Flexible Online Task Assignment in Real-Time Spatial Data

Yongxin Tong [1], Libin Wang[1], Zimu Zhou[2], Bolin Ding[3], Lei Chen[4], Jieping Ye[5], Ke Xu[1]

[1] SKLSDE Lab, Beihang University, Beijing, China, [2] ETH Zurich, Zurich, Switzerland
[3] Microsoft Research, Redmond, WA, USA
[4] The Hong Kong University of Science and Technology, Hong Kong SAR, China
[5] Didi Research Institute, Didi Chuxing, Beijing, China
[1] {yxtong, lbwang, kexu}@buaa.edu.cn, [2] zimu.zhou@tik.ee.ethz.ch, [3] bolind@microsoft.com,
[4] leichen@cse.ust.hk, [5] yejieping@didichuxing.com

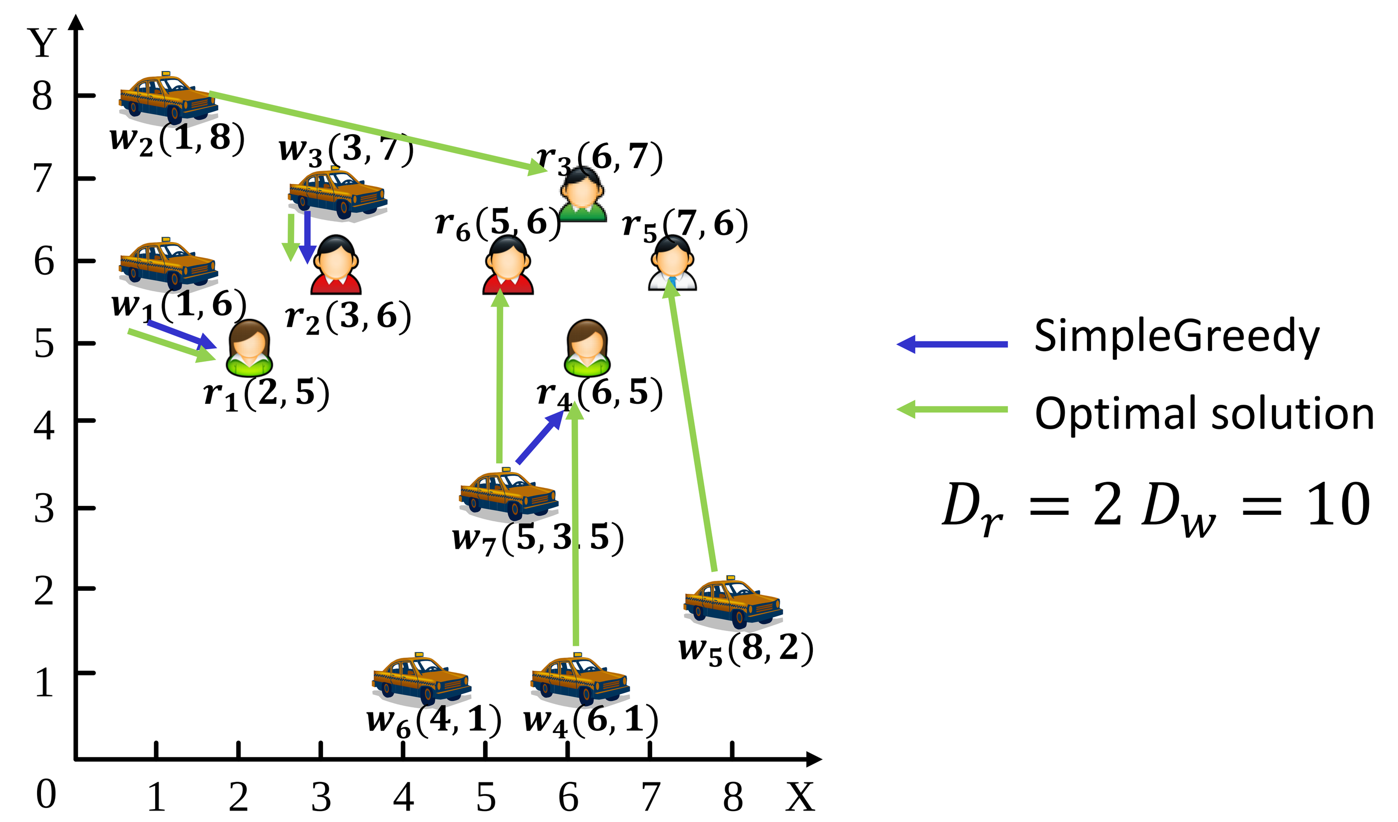
Introduction

- Real-time spatial data is ubiquitous.
  - Online platforms that facilitate spatial tasks to be assigned and performed by workers, e.g. O2O applications.
- Motivation
  - Most O2O applications need to do task assignment in real-time:
    - Real-Time Taxi-Calling Service.
    - Food Delivery Service.
  - The flexibility of workers has not been considered.

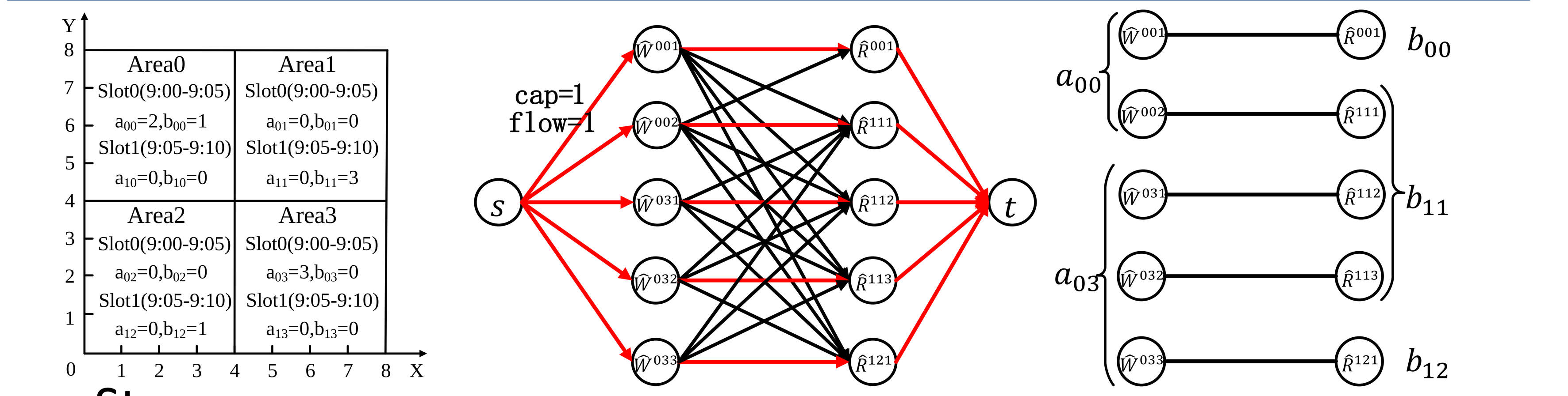
The FTOA Problem

- Given
  - A set of crowd workers W
    - Each $w \in W$: location L_w, arriving time S_w, deadline D_w.
  - A set of spatial tasks R
    - Each $r \in R$: location L_r, arriving time S_r, deadline D_r.
- Find an online allocation M to maximize the assigned pairs *MaxSum(M)*$=\sum_{w\in W, r\in R} I(w,r)$. $I(w,r)=1$ if the following constraints are satisfied.
  - Deadline Constraint.
  - Worker's decision Constraint.
  - Task's decision Constraint.
  - Invariable Constraint: Once a task r is assigned to a worker w, the assigned pairs of (w,r) cannot be changed.
- Online Algorithm Evaluation: Competitive Ratio (CR)
  - IID Model (Stochastic case Analysis):
    - $CR_{iid} = min_{\forall v \in V \ follows\ D_R\ and\ D_W} \frac{MaxSum(M)}{MaxSum(OPT)}$

9:00	9:00	9:01	9:01	9:02	9:03	9:03	9:03	9:04	9:05	9:06	9:07	9:08
w_1	r_1	w_2	w_3	r_2	w_4	w_5	w_6	w_7	r_3	r_4	r_5	r_6

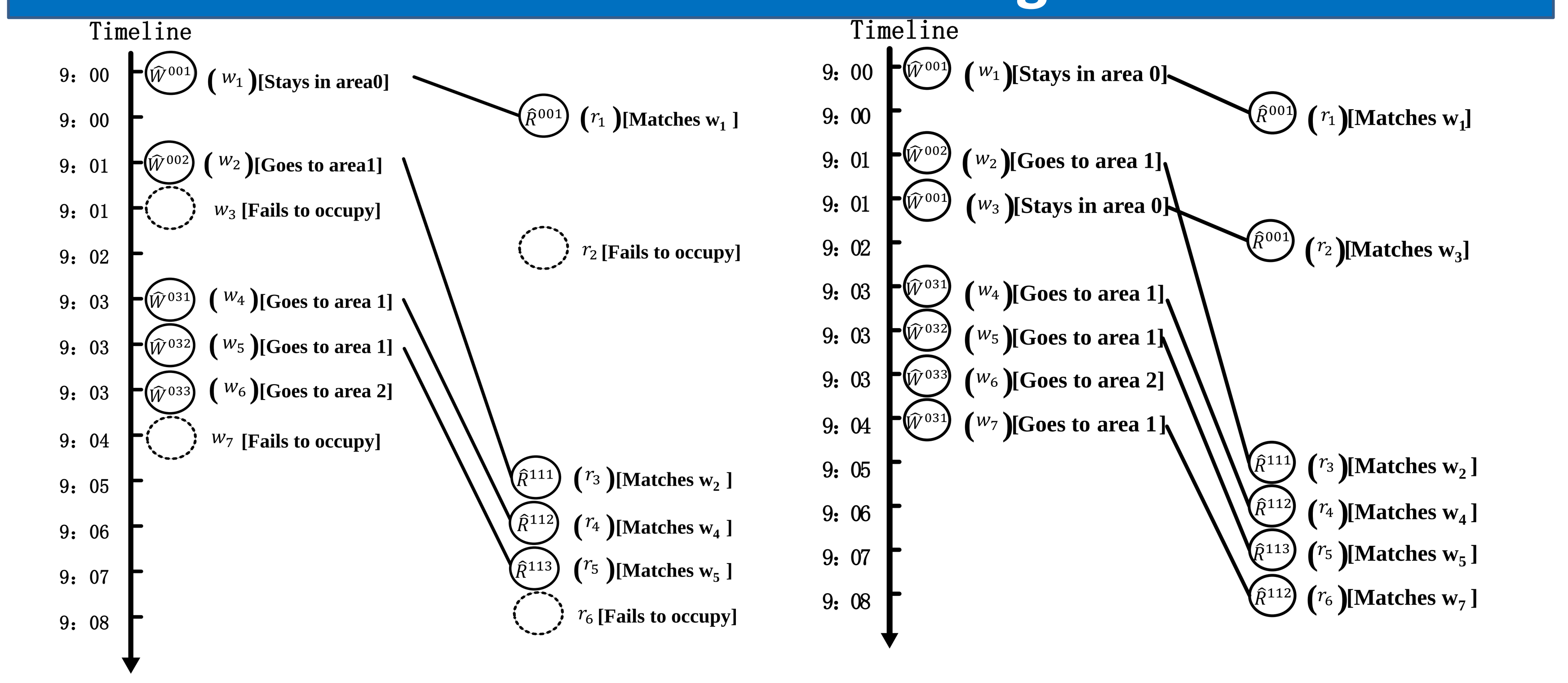


Offline Guide Generation Algorithm



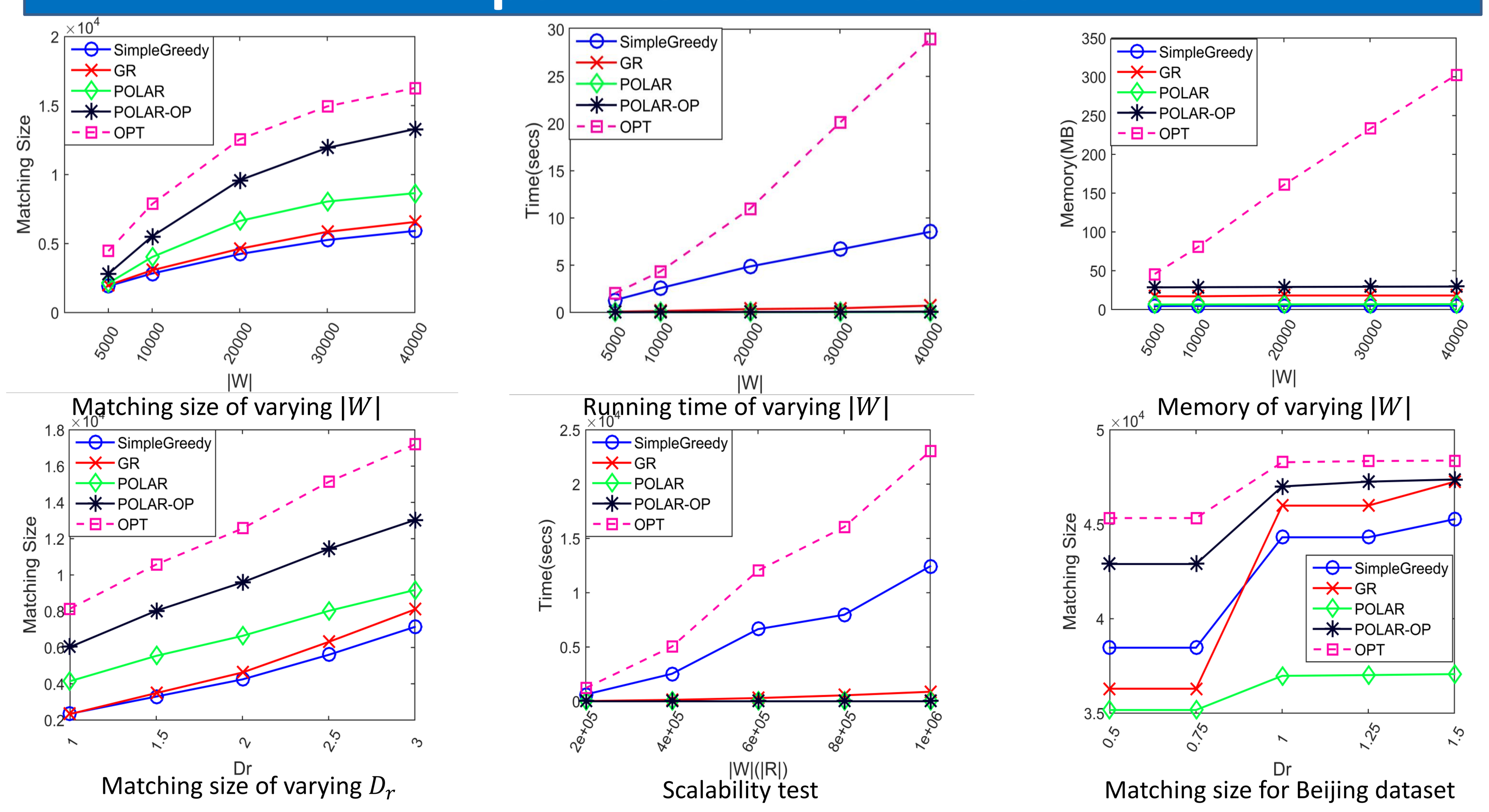
- Steps
  - 1. Construct the bipartite graph according to the prediction.
  - 2. Run the network algorithm to generate the guide.

POLAR and POLAR-OP Algorithm



- POLAR: $CR_{iid} = 0.4$
  - 1. When a new object arrives, occupy a node $w(r)$ in the offline guide.
  - 2. Find the neighbor node $r(w)$ in the offline guide.
  - 3. Match the object who occupies the neighbor node $r(w)$.
- POLAR-OP: $CR_{iid} = 0.47$
  - 1. When a new object arrives, associate the object to its corresponding node $w(r)$ in the offline guide.
  - 2. Find the neighbor node $r(w)$ in the offline guide.
  - 3. Match the object associated to the neighbor node $r(w)$.

Experimental Evaluation



Acknowledgment

Yongxin Tong's work is supported in part by National Grand Fundamental Research 973 Program of China under Grant 2014CB340300, NSFC Grant No. 61502021 and 71531001, and SKLSDE Open Program SKLSDE-2016ZX-13.
Lei Chen's work is supported in part by the Hong Kong RGC Project 16202215, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, NSFC Grant No. 61328202 and 61300031, Microsoft Research Asia Collaborative Grant and NSFC Guang Dong Grant No. U1301253.