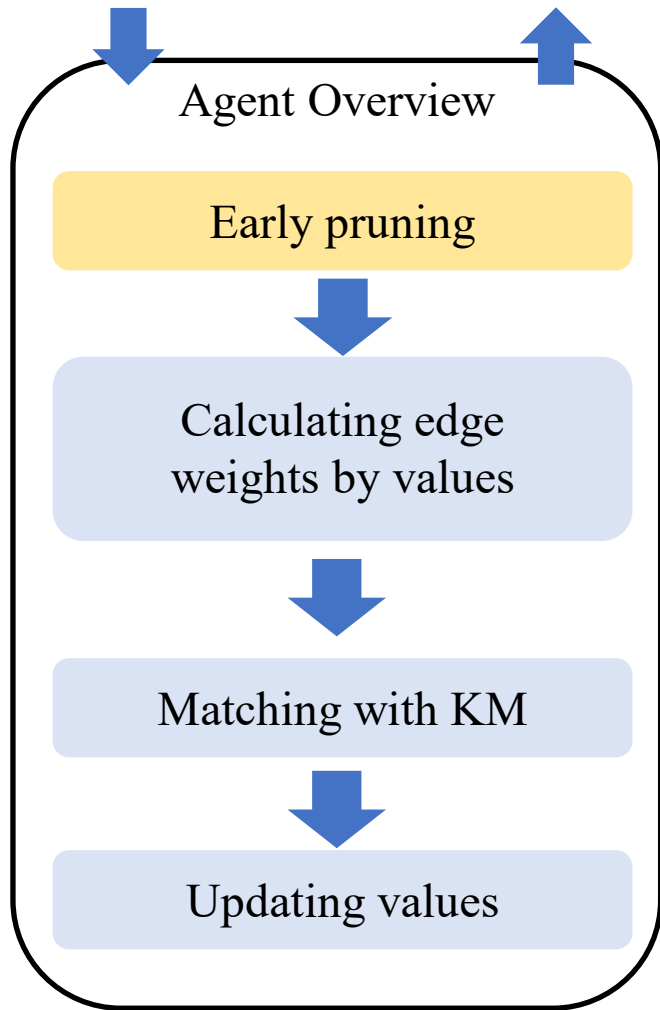
Updating values
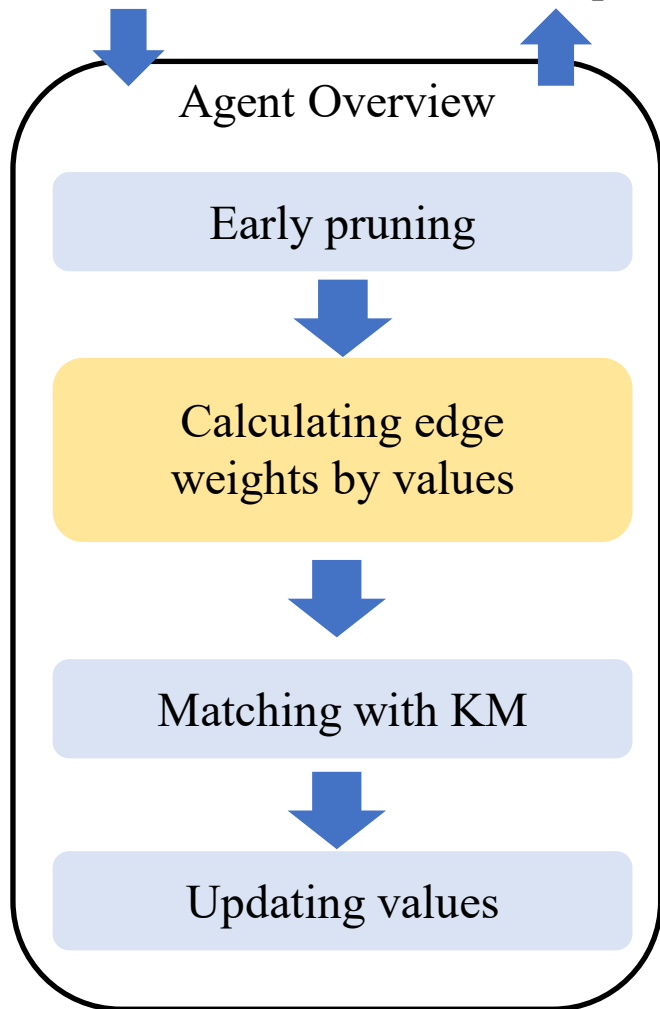
Value representation
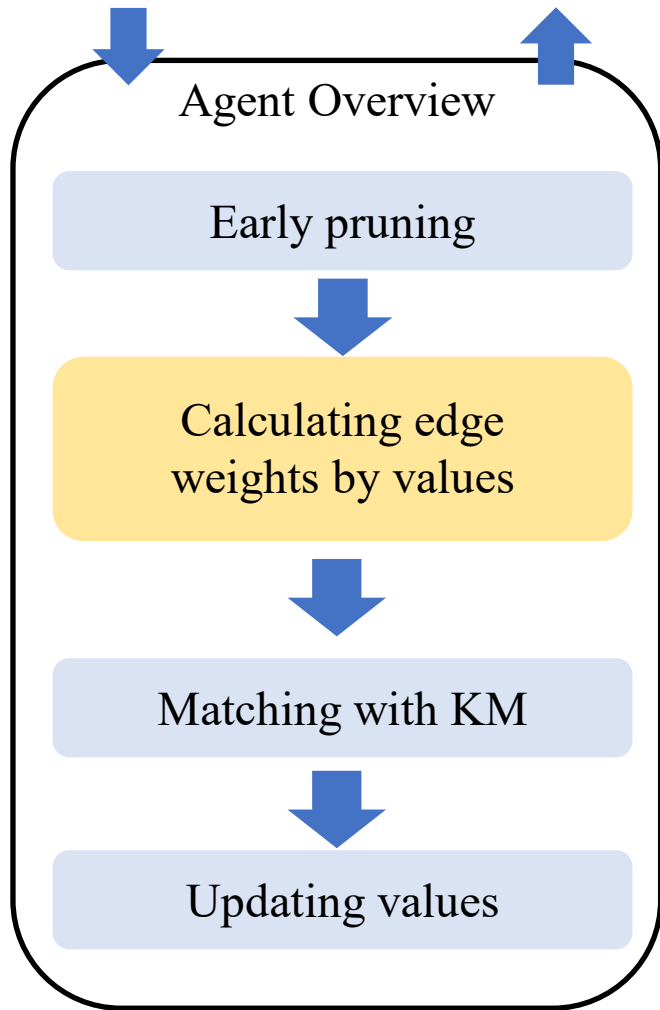
Source: X. Tang et al. KDD 19
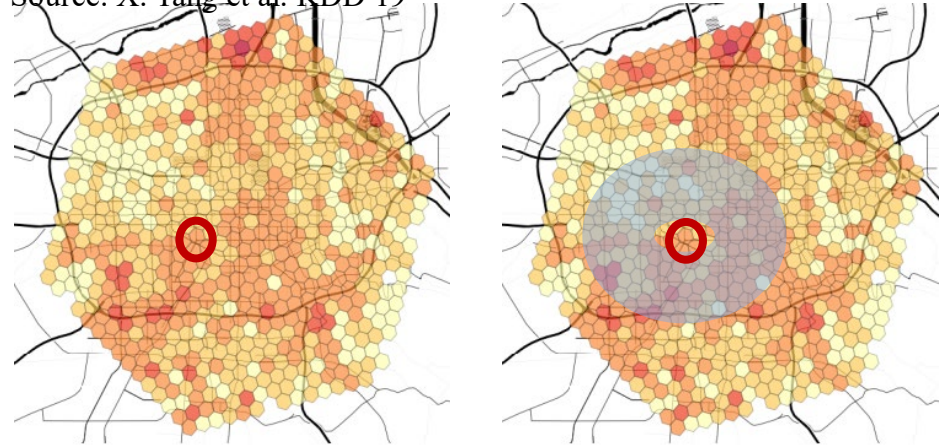
Grid values   Layer values

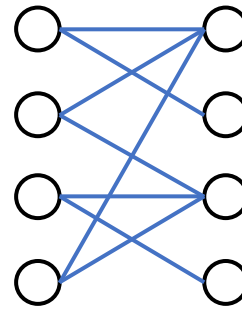Grid id → value   Grid id → $\sum$ layer values of grids nearby

$Value = Mean(Grid\ values, Layer\ values)$

Solution

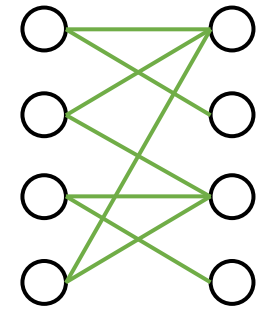Unmatched pairs   Matched pairs

Agent Overview

Early pruning

Calculating edge weights by values

Matching with KM

Updating values

Weight Calculation

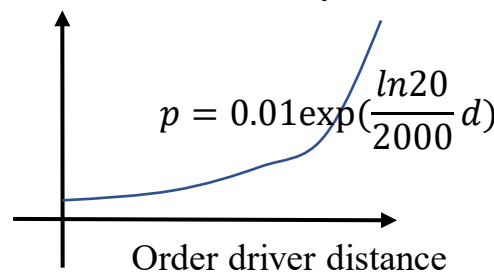Rewards as weights   TD(0) error as weights

$$Expected\ TD(0)\ error$$
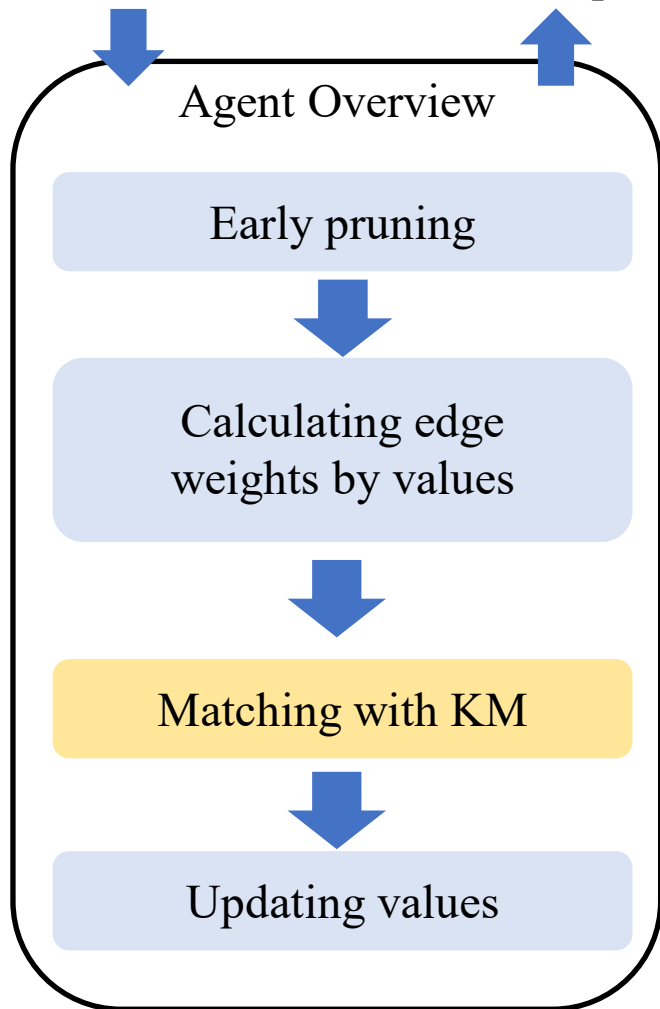
$$U = (1 - p) \cdot (reward + \gamma^t V' - V)$$

Discount factor

Value of driver position

Cancel Probability
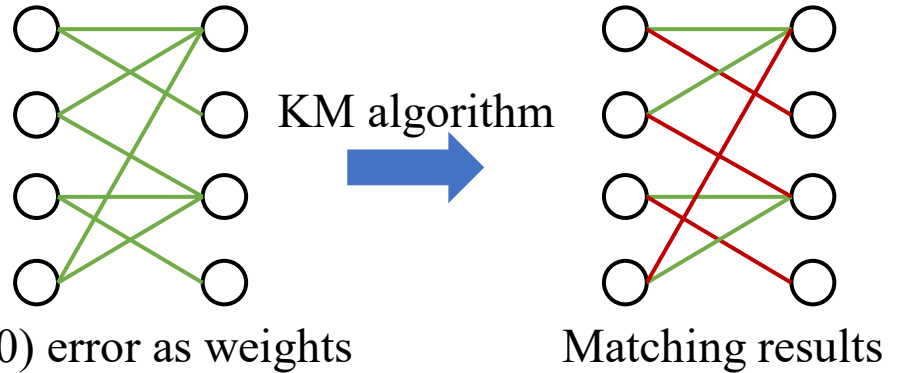
$$p = 0.01\exp(\frac{ln20}{2000} d)$$

Value of order destination

Order driver distance

Solution

Unmatched pairs   Matched pairs

Agent Overview
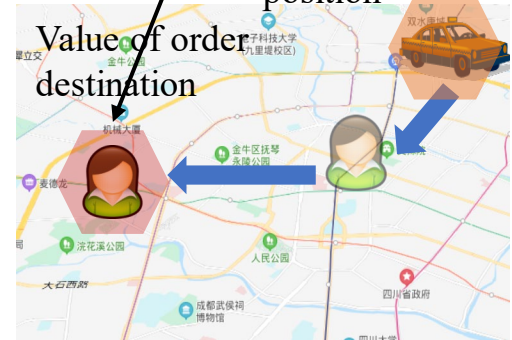
- Early pruning
- Calculating edge weights by values
- Matching with KM
- Updating values

On-policy Decision making

KM algorithm

TD(0) error as weights   Matching results
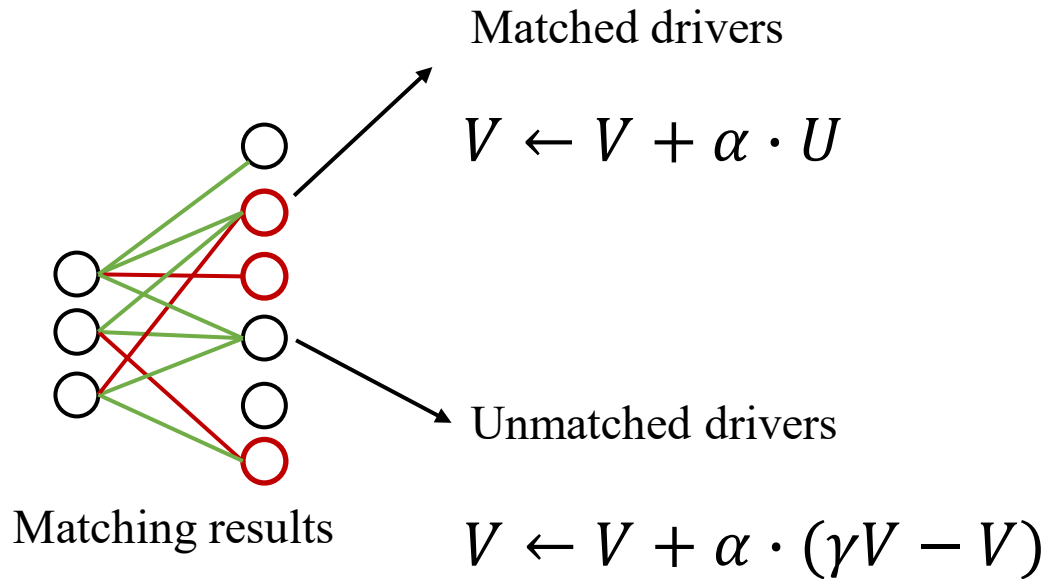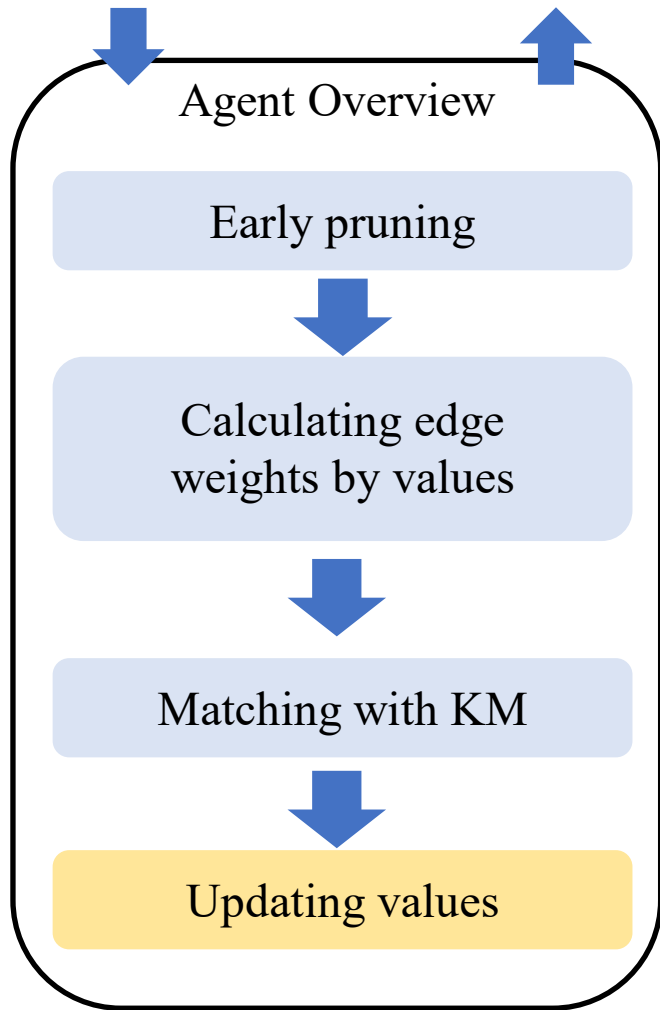
Python implementation: Exceeds runtime limits
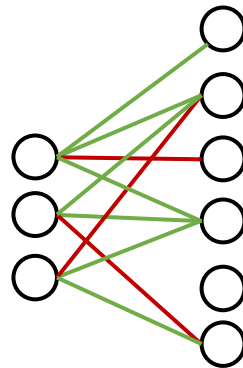
Observation: Imbalanced graph

Implementation of KM for imbalanced graphs with C++

Solution

Unmatched pairs   Matched pairs

Agent Overview

Early pruning

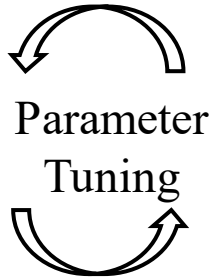Calculating edge weights by values

Matching with KM

Updating values

Matched drivers

$$V \leftarrow V + \alpha \cdot U$$

Matching results

Unmatched drivers

$$V \leftarrow V + \alpha \cdot (\gamma V - V)$$

Solution

Agent Overview

Early pruning

↓

Calculating edge weights by values

↓

Matching with KM

↓

Updating values

Parameter Tuning
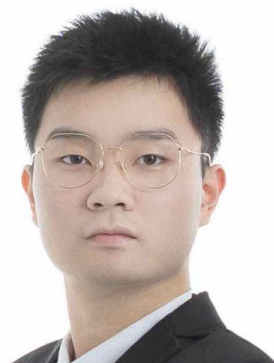
Historical Orders

Driver Simulator

Our Team

Team Leader

Yansheng Wang
Beihang University

Dingyuan Shi
Beihang University

Team Coach

Yongxin Tong
Professor
Beihang University

Yuanfei Luo
4Paradigm Inc.

Maoxiaomin Peng
Beihang University

Yi Xu
Beihang University

Thanks!