众包与群智计算

一种共享经济时代的新型计算范式

童咏昕

计算机学院 软件开发环境国家重点实验室

yxtong@buaa.edu.cn

个人简介

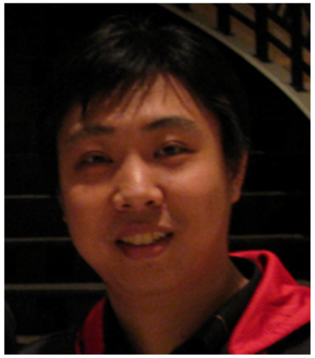
- **北京航空航天大学(目前)**
  - 计算机学院，“卓越百人计划”副教授
  - 软件开发环境国家重点实验室
  - 从事大数据与众包计算的研究
- **香港科技大学(2010年-2015年)**
  - 计算机科学与工程系，研究助理教授(2014-2015)
    - 从事大数据挖掘分析与众包计算的研究
  - 计算机科学与工程系，博士(2010-2014)
    - 从事海量不确定数据管理与挖掘的研究

个人简介

- **联系方式**
  - 办公室：主校区柏彦大厦18层1806
  - 电子邮箱： yxtong@buaa.edu.cn;
  - 个人主页：http://www.nlsde.buaa.edu.cn/~yxtong/
    http://www.cse.ust.hk/~yxtong/

个人简介

www.nlsde.buaa.edu.cn/~yxtong/



Yongxin Tong 童 咏 昕

Associate Professor
State Key Laboratory of Software Development Environment
School of Computer Science and Engineering
Beihang University (BUAA)

Office: Room 1806, Baiyan Building

E-mail: yxtong AT buaa.edu.cn or yongxintong AT gmail.com

[**Short Bio**] [**Research**] [**Publications**] [**Awards**] [**Experiences**] [**Misc**]

Short Biography

Yongxin Tong is an Associate Professor in the State Key Laboratory of Software Development Environment (SKLSDE) of the School of Computer Science and Engineering at Beihang University (BUAA). He received a Ph.D. degree in Computing Science and Engineering from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), under Dr. Lei Chen's supervision. He also received a Master degree in Software Engineering at Beihang University and a Double Bachelor degree in Economics from China Centre for Economic Research (CCER) at Peking University.

Research Interests

- Crowdsourcing
- Uncertain Data Mining and Management
- Social Network Analysis

Overview Slides on My Recent Research Topics

- Spatio-temporal Crowdsourcing

Selected Publications [My DBLP Entry] [Full Publication List]

- **Yongxin Tong**, Jieying She, Bolin Ding, Lei Chen, Tianyu Wo, Ke Xu. "Online Minimum Matching in Real-Time Spatial Data: Experiments and Analysis", to appear in *Proceedings of the 42nd International Conference on Very Large Databases* (**VLDB 2016**), New Delhi, India, September 5-9, 2016. [Slides] [Poster]
- **Yongxin Tong**, Jieying She, Bolin Ding, Libin Wang, Lei Chen. "Online Mobile Micro-Task Allocation in Spatial Crowdsourcing", in *Proceedings of the 32nd International Conference on Data Engineering* (**ICDE 2016**), Helsinki, Finland, May 16-20, 2016. [Slides] [Poster]
- Jieying She, **Yongxin Tong**, Lei Chen, Caleb Chen Cao. "Conflict-Aware Event-Participant Arrangement and its Variant for Online Setting", *IEEE Transactions on Knowledge and Data Engineering* (**TKDE**), 28(9): 2281-2295, September 2016. [Slides]
- Di Jiang, **Yongxin Tong**, Yuanfeng Song. "Cross-Lingual Topic Discovery from Multilingual Search Engine Query Log", to appear in *ACM Transactions on Information Systems* (**TOIS**). [Slides]

课时分配

- 第一部分（14学时）
  - 第一章　大数据及其处理技术（2学时）
  - 第二章　大数据自然科学文选（2学时）
  - 第三章　大图搜索挑战与技术（2学时）
  - 第四章　众包与群智计算（2学时）
  - 第五章　大数据抽样算法（2学时）
  - 第六章　海量流数据处理（2学时）
  - 第七章　时空大数据处理（2学时）
- 第二部分（18学时）
  - 第八章　学术研究与方法初探（2学时）
  - 第九章　大数据前沿研究分析（16学时）

马帅

童咏昕

马帅

同学们

课时分配

- 第一部分（14学时）
  - 第一章　大数据及其处理技术（2学时）
  - 第二章　大数据自然科学文选（2学时）
  - 第三章　大图搜索挑战与技术（2学时）
  - **第四章　众包与群智计算（2学时）**
  - 第五章　大数据抽样算法（2学时）
  - 第六章　海量流数据处理（2学时）
  - 第七章　时空大数据处理（2学时）
- 第二部分（18学时）
  - 第八章　学术研究与方法初探（2学时）
  - 第九章　大数据前沿研究分析（16学时）

马帅

童咏昕

马帅

同学们

Outline

- Crowdsourcing in a Nutshell
- State-of-art
- Crowdsourcing in a Vision

Outline

- Human Computation in a Nutshell
  - A Brief History
  - Representative Applications & Sharing Economy
  - Core Challenges
- State-of-art
- Crowdsourcing in a Vision

Outline

- Human Computation in a Nutshell
  - **A Brief History**
  - Representative Applications & Sharing Economy
  - Core Challenges
- State-of-art
- Crowdsourcing in a Vision

Human Computation

- "Hey, ever thought of this question: what is Computer?"

- "The machines developed based on semi-conductor…"

- "Actually, during its first usage the word means: one who computes"

Human Computation

NACA High Speed Flight Station "Computer Room"(wiki)

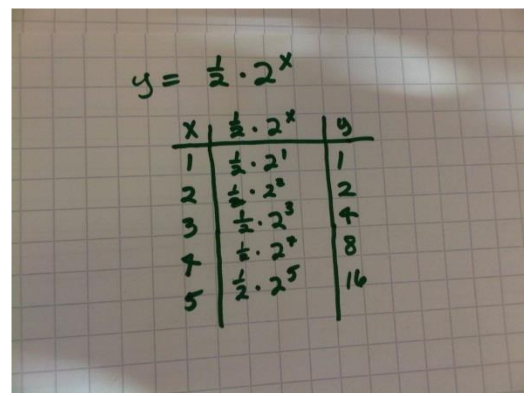
- **"If you can compute, you can serve as a (human) computer"**
- **"During wartime, women are *organized* to process raw flights and map data."**

Human Computation

Math Table Poject(D. A. Grier)

ENIAC, 1946

- **"In 1930's, 450 out-of-work clerks were *organized* to calculate tabular higher mathematical functions."**



- **"Really? That was even before the birth of ENIAC!"**

Human Computation

- **“OK, it was cool. But it is all about history, why it becomes interesting now?”**

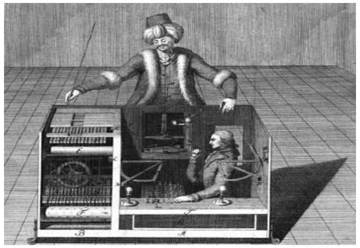
- **“Because we are now in the Web-age.”**

Human Computation

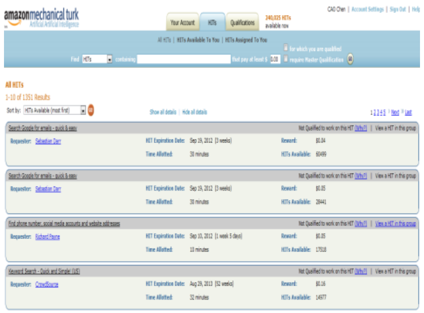
- “What’ s new in web-age?”
- “Did you notice the word I use to describe human computation?”
- “Oh! *Organized!*”

Crowdsourcing

Requesters



AMT



MTurk workers
(Photo By Andrian Chen)

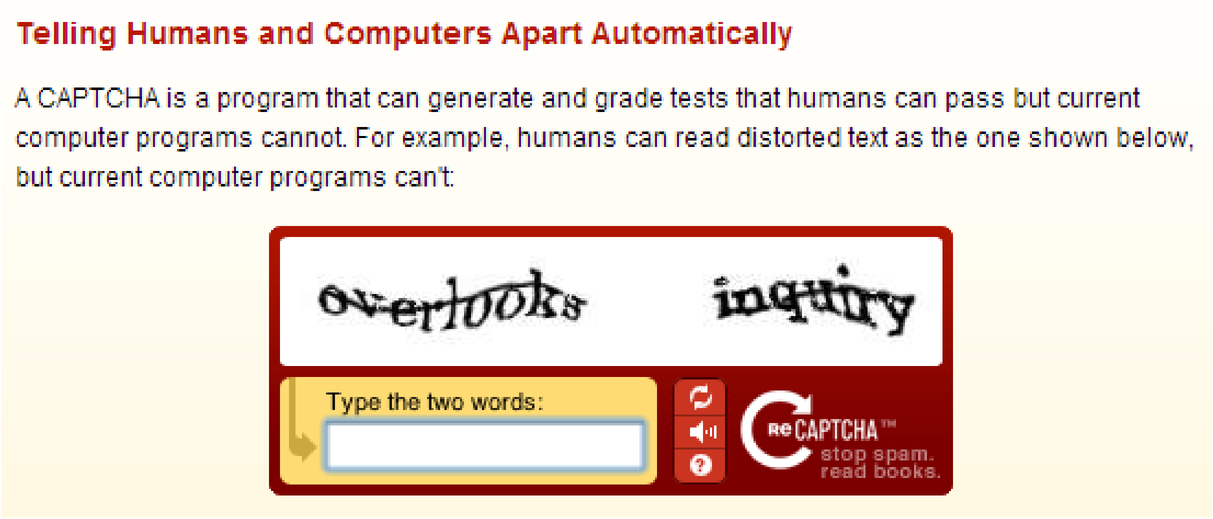
- **“The web connects the requesters and workers.”**

Outline

- Human Computation in a Nutshell
  - A Brief History
  - **Representative Applications & Sharing Economy**
  - Core Challenges
- State-of-art
- Crowdsourcing in a Vision

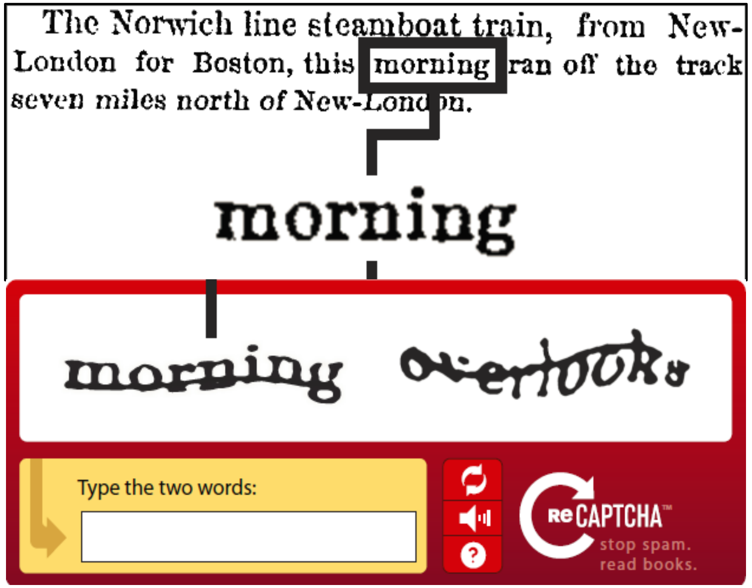
Examples of Crowdsourcing

- reCAPTCHA(Luis von Ahn)
  - Object
    - differentiating Human from Robot
    - digitizing old books one word at a time
  - Human task: input the blurry words



Examples of Crowdsourcing

- reCAPTCHA(Luis von Ahn)
  - Object
    - differentiating Human from Robot
    - digitizing old books one word at a time
  - Human task: input the blurry words



Examples of Crowdsourcing

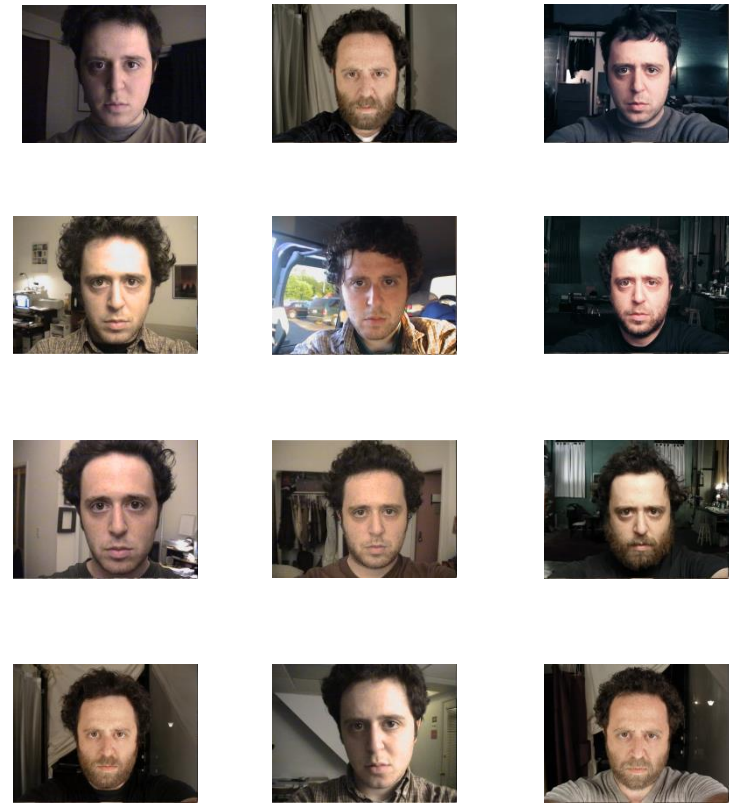
- ESP Game(Luis von Ahn)
  - Object: Images Labeling
  - Human task: online game, two players guessing one common item

PLAYER 1	PLAYER 2
	
GUESSING: CAR	GUESSING: BOY
GUESSING: HAT	GUESSING: CAR
GUESSING: KID	SUCCESS! YOU AGREE ON CAR
SUCCESS! YOU AGREE ON CAR	

Examples of Crowdsourcing

- Human-intrinsic Tasks

- **“Tony, I heard you are a very talented CS expert”**
- **“Yes…?”**
- **“You must be good at sorting “**
- **“I suppose so…?”**
- **“Great, help me to order my photos by age”**
- **“Ok… (lucky there are only 12)”**



Noah Kalina’s Daily Collection

Examples of Crowdsourcing

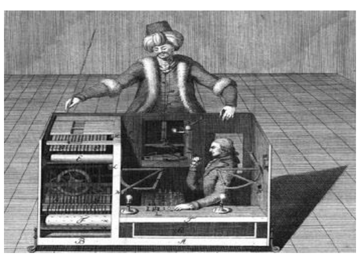
- **Question/Answer Type of Crowdsourcing**

YAHOO! ANSWERS

stackoverflow

Quora

- **General Crowdsourcing**



AMT

oDesk

CrowdFlower

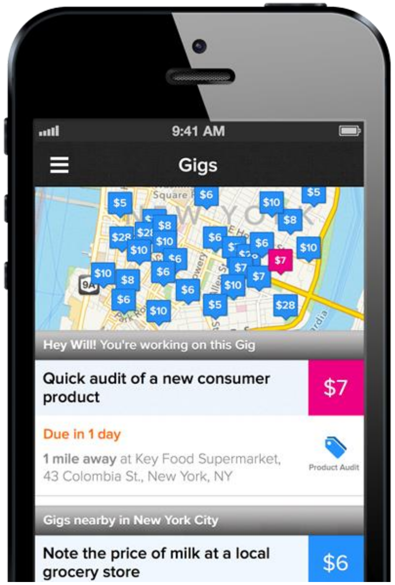
Examples of Spatio-temporal Crowdsourcing

UBER

Gigwalk

FIELD AGENT™

BOSTON taskrabbit
Life is busy. We can help.



饿了么

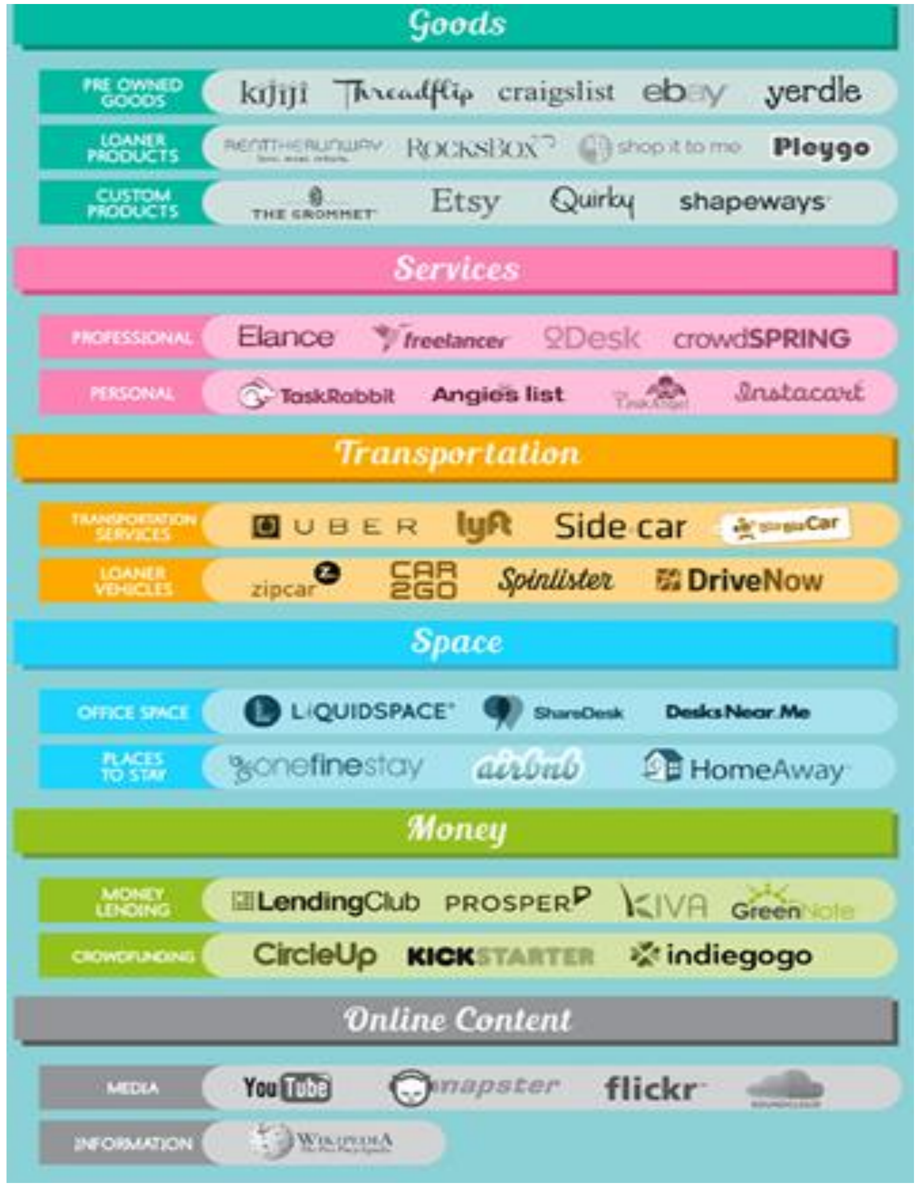
滴滴出行
滴滴一下美好出行
滴滴打车 全新品牌升级

OpenStreetMap

waze

Sharing Economy

Sharing Economy



Outline

- Human Computation in a Nutshell
  - A Brief History
  - Representative Applications & Sharing Economy
  - **Core Challenges**
- State-of-art
- Crowdsourcing in a Vision

Challenges of Crowdsourcing

Any challenges?

Challenges of Crowdsourcing

Challenges of Crowdsourcing

- Task assignment is always the most important issue in crowdsourcing applications
  - Help assign offline tasks to online crowd workers
  - e.g. Offline-to-Online (O2O) applications

Challenges of Crowdsourcing

- **Real-time taxi-calling services usually adopt ‘nearest neighbor (NN)’ strategy to address task assignment issues.**
  - Once a task appears, it should be assigned immediately.

Challenges of Crowdsourcing

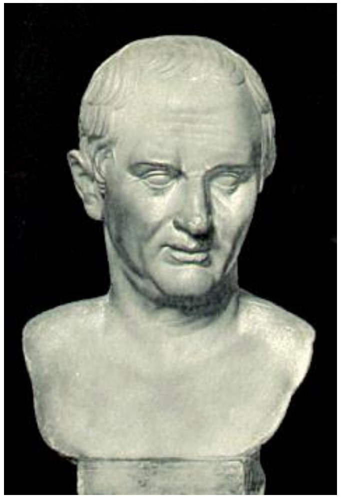
- **Real-time taxi-calling services usually adopt 'nearest neighbor (NN)' strategy to address task assignment issues.**
  - Once a task appears, it should be assigned immediately.
  - If we know everything in advance, the offline OPT is

Challenges of Crowdsourcing

Challenges of Crowdsourcing

- Data Quality is always the **first concern** in crowdsourcing applications

Challenges of Crowdsourcing



- **Erroneous**

  "To err is human"

  —Marcus Tullius Cicero

- **Greedy**

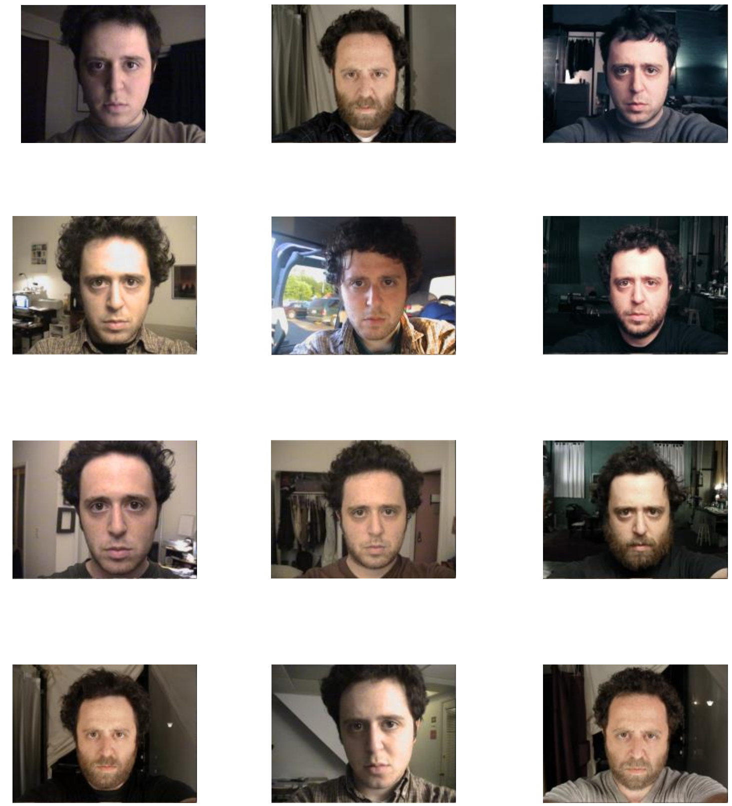
  "…my more-having would be a sauce to make me hunger more"

  —Macbeth act 4, sc. 3, Shakespeare

Challenges of Crowdsourcing

- Human-intrinsic Tasks

- **“Tony, I heard you are a very talented CS expert”**
- **“Yes…?”**
- **“You must be good at sorting “**
- **“I suppose so…?”**
- **“Great, help me to order my photos by age”**
- **“Ok… (lucky there are only 12)”**



Noah Kalina’s Daily Collection

Challenges of Crowdsourcing

- During the Big Data Era

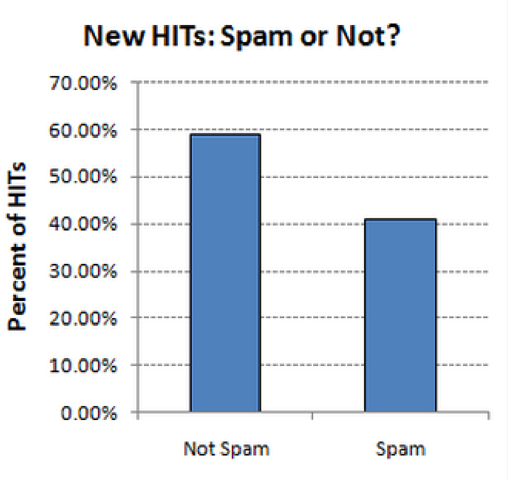
(do you still remember Tony)

- “Tony, did you finish it?”
- “Yep”
- “Cool! I know I can trust the entire collection on you!”
- “Ohhh…, we really need a CS expert, CrowdSourcing, not just Computer Science”

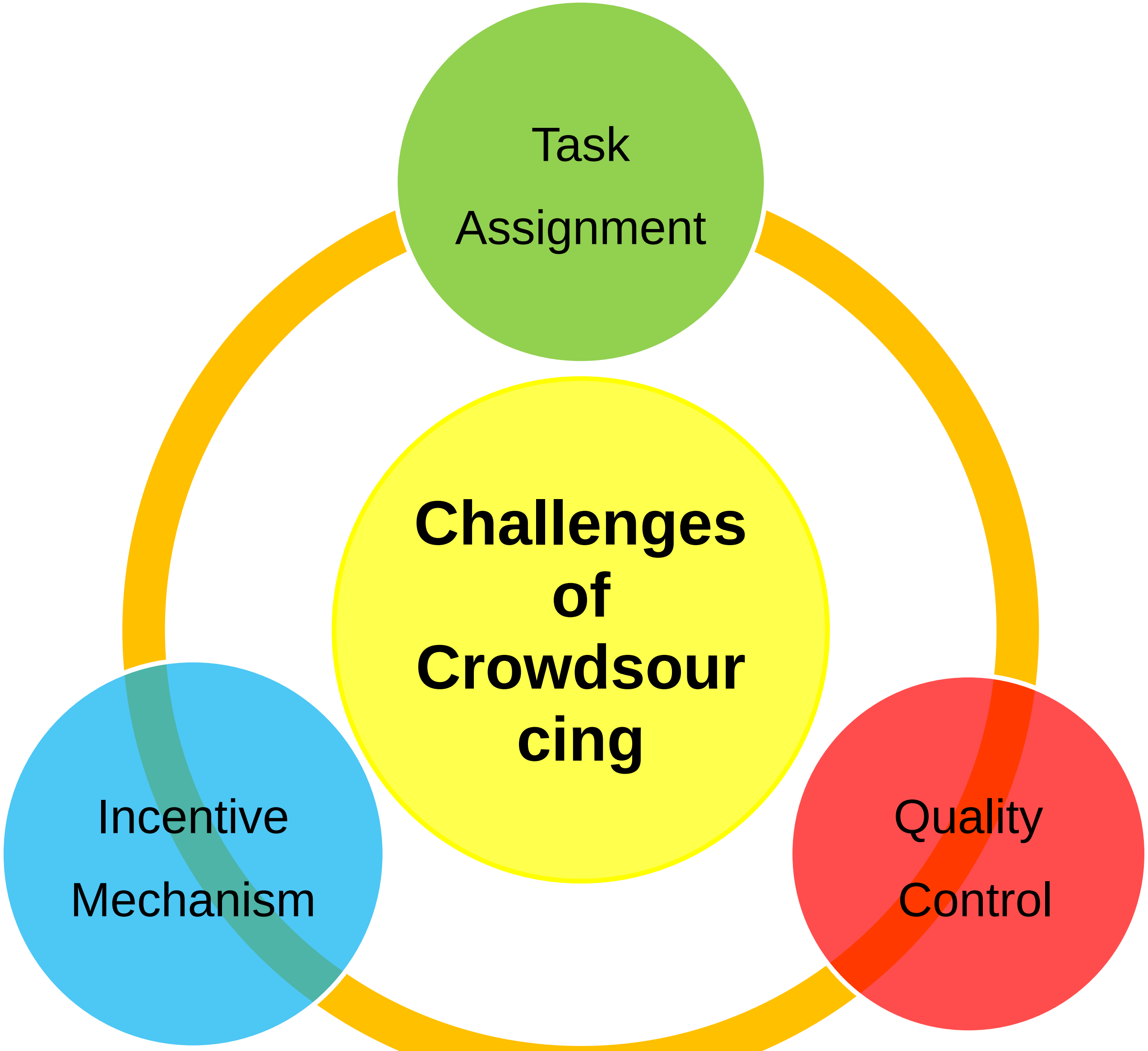
Challenges of Crowdsourcing

- Spammers
  - Spam Worker
    - Finish tasks simply for rewards
    - Low quality of the answers

  - Spam HITs
    - Some tasks are to spamming "social media" metrics
    - Some tricks the worker by refuse to pay for their answers



Challenges of Crowdsourcing

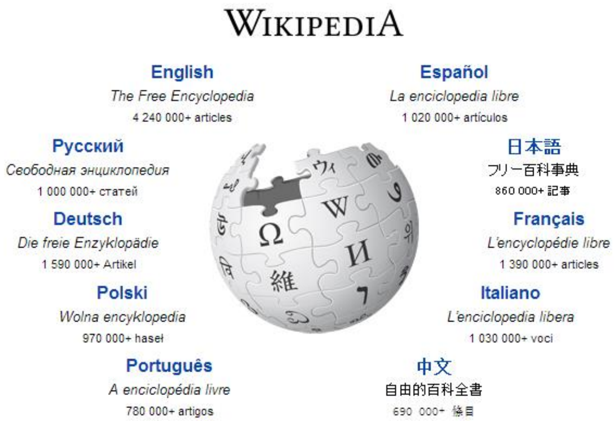


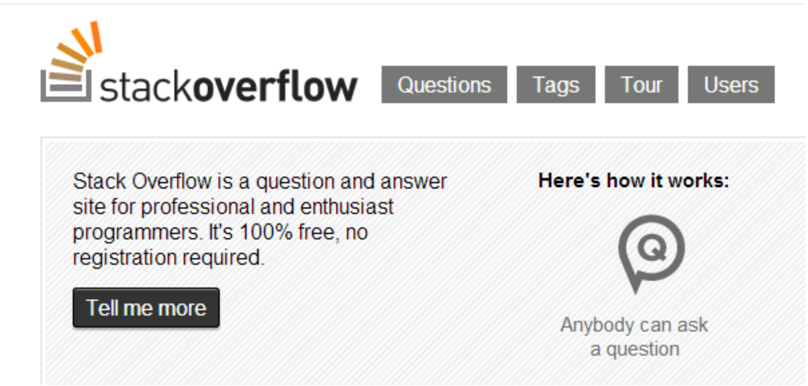
Challenges of Crowdsourcing

How to motivate workers?

Challenges of Crowdsourcing

- Altruistic?

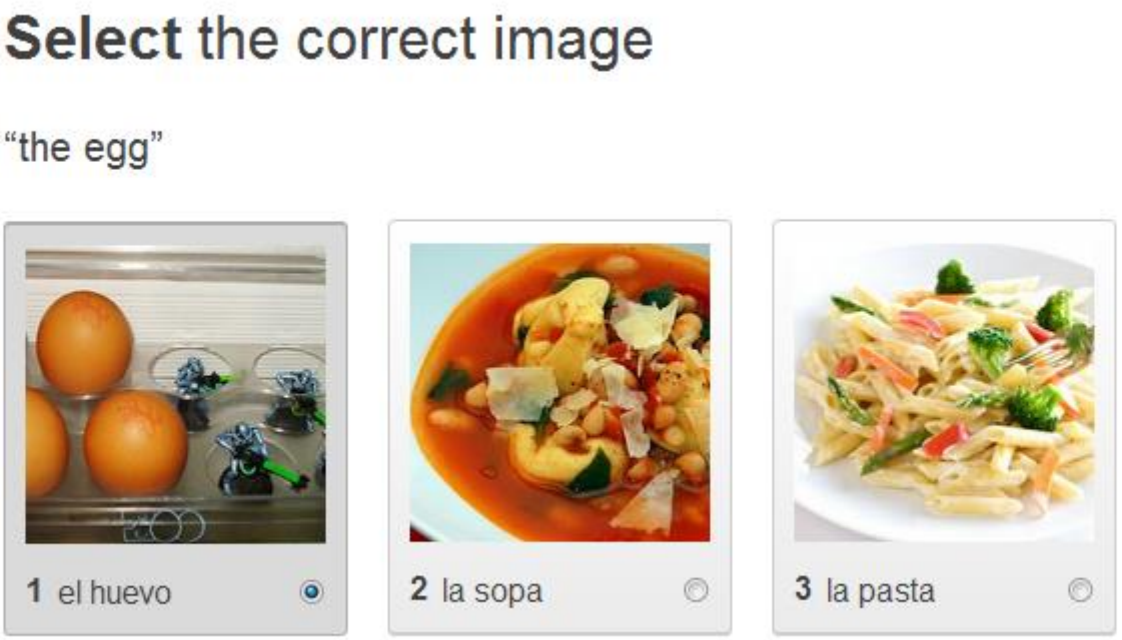




- “Hey I helped solve a .Net problem on stackoverflow!”
- Why are you so happy? Got paid?
- No… It feels good.

Challenges of Crowdsourcing

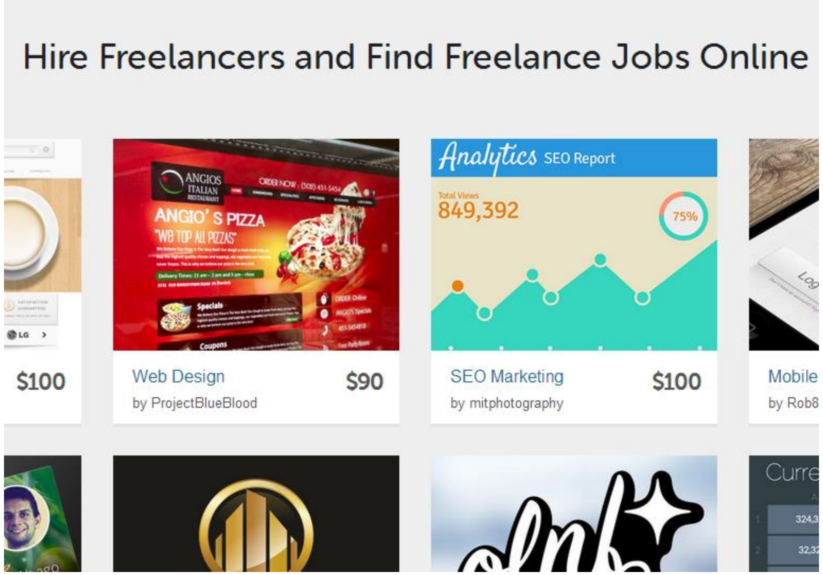
- For fun?



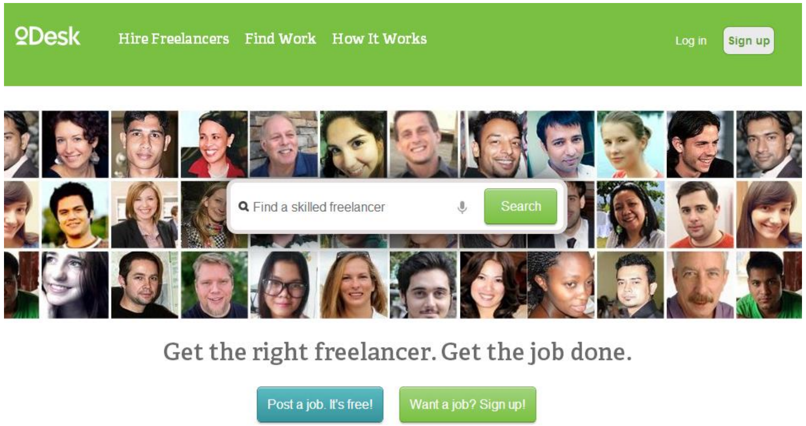
Learning Spanish via Duolingo.com

Challenges of Crowdsourcing

- Money-expecting?



From freelancer.com



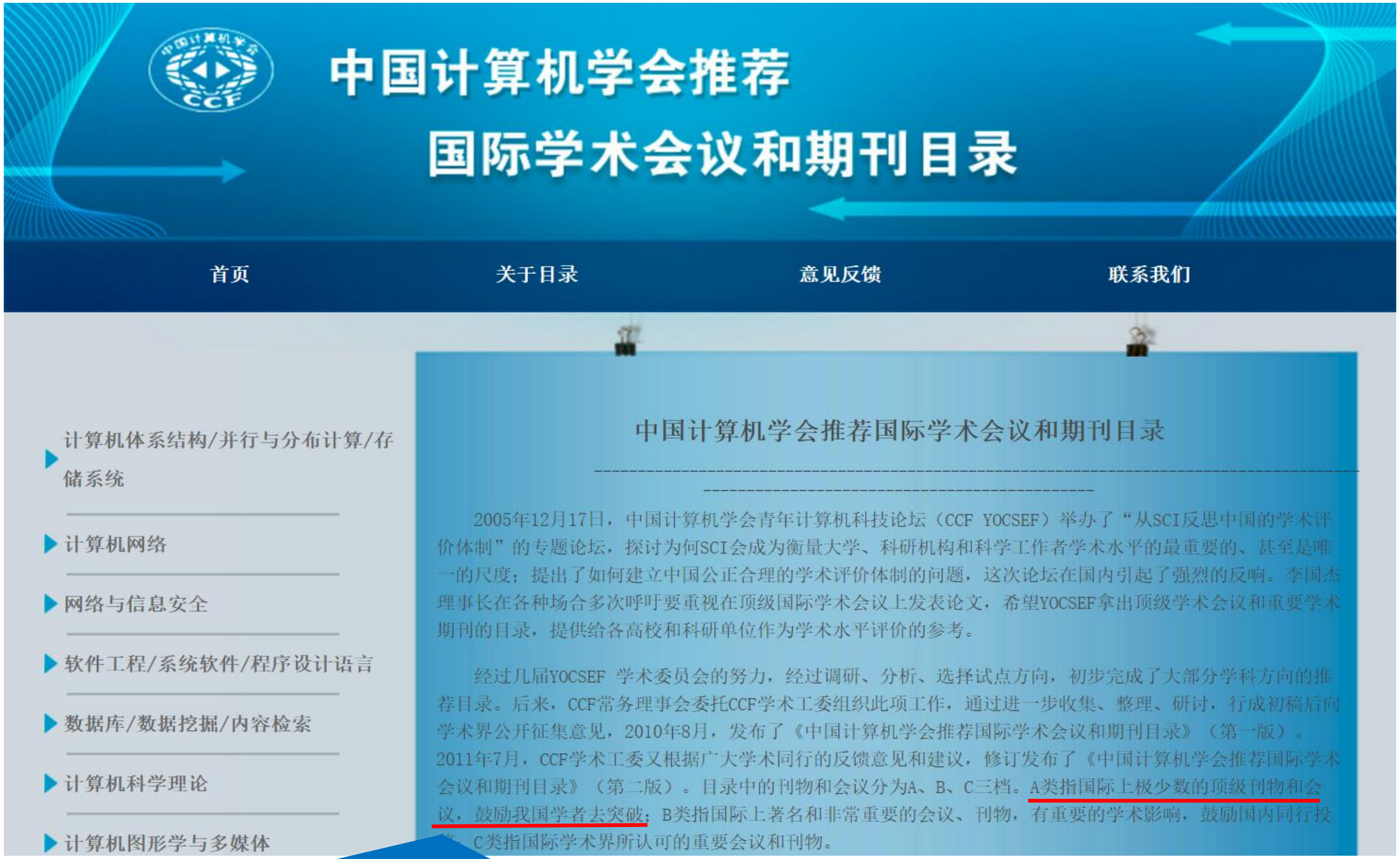
From odesk.com

Outline

- Human Computation in a Nutshell
- State-of-art
  - **A Review of Crowdsourced Databases**
  - Task Assignment
    - Online Mobile Micro-Task Assignment [ICDE’2016]
    - Online Matching in Real-Time Spatial Data [VLDB’2016]
  - Quality Control
    - Whom to Ask [VLDB’2012]
- Crowdsourcing in a Vision

Ranking of CS Conferences/Journals

网址：http://www.ccf.org.cn/sites/ccf/paiming.jsp



A类（CCF A）指国际上极少数顶级刊物和会议，代表计算机学科国际最前沿的发展动态与趋势，鼓励我国学者图突破！

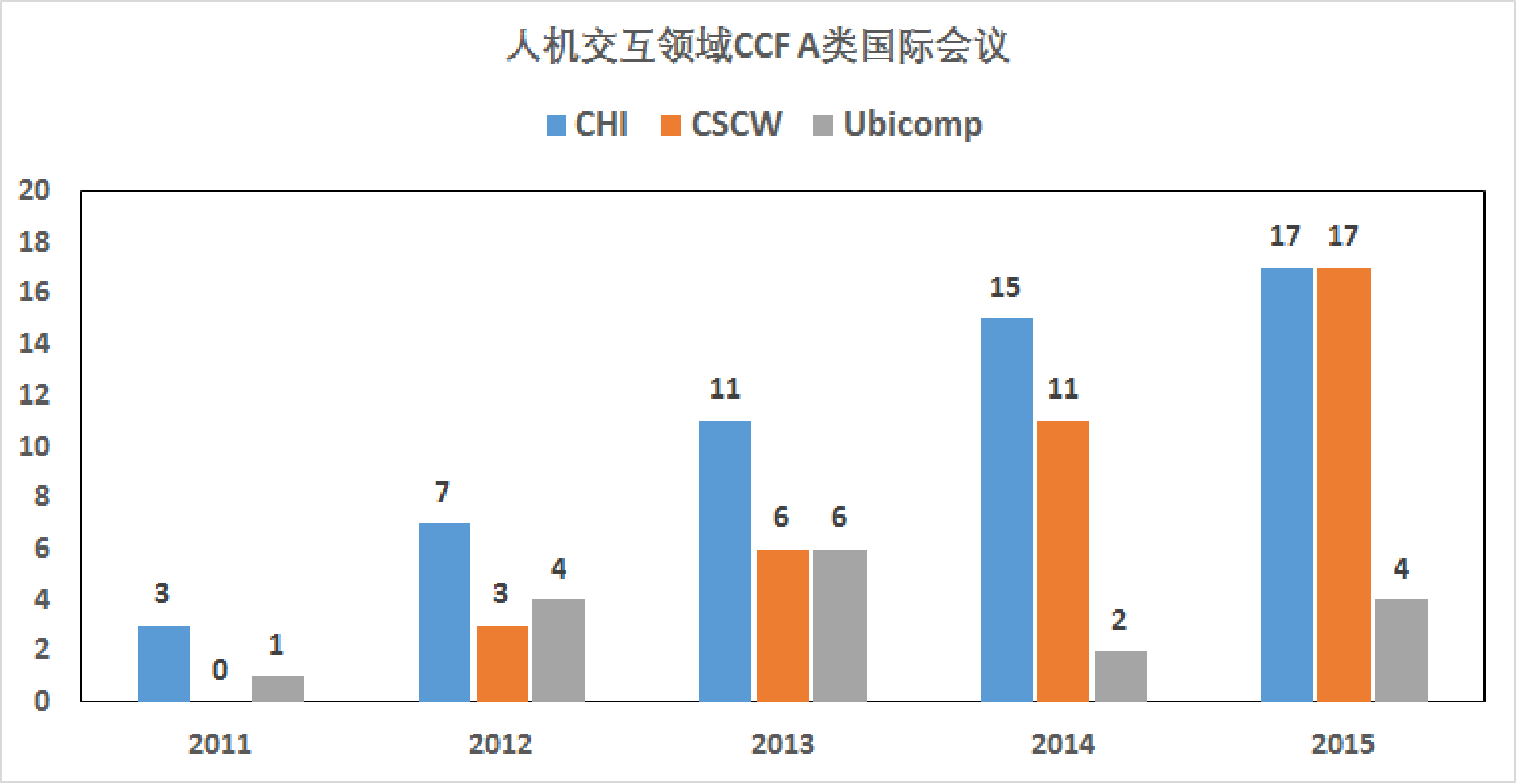
Ranking of CS Conferences/Journals

- 十类研究方向

序号	研究方向
1	计算机体系结构 / 高性能计算 / 存储系统
2	计算机网络
3	网络与信息安全
4	软件工程/系统软件/程序设计语言
5	数据库/数据挖掘/内容检索
6	计算机科学理论
7	计算机图形学与多媒体
8	人工智能
9	人机交互与普适计算
10	交叉/新兴 / 综合等

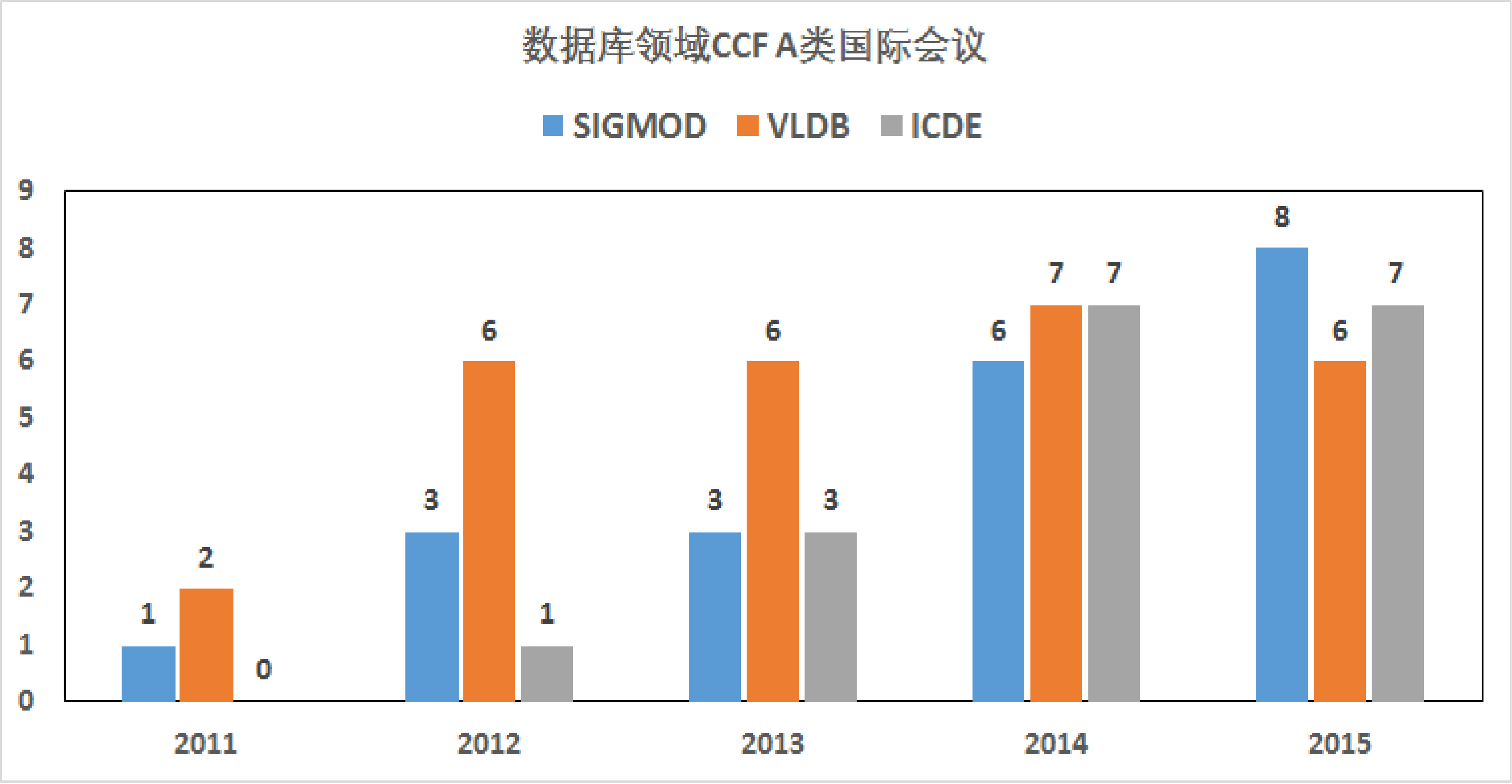
Research Trends in Recent 5 Years

- 3 Top Conferences of CHI
  - CHI, Ubicomp, CSCW



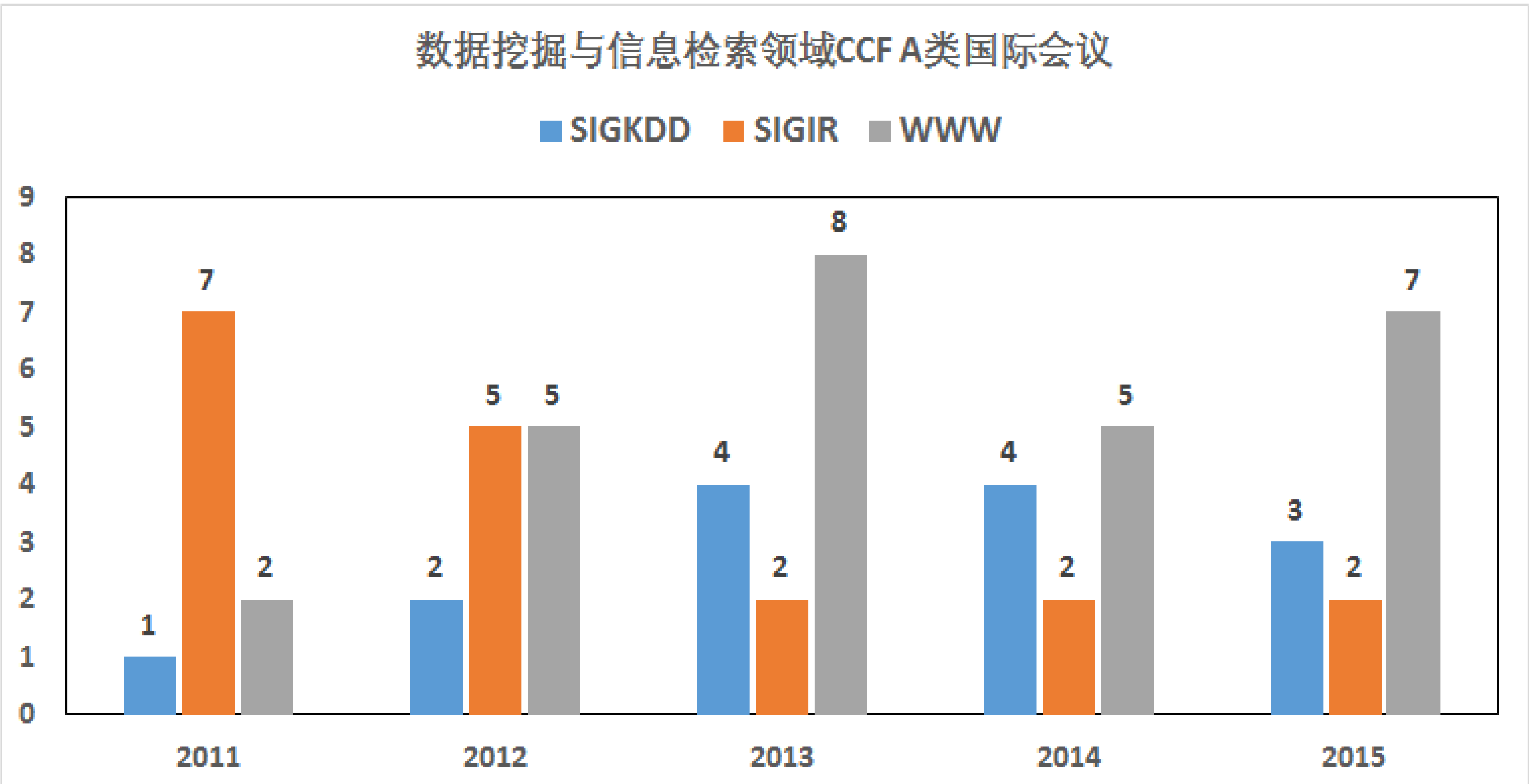
Research Trends in Recent 5 Years

- 3 Top Conferences of DB
  - SIGMOD, VLDB, ICDE



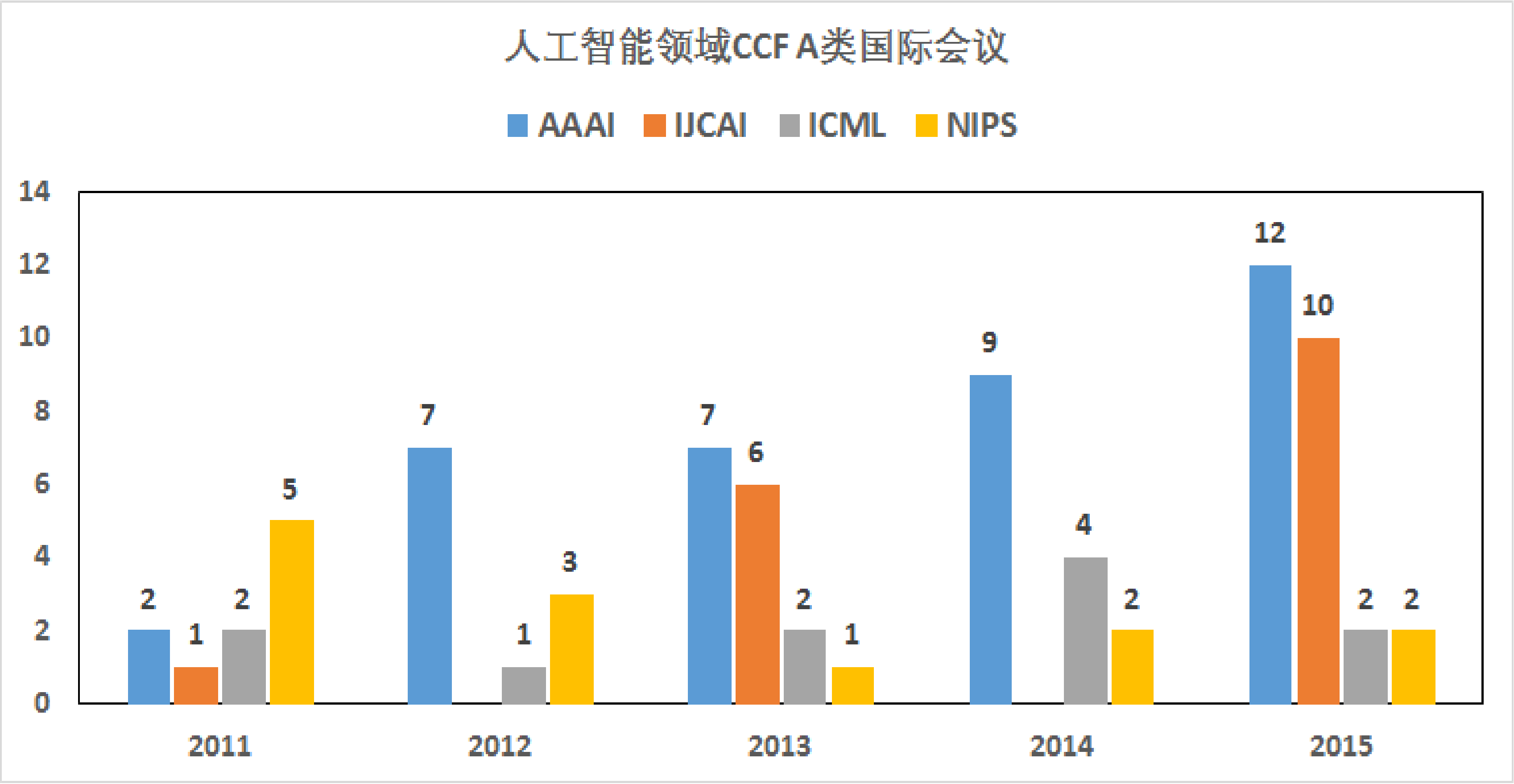
Research Trends in Recent 5 Years

- 3 Top Conferences of DM & IR
  - SIGKDD, SIGIR, WWW



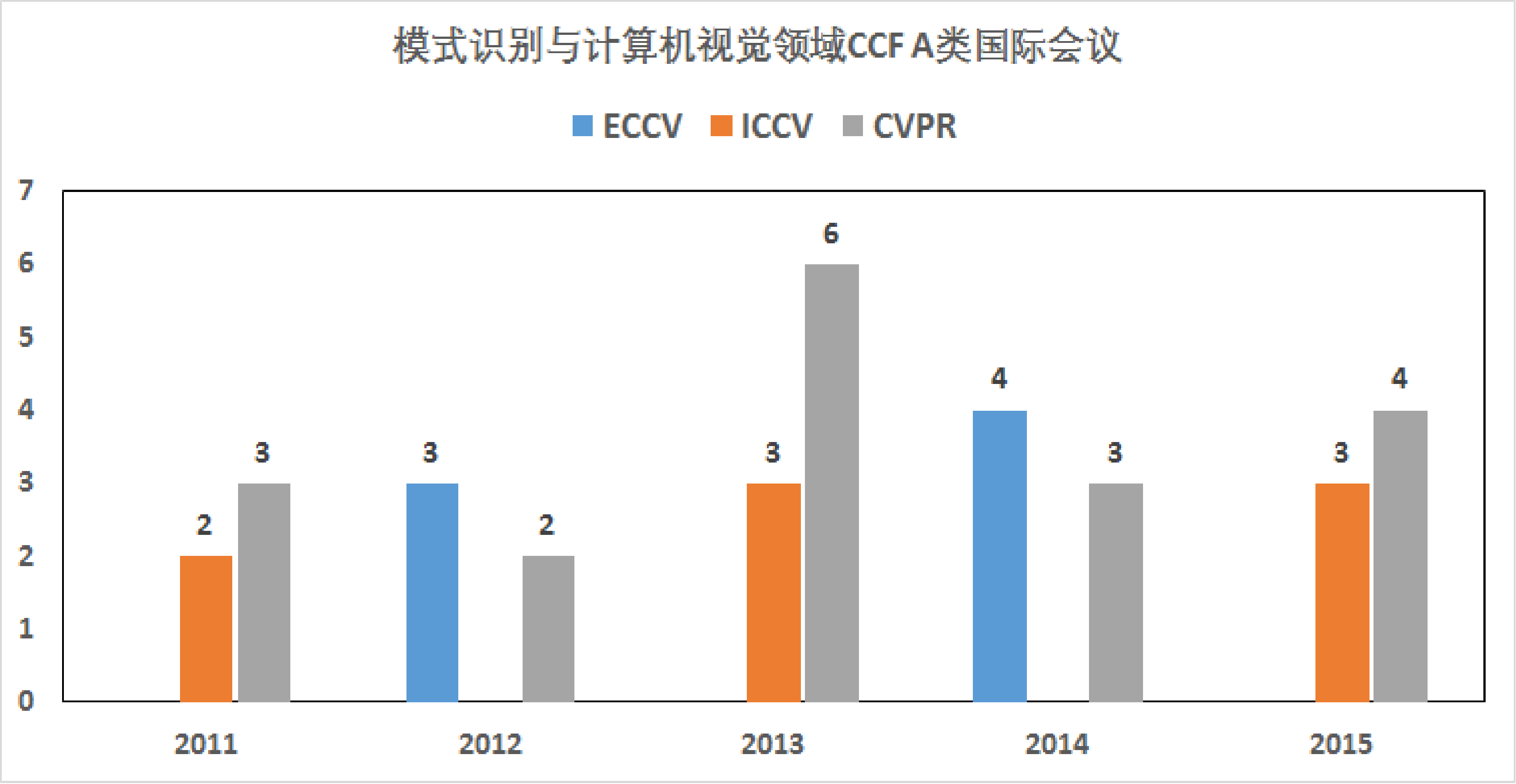
Research Trends in Recent 5 Years

- 4 Top Conferences of AI
  - AAAI, IJCAI, ICML, NIPS



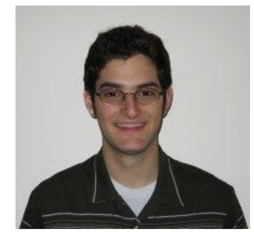
Research Trends in Recent 5 Years

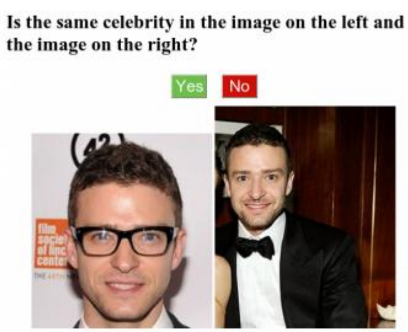
- 3 Top Conferences of PR & CV
  - CVPR, ICCV, ECCV



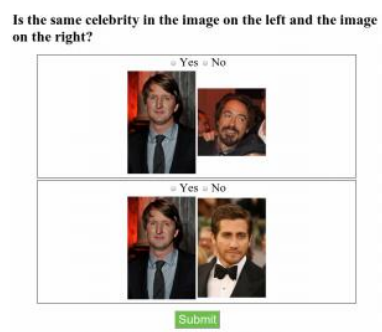
Crowd-powered DB Systems

- Qurk, "*Human powered Sorts and Joins*", MIT
  - Marcus et al. [A. Marcus, CIDR'11], [A. Marcus, SIGMOD'11], [A. Marcus, VLDB'12]
  - Join
  - Batching

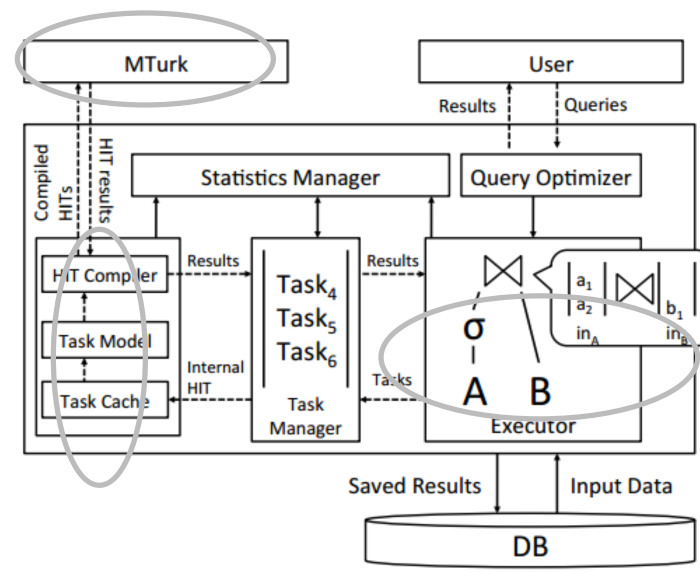




(a) Simple Join



(b) Naive Batching



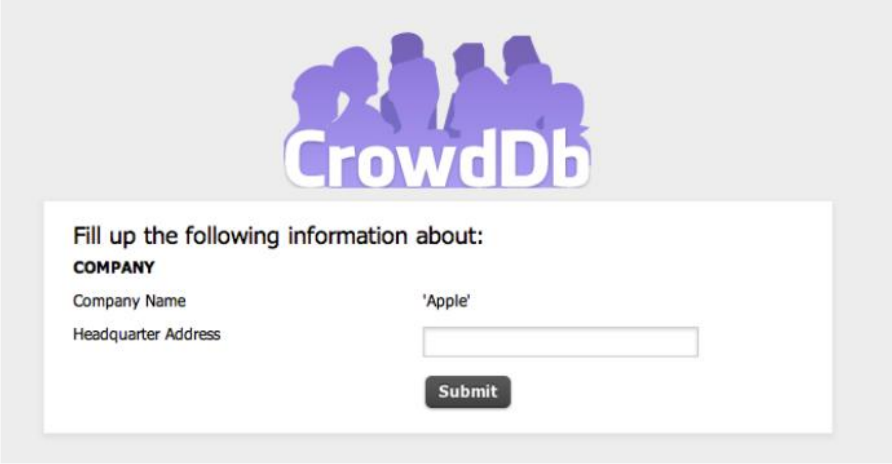
Crowd-powered DB Systems

- Deco/Scoop, *"A System for Declarative Crowdsourcing"*, Stanford
  - Aditya et al.
    - Vision Paper, [P. Aditya, CIDR' 11]
    - Demo System, [H. Park, VLDB' 12]
  - Supporting various following-up operators
    - Max
      - Kenemy Rule + ML, [S. Guo, SIGMOD' 12]
      - Hybrid Ranking Rule, [P. Venetis, WWW' 12]
    - Human-assisted Graph Search, [P. Aditya, VLDB' 11]
    - Filtering [P. Aditya, SIGMOD' 12]
    - Entity Matching [K. Bellare, KDD' 12]

Crowd-powered DB Systems

- CrowdDB, "*Answering Queries with Crowdsourcing*", UC Berkeley
  - Franklin et al, [M.J. Franklin, VLDB2011], [A. Feng, SIGMOD2011]
  - CrowdSQL, "CROWD" keyword

CREATE TABLE company(
Name STRING PRIMARY KEY,
hq_address CROWD STRING);

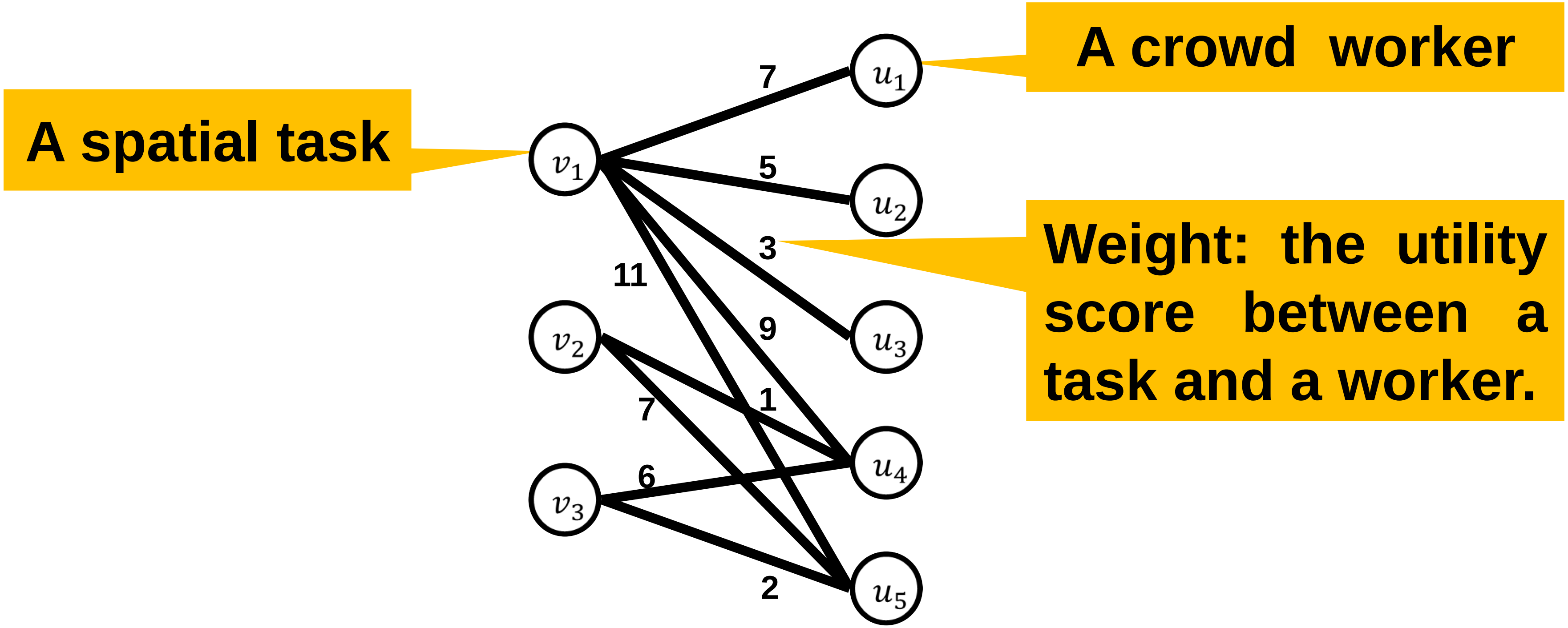


Outline

- Human Computation in a Nutshell
- State-of-art
  - A Review of Crowdsourced Databases
  - **Task Assignment**
    - **Online Mobile Micro-Task Assignment [ICDE'2016]**
    - Online Matching in Real-Time Spatial Data [VLDB'2016]
  - Quality Control
    - Whom to Ask [VLDB'2012]
- Human Computation in a Vision

Existing Task Assignment

- **Existing research: considered as the classical "*maximum weighted bipartite graph matching*" problem.**



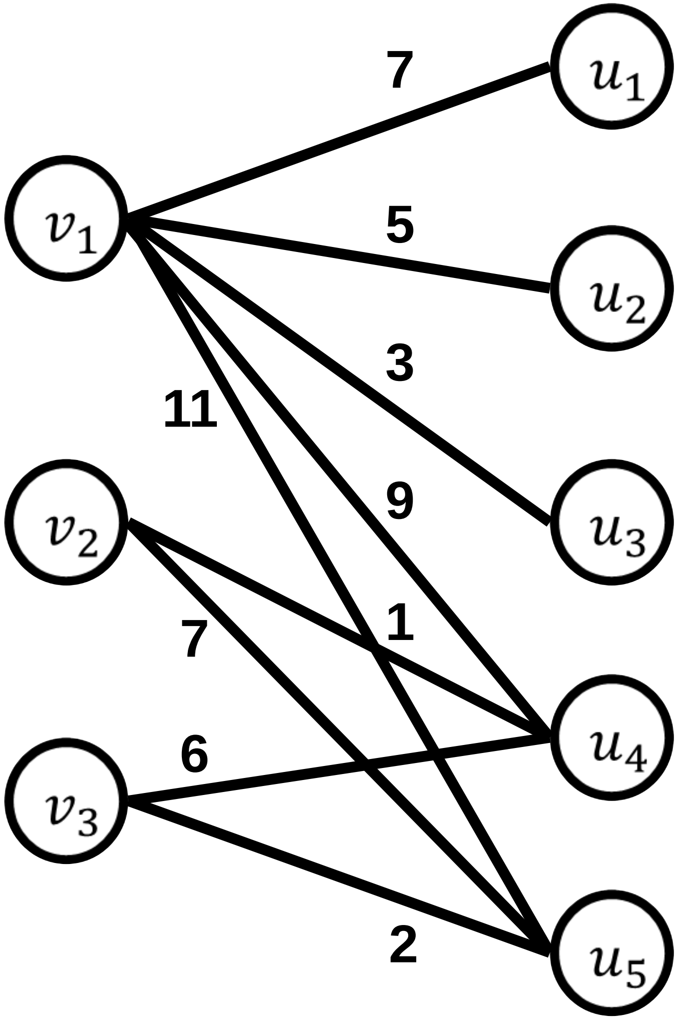
L. Kazemi et al. Geocrowd: enabling query answering with spatial crowdsourcing. In GIS 2012.
H. To et al. A server-assigned spatial crowdsourcing framework. In TASA 2015.

Challenges of Existing Task Assignment

- **Most O2O applications need to be addressed in real-time:**
  - **Fast Food Delivery**
  - **Real-Time Taxi-Calling Service**
  - **Restaurant/Supermarket Queue Monitoring**

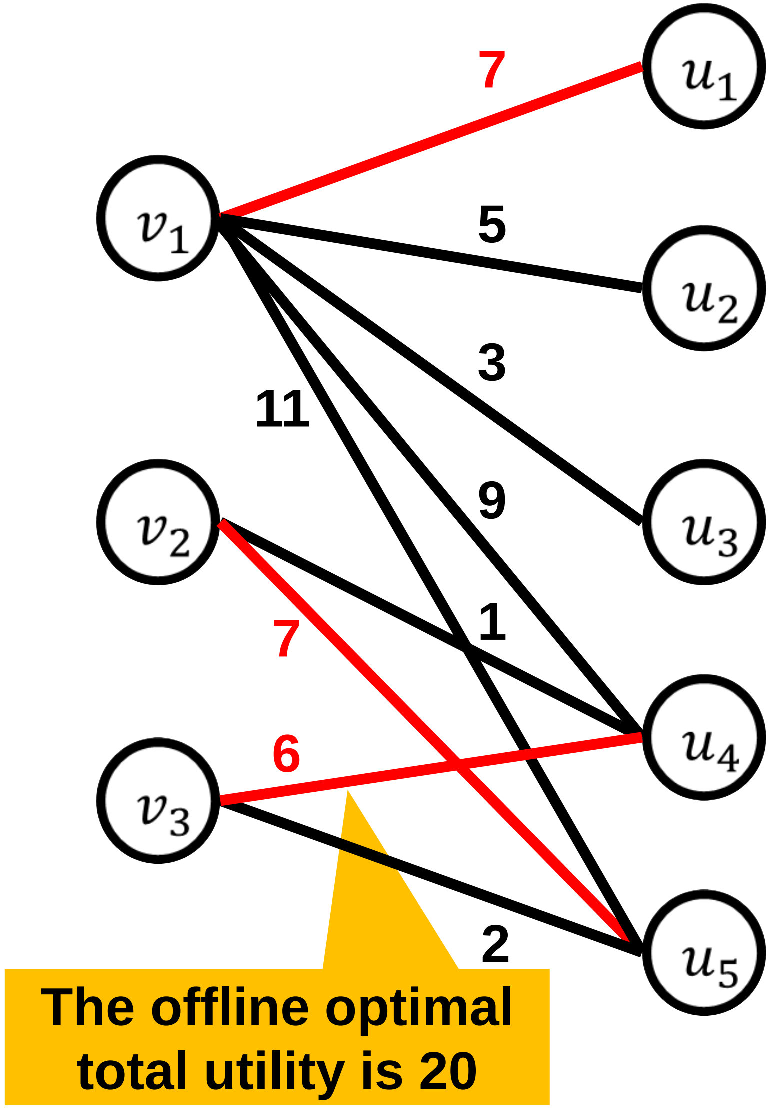
- **Maximum weighted bipartite graph matching is not suitable for dynamic scenarios**
  - **Tasks/workers appear dynamically -- full bipartite graph cannot be known in advance**
  - **Once a task/worker appears, it needs to be immediately assigned based on partial information only**

Online Maximum Weighted Bipartite Matching



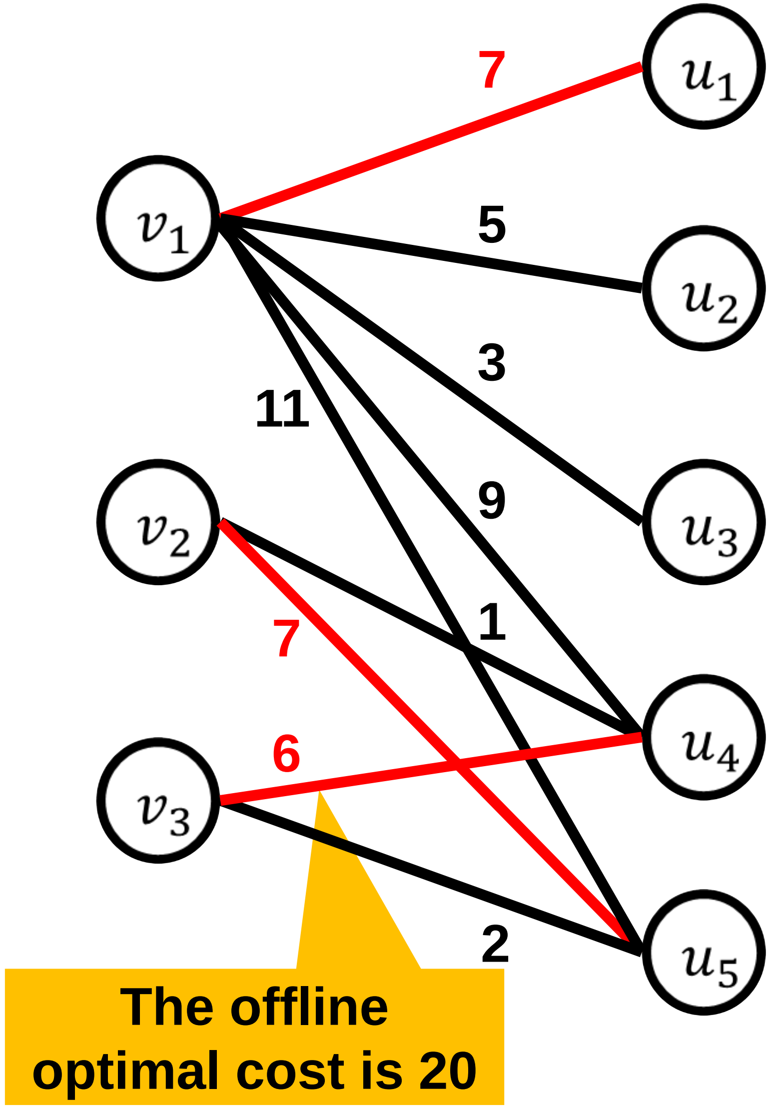
Offline Scenario

Online Maximum Weighted Bipartite Matching



Offline Scenario

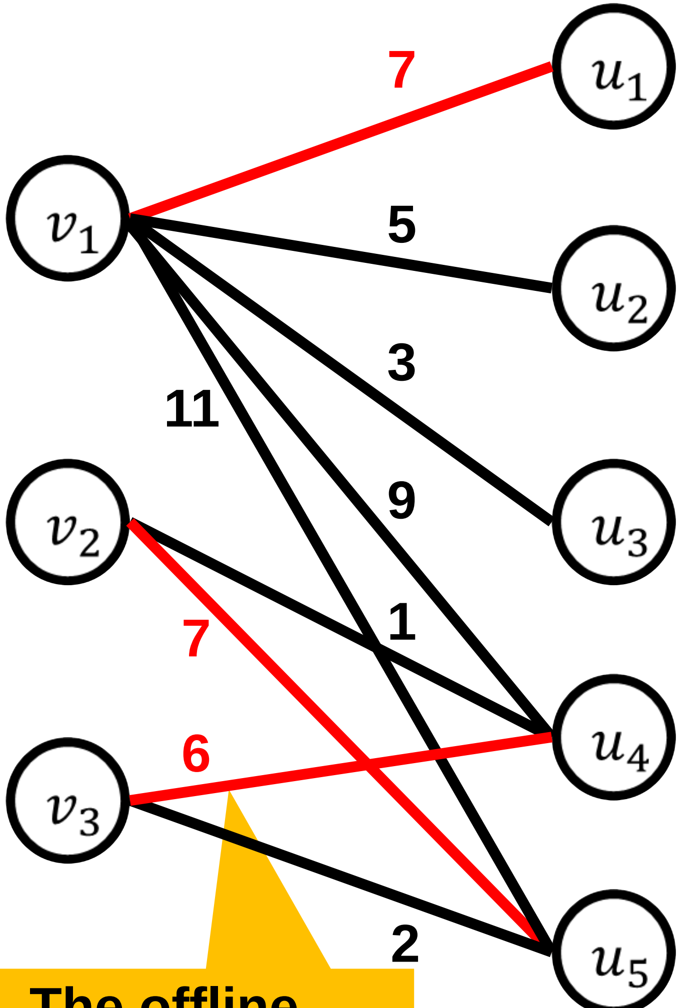
Online Maximum Weighted Bipartite Matching



Offline Scenario

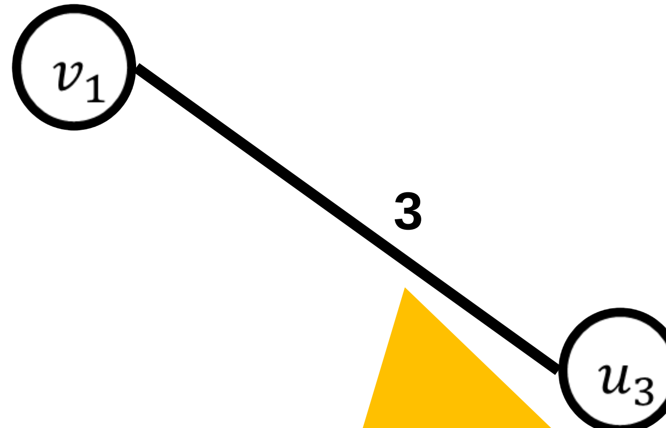
(Two-sided) Online Scenario

Online Maximum Weighted Bipartite Matching



The offline optimal cost is 20

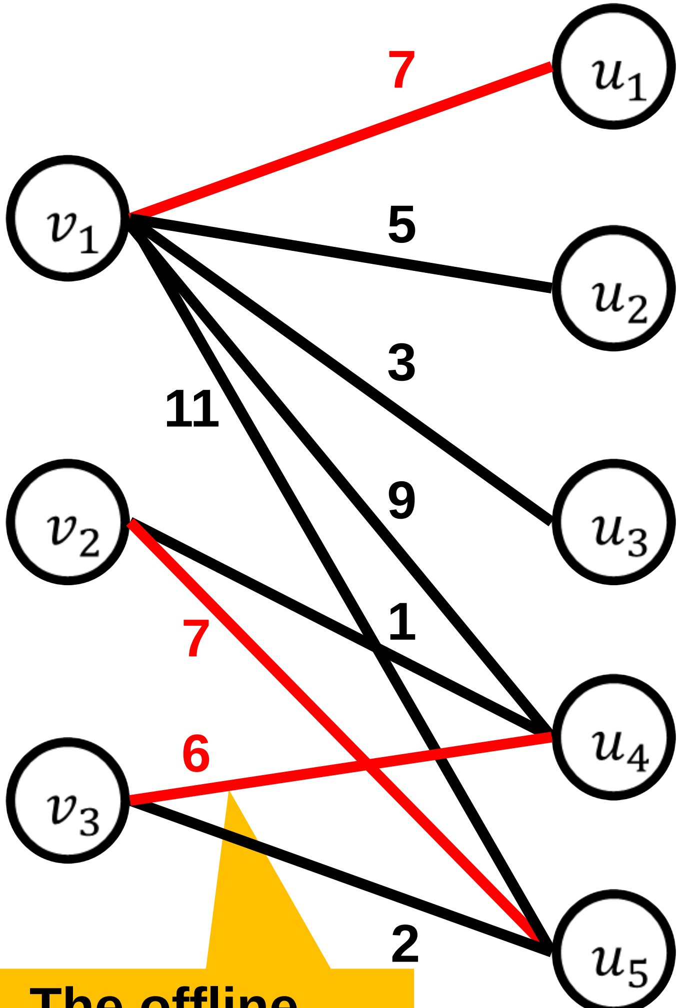
Offline Scenario



1. **Full bipartite graph cannot be known.**
2. **The new arrival object needs to be immediately made a decision based on partial information.**

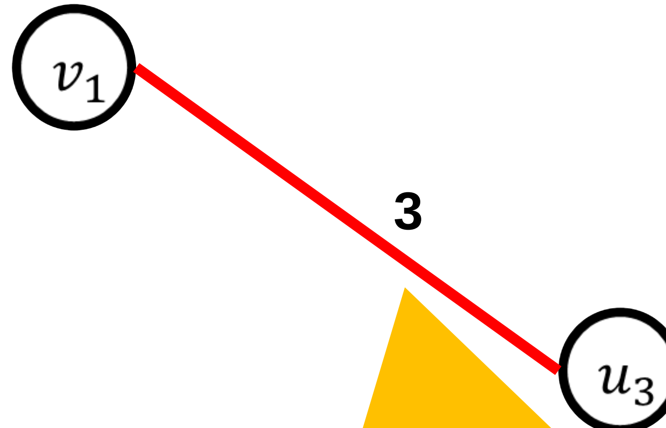
(Two-sided) Online Scenario

Online Maximum Weighted Bipartite Matching



The offline optimal cost is 20

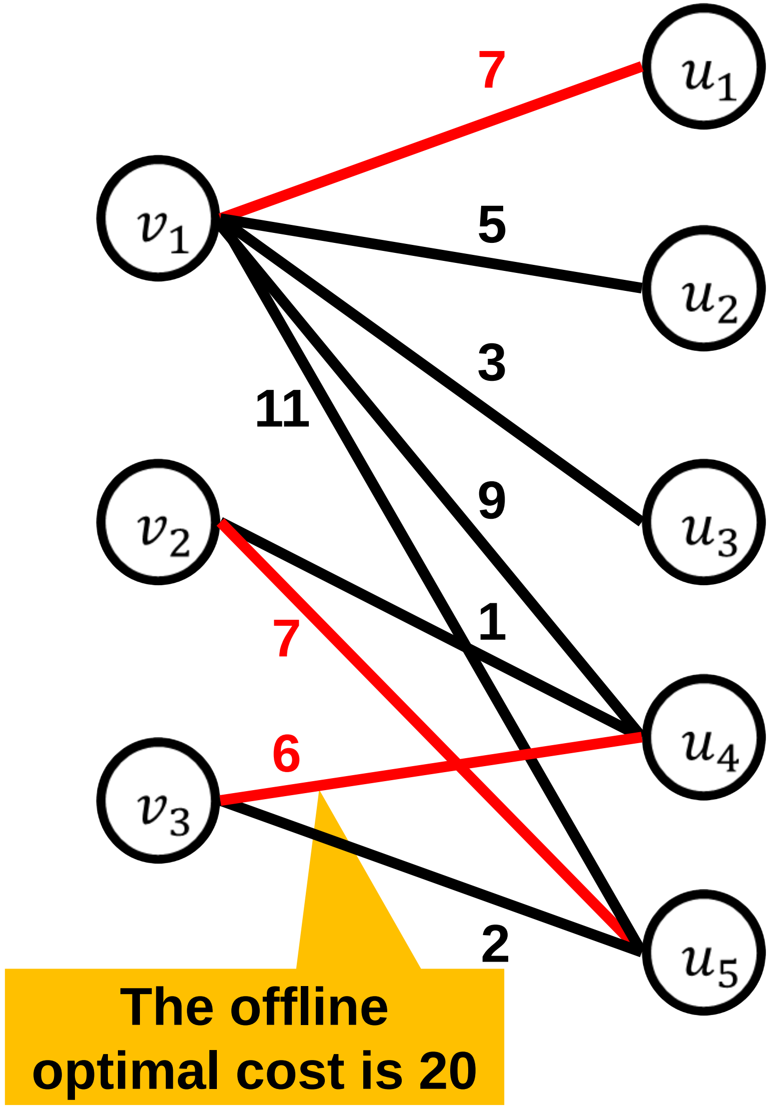
Offline Scenario



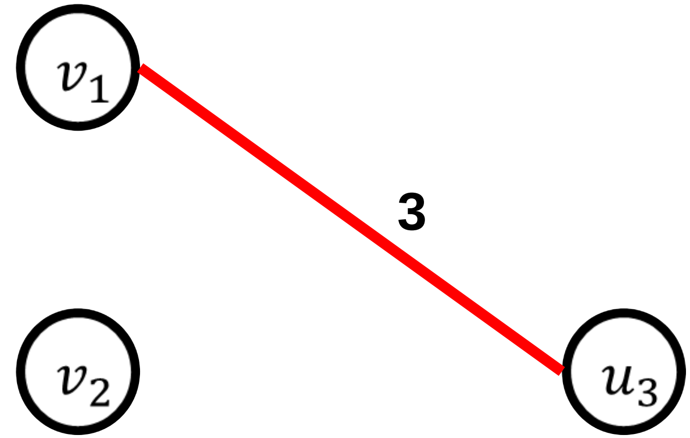
1. **Full bipartite graph cannot be known.**
2. **The new arrival object needs to be immediately made a decision based on partial information.**

(Two-sided) Online Scenario

Online Maximum Weighted Bipartite Matching

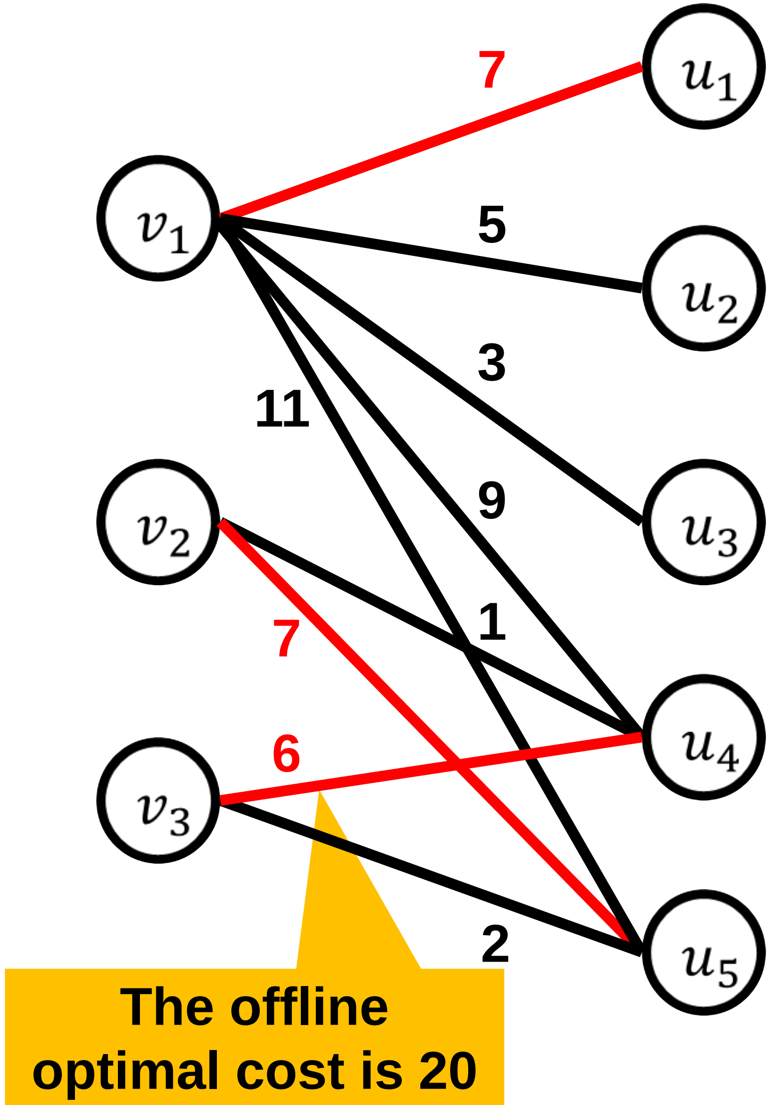


Offline Scenario

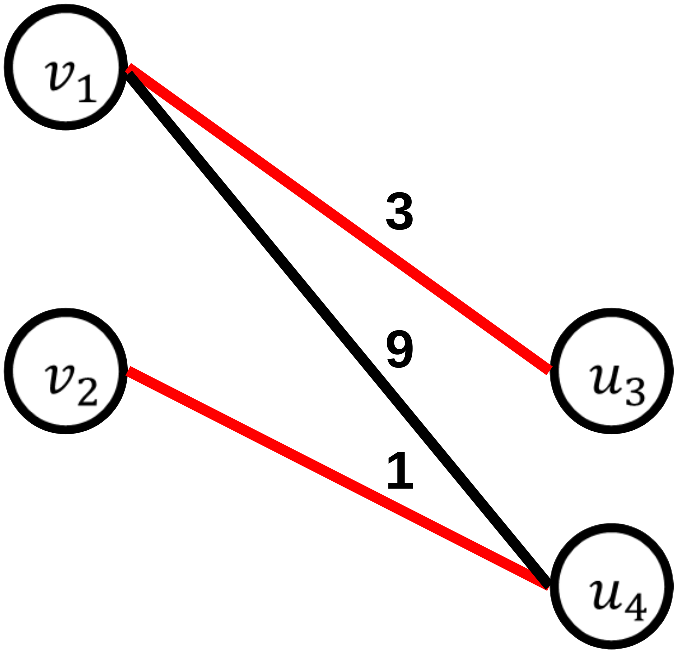


(Two-sided) Online Scenario

Online Maximum Weighted Bipartite Matching

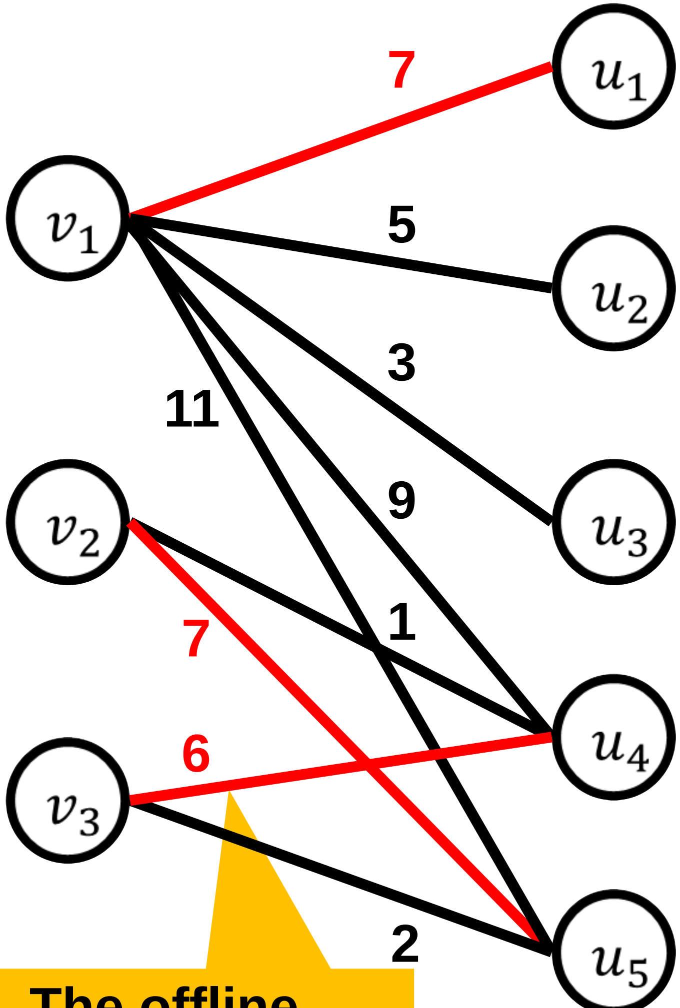


Offline Scenario



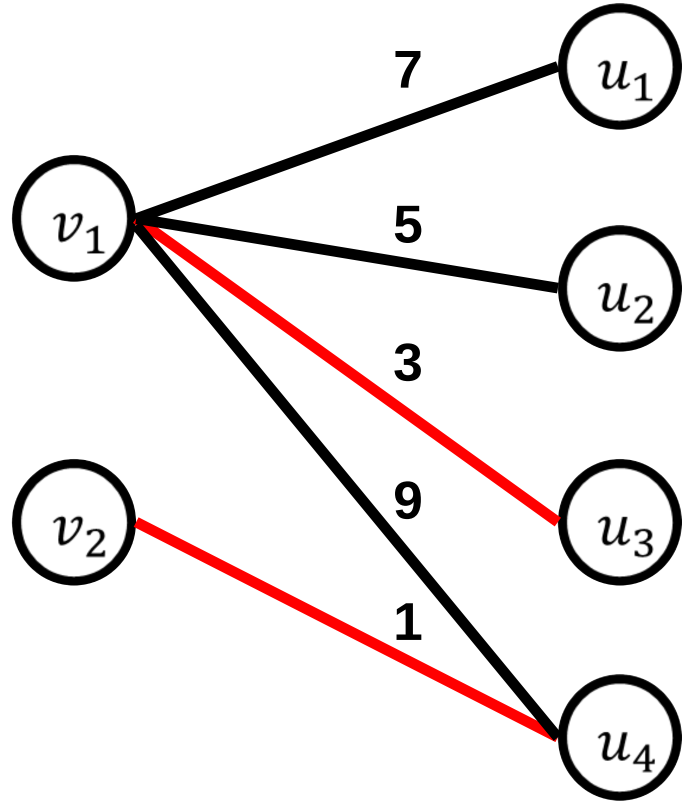
(Two-sided)
Online Scenario

Online Maximum Weighted Bipartite Matching



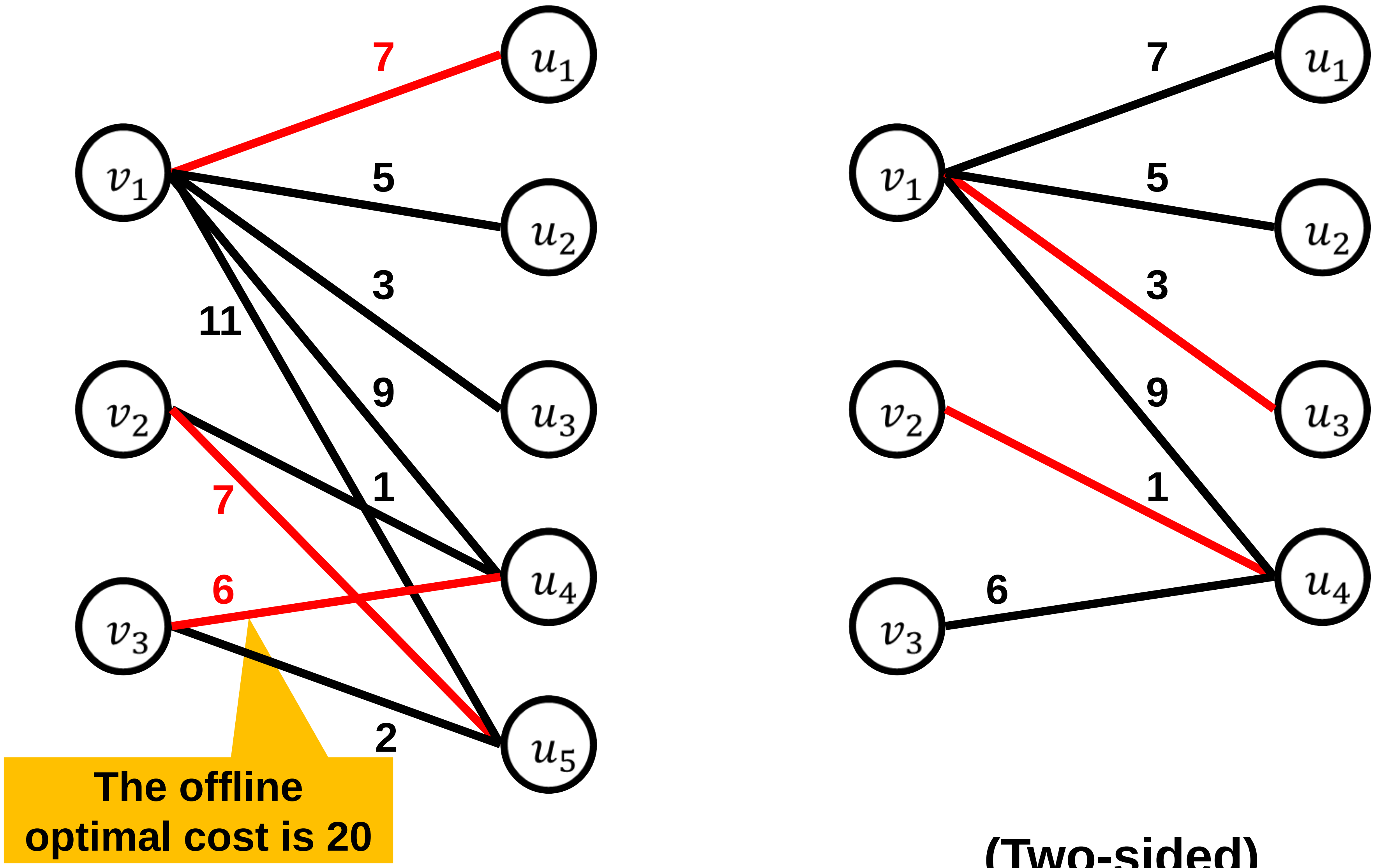
The offline optimal cost is 20

Offline Scenario



(Two-sided)
Online Scenario

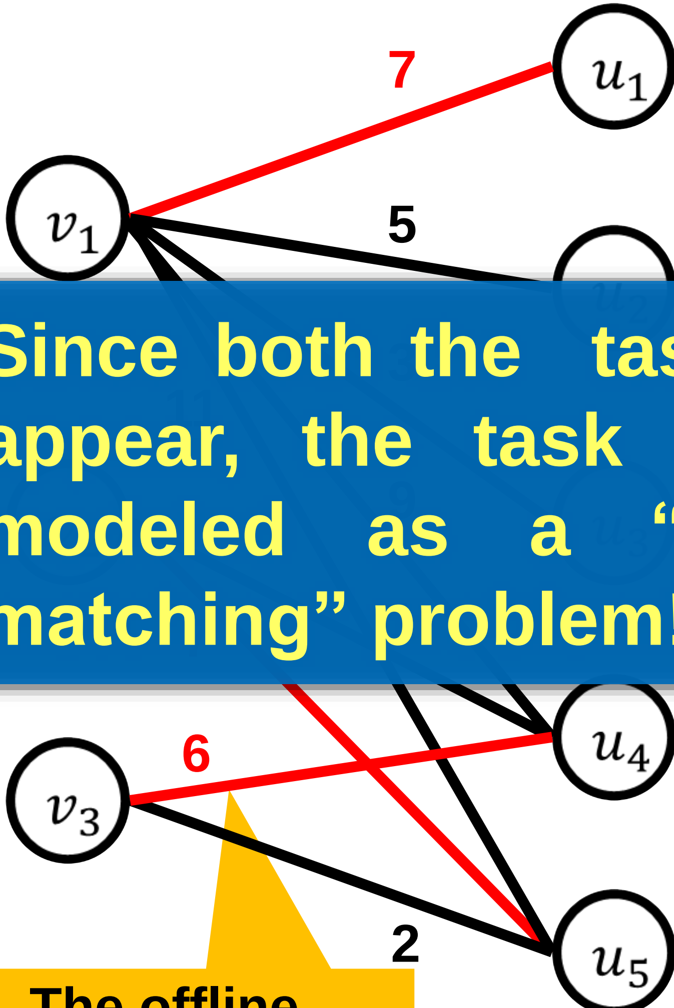
Online Maximum Weighted Bipartite Matching



Offline Scenario

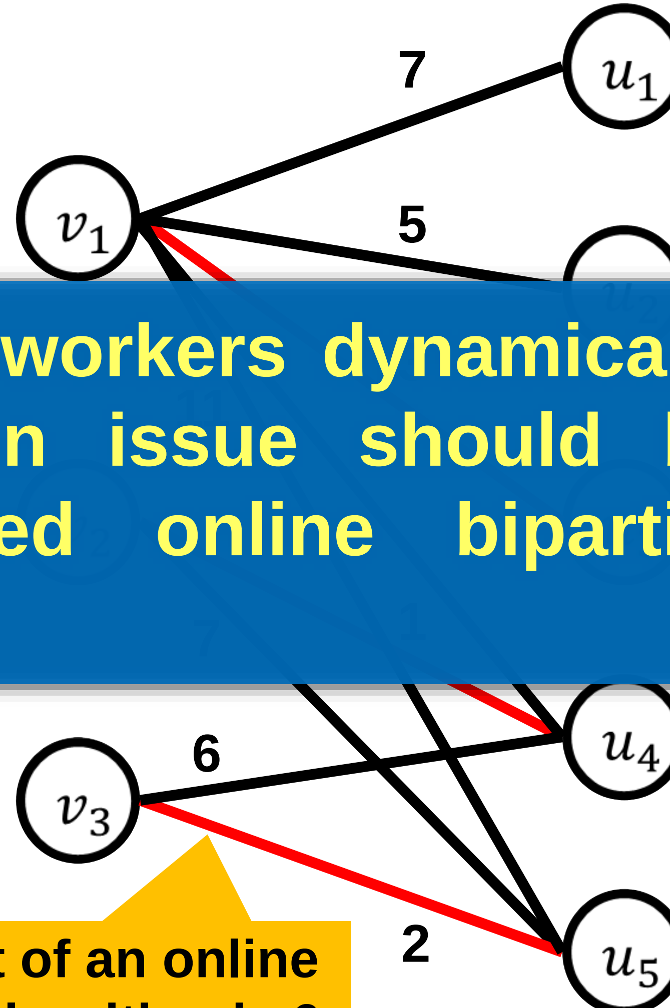
(Two-sided) Online Scenario

Online Maximum Weighted Bipartite Matching



The offline optimal cost is 20

Offline Scenario



The cost of an online greedy algorithm is 6

(Two-sided) Online Scenario

Since both the tasks and workers dynamically appear, the task allocation issue should be modeled as a "Two-Sided online bipartite matching" problem!

Problem Statement

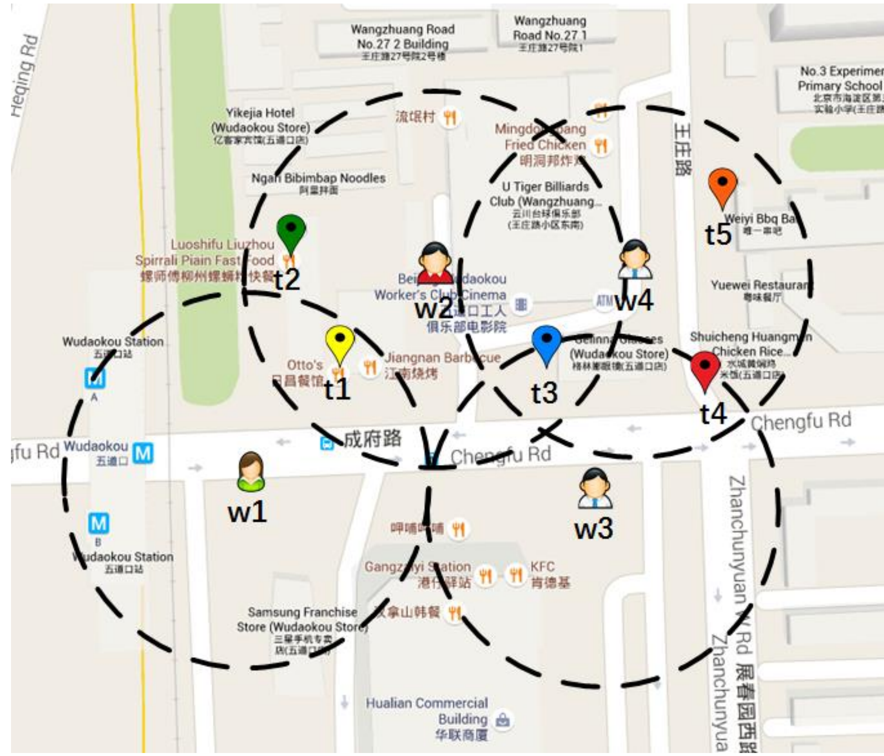
- **Global Online Micro-task Allocation (GOMA)**
- **Given**
  - **A set of spatial tasks** T
    - **Each** $t \in T$**: location** $\boldsymbol{l_t}$**, arriving time** a_t**, deadline** d_t **and payoff** p_t**.**
  - **A set of crowd workers** W
    - **Each** $w \in W$**: location** $\boldsymbol{l_w}$**, arriving time** a_w**, deadline** d_w**, range radius** r_w**, capacity** c_w **and success ratio** δ_w**.**
  - **Utility Function:** $U(t,w) = p_t \times \delta_w$**.**
- **Find an online allocation** M **to maximize the total utility** ***MaxSum(M)***=$\sum_{t \in T, w \in W} U(t,w)$ **s.t.**
  - **Deadline Constraint.**
  - **Capacity Constraint.**
  - **Range Constraint.**
  - **Invariable Constraint (Online Scenarios Only):** *Once a task t is assigned to a worker w, the allocation of (t, w) cannot be changed.*

Offline Optimal
Total Utility=27

Problem Statement

	t_1	t_2	t_3	t_4	t_5
$w_1(1)$	**4**	-	-	-	-
$w_2(1)$	3	2	**7**	-	-
$w_3(1)$	-	-	2	**11**	-
$w_4(2)$	-	-	6	3	**5**

The task is out of the range of the crowd worker



Total utility=16

Arrival Time	8:00	8:01	8:02	8:07	8:08	8:09	8:09	8:15	8:18
1st Order	w_1	t_1	t_2	w_2	t_3	w_3	t_4	w_4	t_5
2nd Order	t_1	w_1	t_2	t_3	w_2	t_4	w_3	w_4	t_5

Total utility=27

Evaluation for Online Algorithms

- **Competitive Ratio (CR)**
  - $CR = \dfrac{\textit{Cost of an online algorithm}}{\textit{Cost of the corresponding offline algorithm}}$

- **Input Models**
  - **Adversarial Model (Worst-Case Analysis)**
    - $CR_A = min_{\forall G(T,W,U) and \forall v \in V} \dfrac{MaxSum(M)}{MaxSum(OPT)}$

The worst bipartite graph

The worst arrival order

  - **Random Order Model (Average-Case Analysis)**
    - $CR_{RO} = min_{\forall G(T,W,U)} \dfrac{\mathbb{E}[MaxSum(M)]}{MaxSum(OPT)}$

The worst bipartite graph

The expectation of the total utility of all possible arrival orders

Summary of Online Maximum Weighted Bipartite Matching Algorithms

	One-sided Online Matching		Two-sided Online Matching	
	Unweighted	**Weighted**	**Unweighted**	**Weighted**
Adversarial Model (Worst-Case Analysis)	CR_A=1- $\frac{1}{e}$ [STOC'1990]	CR_A=ln n [TCS'2015]	CR_A=0.526 [ICALP'2015]	Open Question
Random-Order Model (Average-Case Analysis)	CR_{RO}=1- $\frac{1}{e}$ [SODA'2008] The upper bound of competitive ratios of any deterministic online algorithm is CR_{RO}= $\frac{3}{4}$ [SODA'2011]	CR_{RO}= $\frac{1}{e}$ [ESA'2013]	**Our Result:** **CR_{RO}= $\frac{1}{4}$**	

Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Choose an integer *k* from 0 to** $\lceil \ln(U_{max} + 1) \rceil$ **randomly.**
  - **Filter the edges with weights lower than** e^k**.**
  - **Use a greedy strategy on the remaining edges.**

Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Choose an integer *k* from 0 to** $\lceil \ln(U_{max}+1) \rceil$ **randomly, i.e.** $\lceil \ln(11+1) \rceil = 3$, $k \in \{0,1,2,3\}$.
  - $k = 0$.
  - **Filter the edges with weights lower than** $\mathrm{e}^0 = 1$.

	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	$\boldsymbol{t_3}$	$\boldsymbol{t_4}$	$\boldsymbol{t_5}$
$\boldsymbol{w_1}$**(1)**	4	-	-	-	-
$\boldsymbol{w_2}$**(1)**	3	2	7	-	-
$\boldsymbol{w_3}$**(1)**	-	-	2	11	-
$\boldsymbol{w_4}$**(2)**	-	-	6	3	5

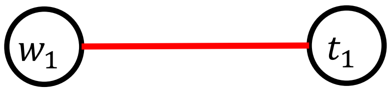
Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Filter the edges with weights lower than $\mathrm{e}^0 = 1$.**
  - **For each new arriving object, use a greedy strategy on the remaining edges.**

w_1(1)					

Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Filter the edges with weights lower than $e^0 = 1$.**
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1				
w_1(1)	4				

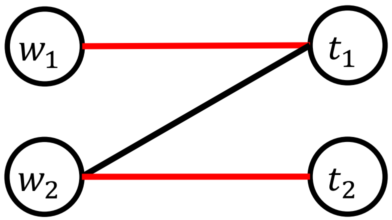
Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Filter the edges with weights lower than** $e^0 = 1$.
  - **For each new arriving object, use a greedy strategy on the remaining edges.**

	t_1	t_2			
w_1(1)	4	-			

Baseline: Extended Greedy-RT Algorithm

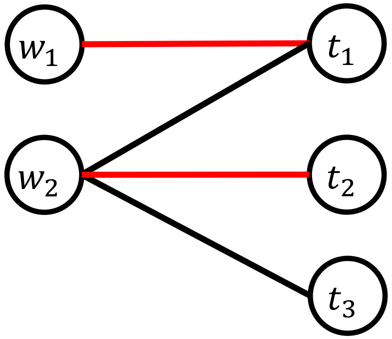
- **Basic idea**
  - **Filter the edges with weights lower than $\mathrm{e}^0 = 1$.**
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1	t_2			
$w_1(1)$	4	-			
$w_2(1)$	3	2			

Baseline: Extended Greedy-RT Algorithm

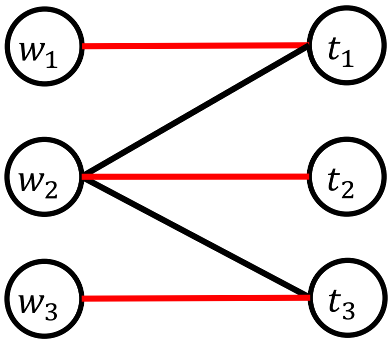
- **Basic idea**
  - **Filter the edges with weights lower than $e^0 = 1$.**
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1	t_2	t_3		
$w_1(1)$	4	-	-		
$w_2(1)$	3	2	7		

Baseline: Extended Greedy-RT Algorithm

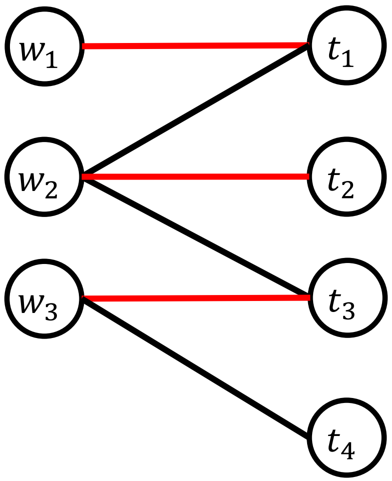
- **Basic idea**
  - **Filter the edges with weights lower than $\mathrm{e}^0 = 1$.**
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1	t_2	t_3		
$w_1(1)$	4	-	-		
$w_2(1)$	3	2	7		
$w_3(1)$	-	-	2		

Baseline: Extended Greedy-RT Algorithm

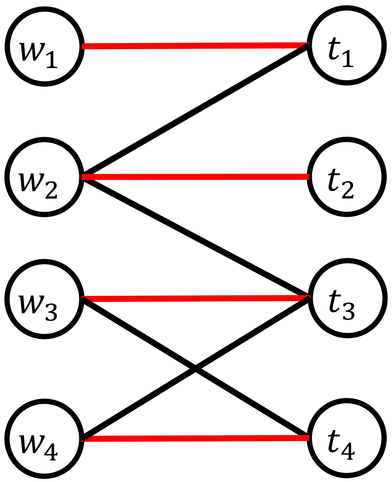
- **Basic idea**
  - **Filter the edges with weights lower than** $e^0 = 1$.
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1	t_2	t_3	t_4	
$w_1(1)$	4	-	-	-	
$w_2(1)$	3	2	7	-	
$w_3(1)$	-	-	2	11	

Baseline: Extended Greedy-RT Algorithm

- **Basic idea**
  - **Filter the edges with weights lower than** $\mathrm{e}^0 = 1$.
  - **For each new arriving object, use a greedy strategy on the remaining edges.**

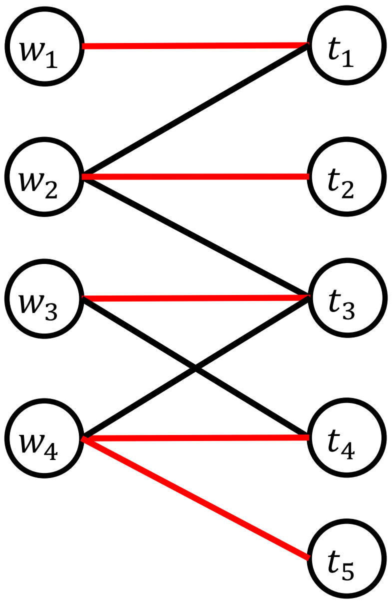


	t_1	t_2	t_3	t_4	
w_1(1)	4	-	-	-	
w_2(1)	3	2	7	-	
w_3(1)	-	-	2	11	
w_4(2)	-	-	6	3	

Baseline: Extended Greedy-RT Algorithm

Competitive Ratio $CR_A = \frac{1}{2e\ln(1+U_{max})}$

- **Basic idea**
  - **Filter the edges with weights lower than** $e^0 = 1$.
  - **For each new arriving object, use a greedy strategy on the remaining edges.**



	t_1	t_2	t_3	t_4	t_5
w_1(1)	4	-	-	-	-
w_2(1)	3	2	7	-	-
w_3(1)	-	-	2	11	-
w_4(2)	-	-	6	3	5

  - **When k=0, the utility is 16. For all possible values of k, the expectation of utilities is** $\frac{16+27+11+0}{4} = 13.5$

TGOA Algorithm

- **Basic idea**
  - **Take a fixed fraction of arriving objects as samples and dispose them greedily.**
  - **When a new object arrives, compute the optimal matching on the revealed part of the graph.**
  - **Match the new object to its adjacent node in the optimal matching if possible.**

TGOA Algorithm

- **Basic idea**
  - **The first half of objects is filtered and matched greedily.**

w_1

$\boldsymbol{w_1}$**(1)**					

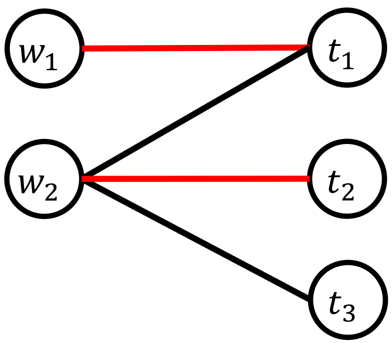
TGOA Algorithm

- **Basic idea**
  - **A half of objects is filtered and matched greedily.**

	t_1				
$w_1(1)$	4				

TGOA Algorithm

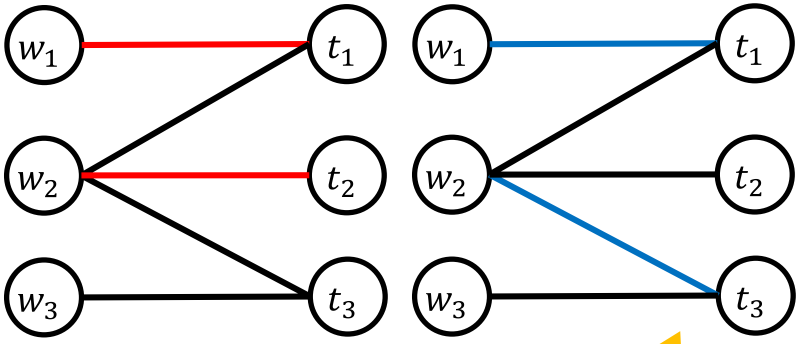
- **Basic idea**
  - **A half of objects is filtered and matched greedily.**



	t_1	t_2	t_3		
$w_1(1)$	4	-	-		
$w_2(1)$	3	2	7		

TGOA Algorithm

- **Basic idea**
  - **For the second half of objects, once a new object arrives, compute the optimal matching on the revealed part of the graph.**
  - **Match the new object to its adjacent node in the optimal matching if possible.**

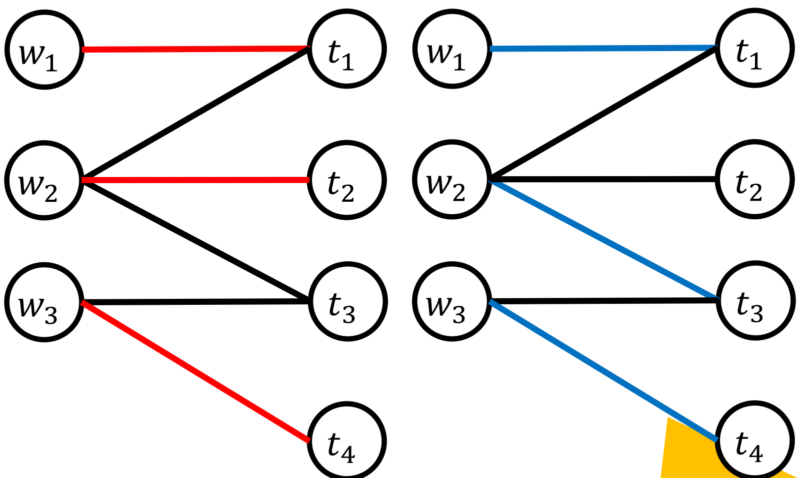


The optimal matching on the revealed part of graph, so no new edge is inserted.

	t_1	t_2	t_3		
$w_1(1)$	4	-	-		
$w_2(1)$	3	2	7		
$w_3(1)$	-	-	2		

TGOA Algorithm

- **Basic idea**
  - **For the second half of objects, once a new object arrives, compute the optimal matching on the revealed part of the graph.**
  - **Match the new object to its adjacent node in the optimal matching if possible.**

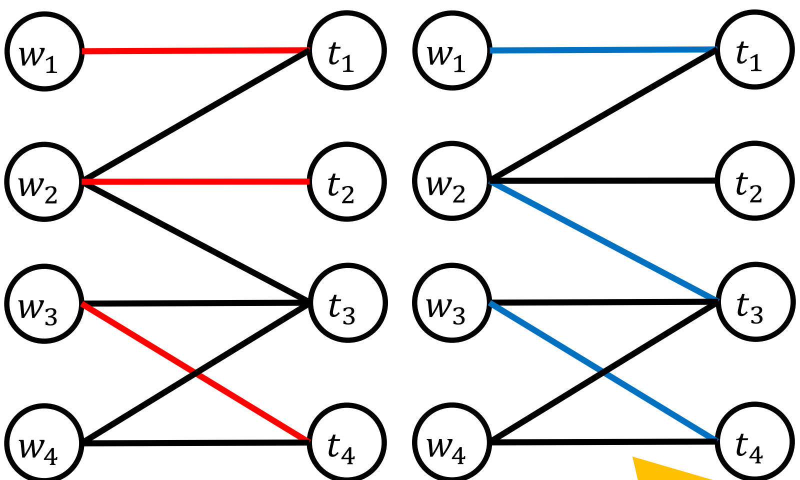


	t_1	t_2	t_3	t_4	
$w_1(1)$	4	-	-	-	
$w_2(1)$	3	2	7	-	
$w_3(1)$	-	-	2	11	

The optimal matching on the revealed part of graph, the new edge (w_3, t_4) is inserted

TGOA Algorithm

- **Basic idea**
  - **For the second half of objects, once a new object arrives, compute the optimal matching on the revealed part of the graph.**
  - **Match the new object to its adjacent node in the optimal matching if possible.**



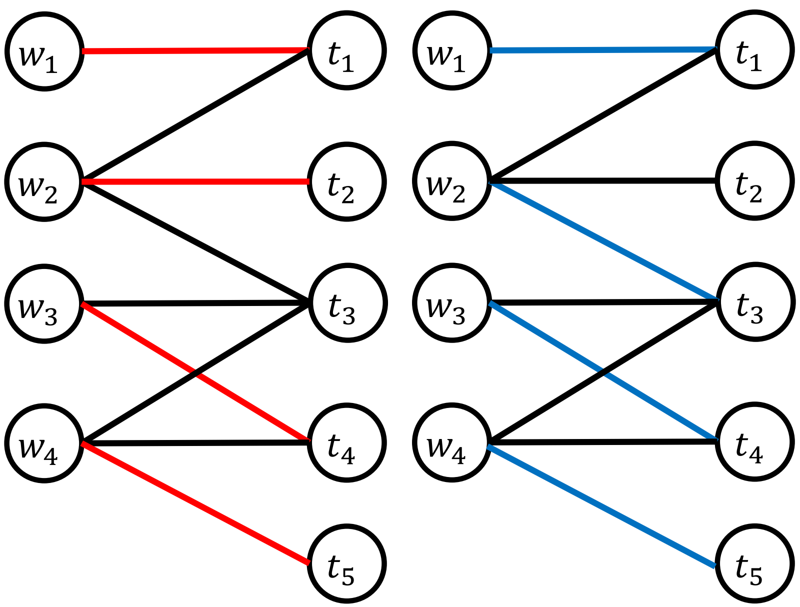
	t_1	t_2	t_3	t_4	
w_1(1)	4	-	-	-	
w_2(1)	3	2	7	-	
w_3(1)	-	-	2	11	
w_4(2)	-	-	6	3	

The optimal matching on the revealed part of graph, so no new edges is inserted.

TGOA Algorithm

Competitive Ratio $CR_{RO}=\frac{1}{4}$

- **Basic idea**
  - **For the second half of objects, once a new object arrives, compute the optimal matching on the revealed part of the graph.**
  - **Match the new object to its adjacent node in the optimal matching if possible.**



	t_1	t_2	t_3	t_4	t_5
w_1(1)	4	-	-	-	-
w_2(1)	3	2	7	-	-
w_3(1)	-	-	2	11	-
w_4(2)	-	-	6	3	5

- **For the arrival order, the utility of TGOA is 22.**

TGOA-Greedy Algorithm

- **Basic idea**

Competitive Ratio $CR_{RO}=\frac{1}{8}$

  - **Although TGOA provides a better competitive ratio, it has high computational complexity due to the offline optimal matching algorithm, e.g. Hungary algorithm.**

  - **Optimize the efficiency using a greedy solution to get the offline matching instead of the offline optimal matching in the second phase.**

Experimental Setting

- **Real Datasets**
  - **WeSender (|T|=4036, |W|=817)**
  - **gMission (|T|=713, |W|=532)**

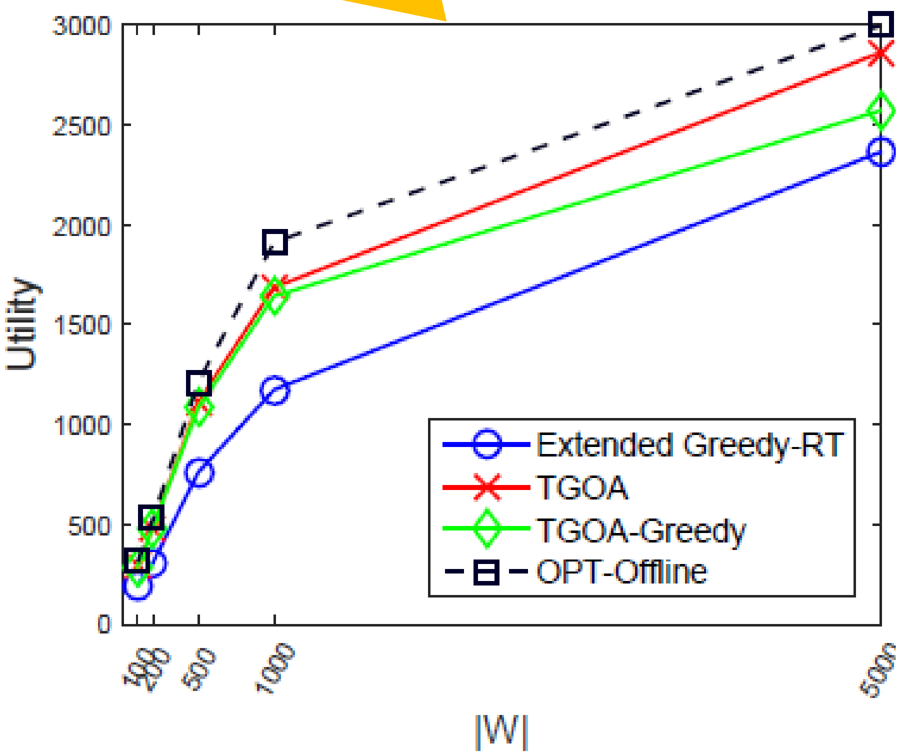
Factor	Setting
$\|T\|$	500, 1000, **2500**, 5000, 10000
$\|W\|$	100, 200, **500**, 1000, 5000
c_w	1, 2, **5**, 10, 20
r_w	1.0, 1.5, **2.0**, 2.5, 3.0
δ_w	0.1, 0.3, **0.5**, 0.7, 0.9
Deadline	2, 4, **6**, 8, 10
p_t	2, 5, **10**, 20, 50
Scalability	$\|T\|$ = 10K, 20K, 30K, 40K, 50K, 100K $\|W\|$ = 500, 1000, 2500

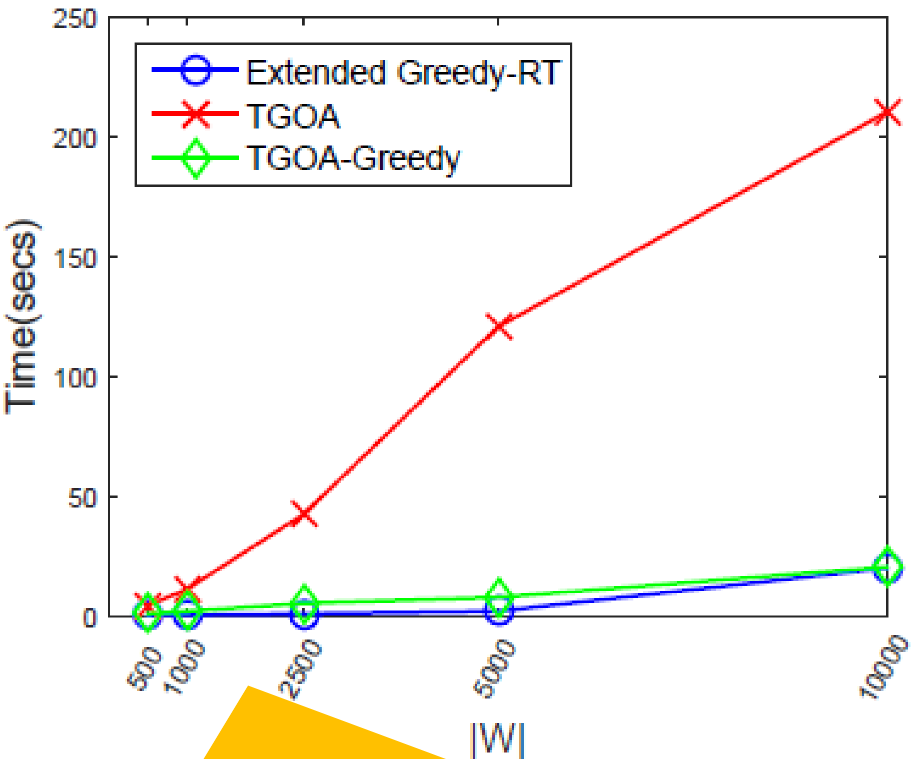
- **Synthetic Dataset**
  - **|T|: The number of spatial tasks.**
  - **|W|: The number of crowd workers.**
  - c_w**: The maximum workload of crowd workers.**
  - r_w**: The range radius of crowd workers.**
  - δ_w **: The average success ratio of crowd workers.**
  - d_t/d_w **: The deadlines of spatial tasks and crowd workers.**
  - p_t**: The average payoffof spatial tasks.**

- **Compared Algorithms**
  - **Extended Greedy-RT (Baseline), TGOA and TGOA-Greedy**

Experiments: Vary |T|

TGOA is the most effective but most inefficient.

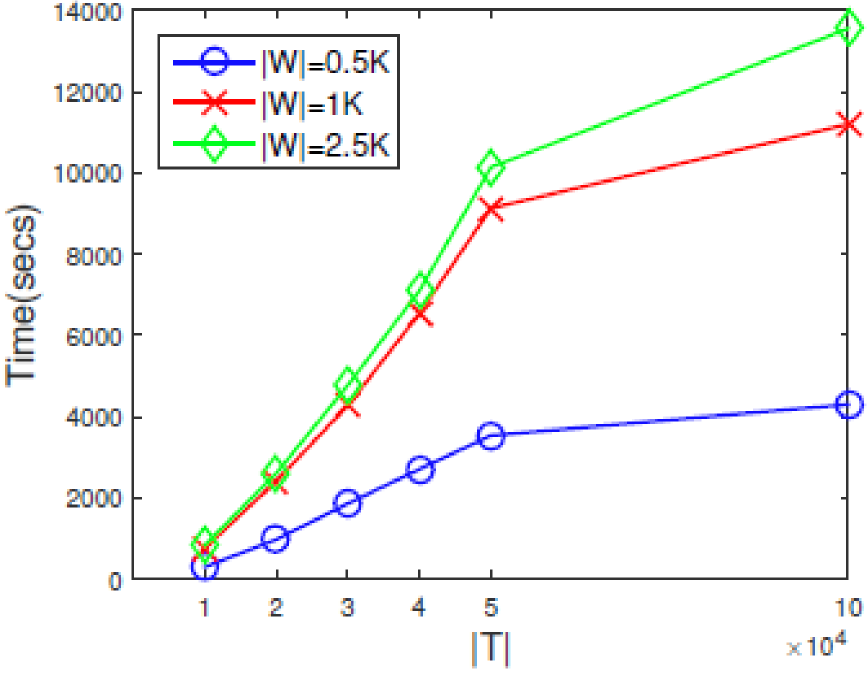


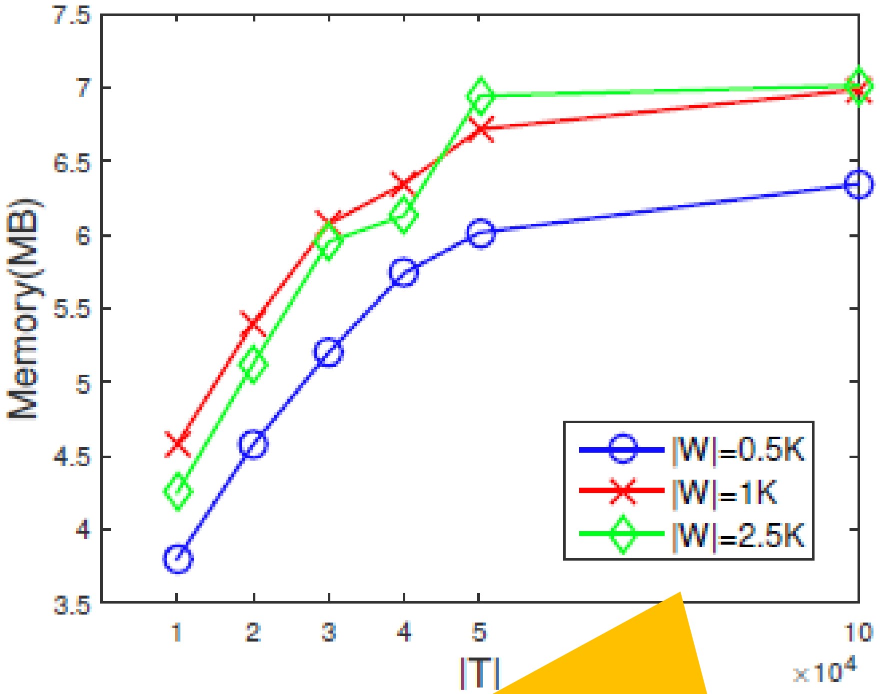


TGOA-greedy trades off well between effectiveness and efficiency!

Extended Greedy-RT (Baseline algorithm) is the fastest one, but its effectiveness is the lowest!

Experiments: Scalability





Different curves mean the different number of crowd workers.

Conclusion

- Identify a new two-sided online micro-task allocation problem, called Global Online Micro-task Allocation (GOMA), in spatial crowdsourcing.

- Design a two-phase-based framework with a constant competitive ratio under the random order model and a greedy optimization technique.

- Extensive experiments on both real and synthetic datasets to verify our solutions.

Outline

- Human Computation in a Nutshell
- State-of-art
  - A Review of Crowdsourced Databases
  - **Task Assignment**
    - Online Mobile Micro-Task Assignment [ICDE'2016]
    - **Online Matching in Real-Time Spatial Data [VLDB'2016]**
  - Quality Control
    - Whom to Ask [VLDB'2012]
- Crowdsourcing in a Vision

Review of Taxi Assignment

- **Real-time taxi-calling services usually adopt 'nearest neighbor (NN)' strategy to address task assignment issues.**
  - Once a task appears, it should be assigned immediately.
  - If we know everything in advance, the offline OPT is shown

Problem Statement

- **Online Minimum Bipartite Matching (OMBM)**
- **Given**
  - **A set of (static) crowd workers** W
    - **Each** $w \in W$**: location** $\boldsymbol{l_w}$**.**
  - **A set of (dynamic) spatial tasks** T
    - **Each** $t \in T$**: location** $\boldsymbol{l_t}$**, arriving time** a_t**.**
  - **Cost Function:** ***Any Metric Distance Function*****.**
- **Find an online allocation** M **to minimize the total cost** ***MinSum(M)*****=**$\sum_{t\in T, w\in W} \boldsymbol{dis}(t, w)$ **s.t.**
  - **Cardinality Constraint: |M|=min{|T|, |W|}**
  - **Real-Time Constraint (Online Scenarios):** *Once a task t appears, a worker must be immediately assigned to t* ***before*** *the next task appears.*
  - **Invariable Constraint (Online Scenarios):** *Once a task t is assigned to a worker w, the allocation of (t, w)* ***cannot be changed****!*

Review of Taxi Assignment

- **Real-time taxi-calling services usually adopt 'nearest neighbor (NN)' strategy to address task assignment issues.**
  - Once a task appears, it should be assigned immediately.

Evaluation for Online Algorithms

- **Competitive Ratio (CR)**
  - $CR = \frac{Cost\ of\ an\ online\ algorithm}{Cost\ of\ the\ corresponding\ offline\ algorithm}$

- **Input Models**
  - **Adversarial Model (Worst-Case Analysis)**
    - $CR_A = \mathbf{max}_{\forall G(T,W,U) and \forall v \in V} \frac{MaxSum(M)}{MaxSum(OPT)}$

The worst bipartite graph

The worst arrival order

  - **Random Order Model (Average-Case Analysis)**
    - $CR_{RO} = \mathbf{max}_{\forall G(T,W,U)} \frac{\mathbb{E}[MaxSum(M)]}{MaxSum(OPT)}$

The worst bipartite graph

The expectation of the total cost of all possible arrival orders

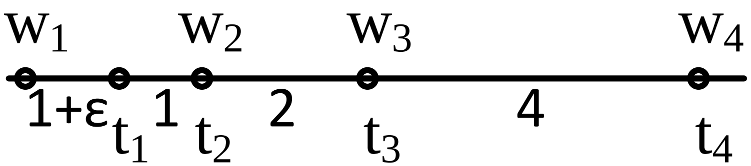
Existing Online Minimum Bipartite Matching Algorithms

Algorithms	Time Complexity per Each Arrival Vertex	Randomization	Data Structure	Competitive Ratio
Greedy [STOC'1990]	$O(k)$	Deterministic	No	$O(2^k)$
Permutation [SODA'1991]	$O(k^3)$	Deterministic	No	$O(2k-1)$
HST-Greedy [SODA 2006]	$O(k)$	Randomized	HST	$O(\log^3 k)$
HST-Reassignment [ESA 2007]	$O(k^2)$	Randomized	HST	$O(\log^2 k)$

Hierarchically Separated Tree

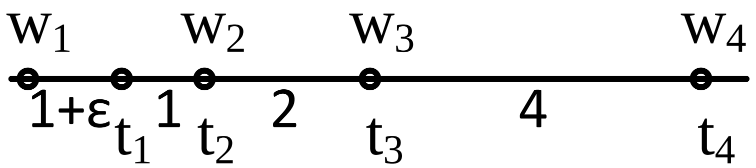
Is the greedy algorithm really the worst?

Greedy Revisited

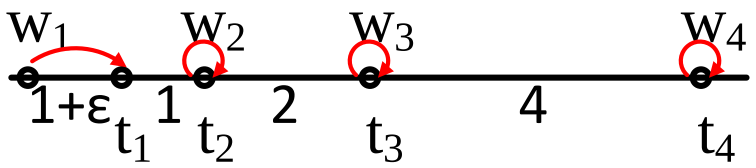


(a)Locations

Greedy Revisited

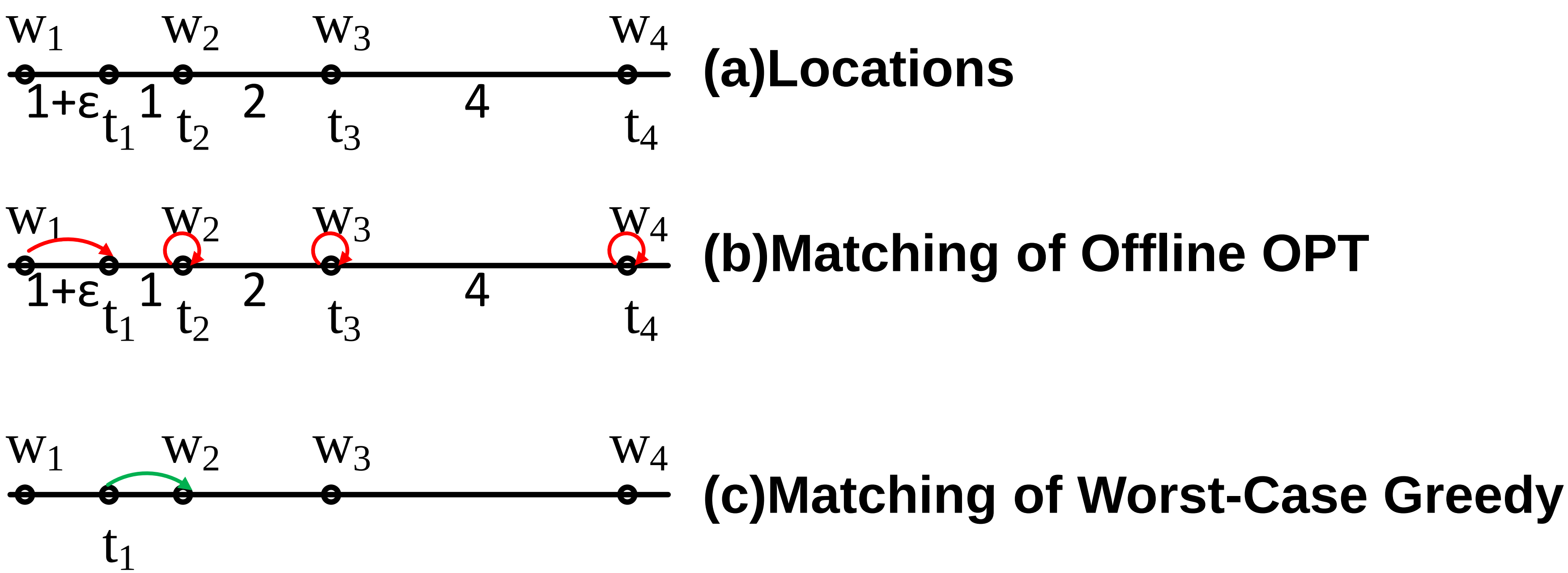


(a)Locations



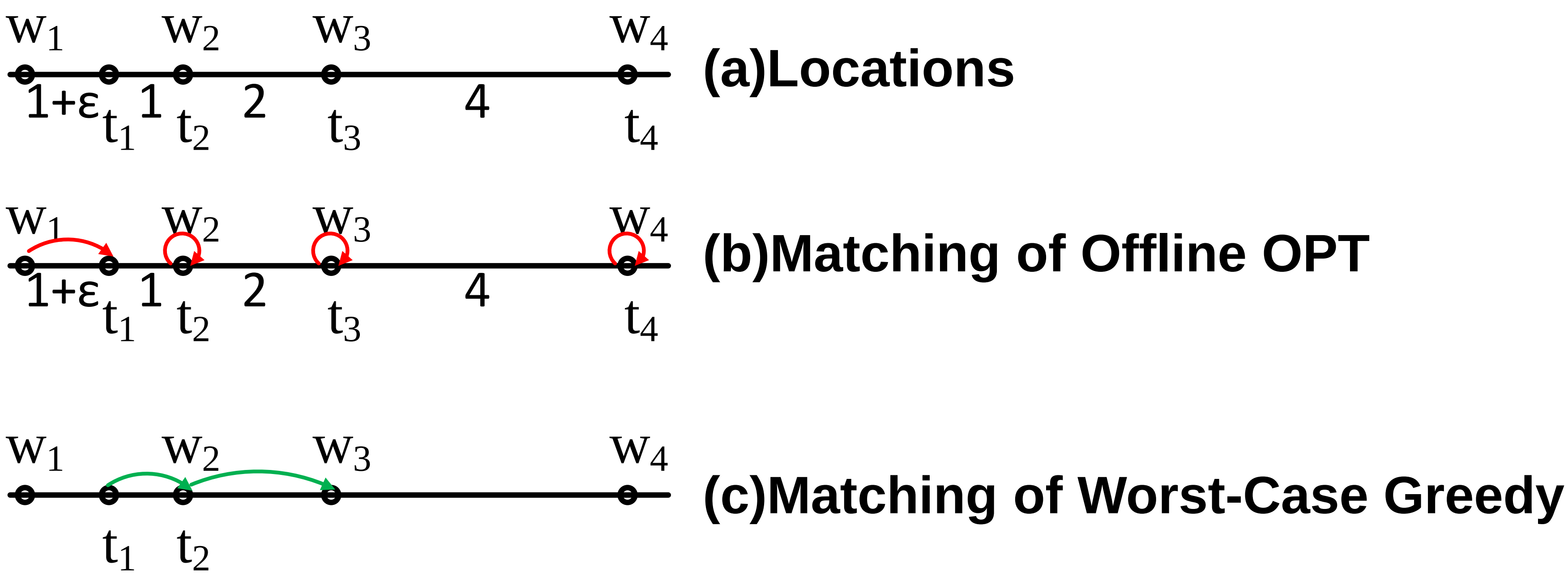
(b)Matching of Offline OPT

Greedy Revisited



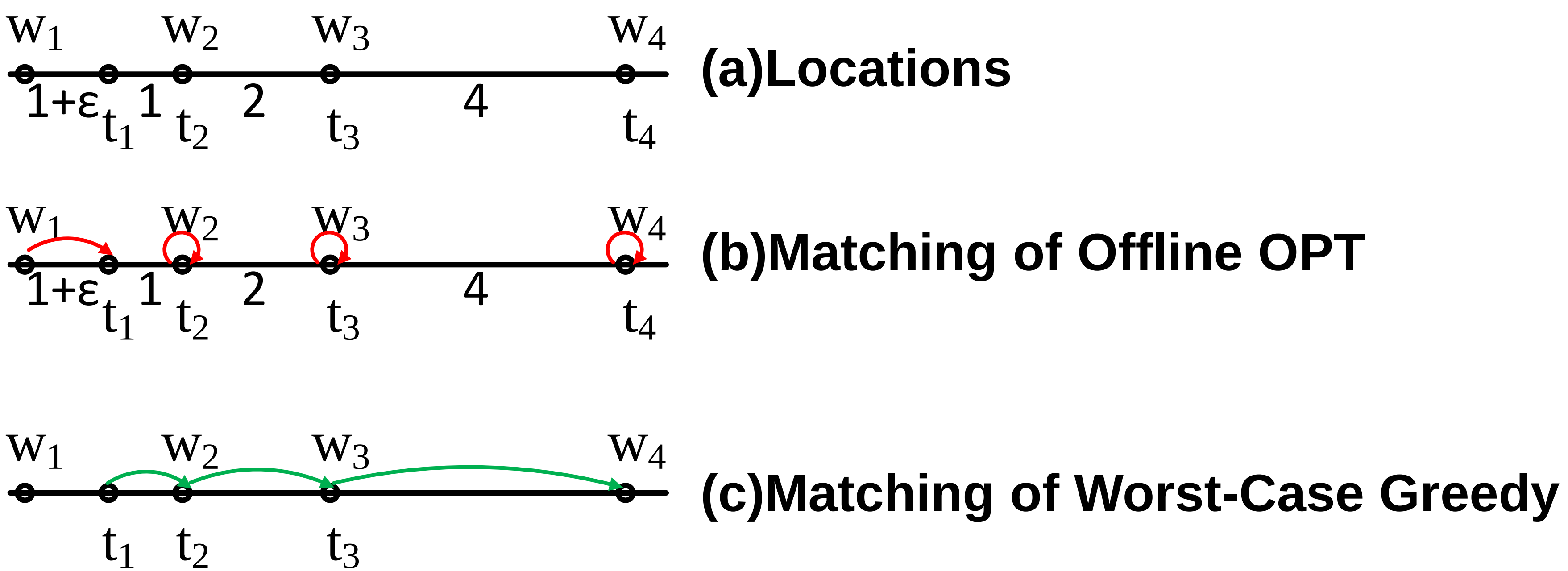
When t_1 appears, w_2 will be assigned to t_1 by Greedy.

Greedy Revisited



When t_2 appears, w_3 will be assigned to t_2 by Greedy.

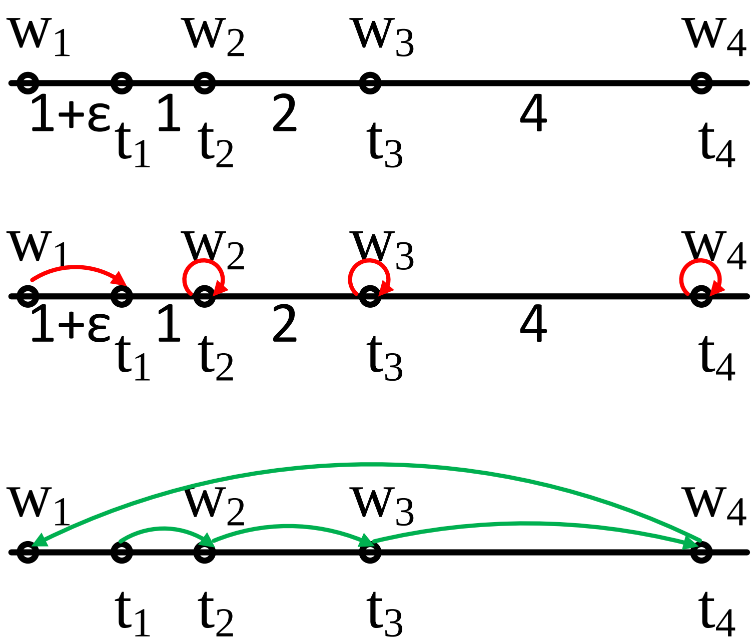
Greedy Revisited



When t_3 appears, w_4 will be assigned to t_3 by Greedy.

Greedy Revisited

Competitive Ratio CR_{RO}=3.195



(a)Locations

(b)Matching of Offline OPT

(c)Matching of Worst-Case Greedy

When t_4 appears, w_1 has to be assigned to t_4 by Greedy.

Open question: the average-case competitive ratio under the random order model of Greedy for the OMBM problem should be constant!

Experimental Setting

- **Real Datasets (Shenzhou Taxi “神州专车”)**
  - **15082 Shenzhou Taxis at Beijing in May 2015)**
  - **Average 115364 calling-taxi requests per day**

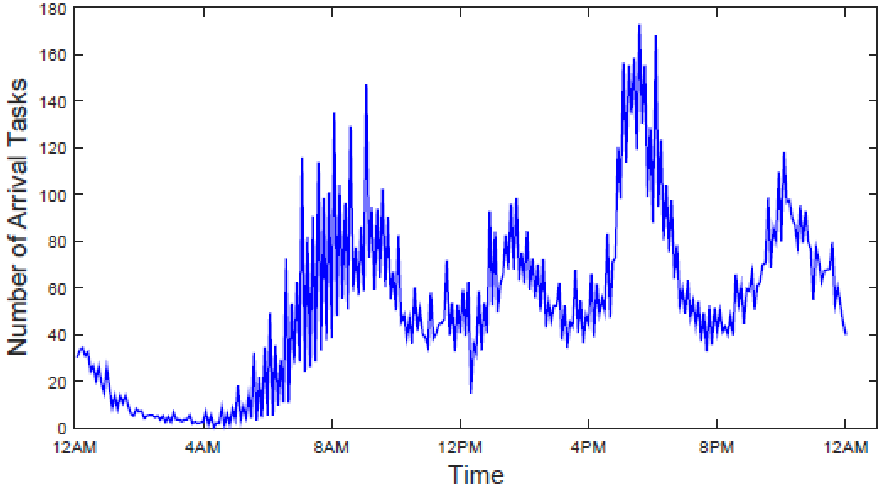


Figure 5: Average number of tasks of taxi-calling per day

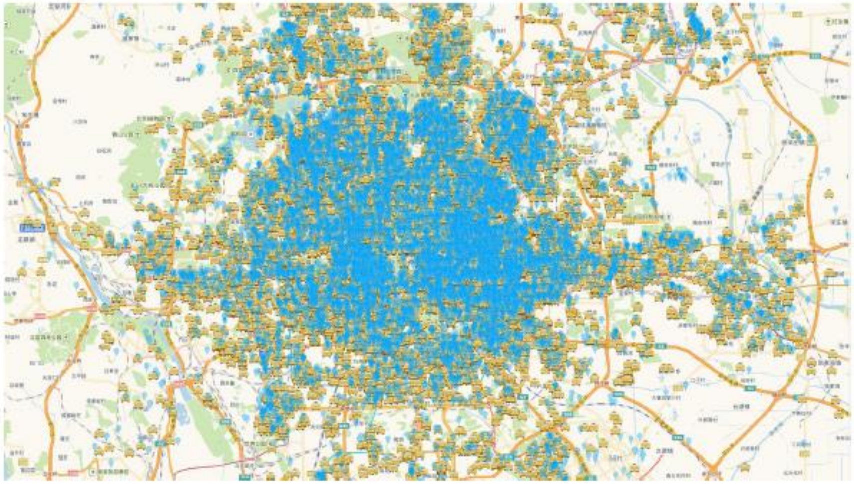
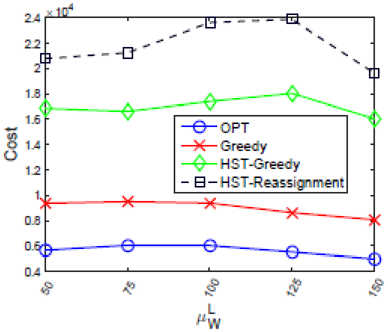


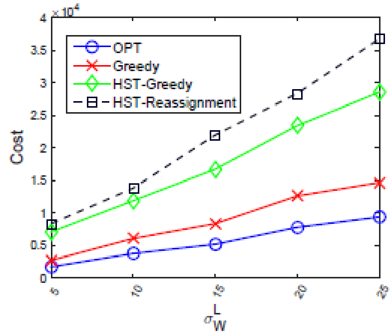
Figure 6: Location distribution of the users and service providers at the ShenZhou taxi-calling platform on one day in Beijing

- **Synthetic Dataset**
  - **|T| (The number of spatial tasks)=5000**
  - **|W|(The number of crowd workers)=5000**
  - **The locations of tasks randomly follows Normal distribution, Uniform distribution, Power-law distribution and Exponential distribution.**

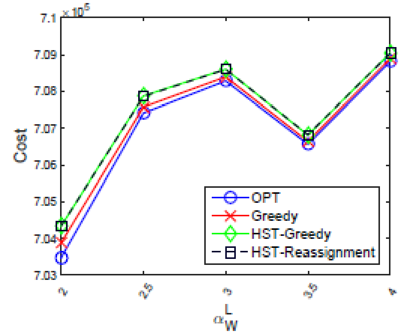
Experiments: Task (Normal)



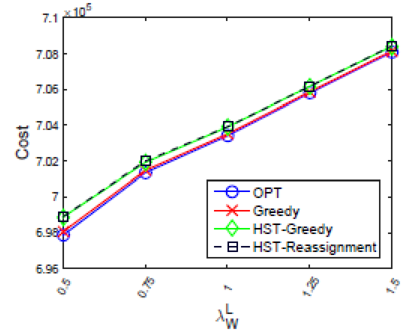
(a) Cost of varied Normal μ_W^L



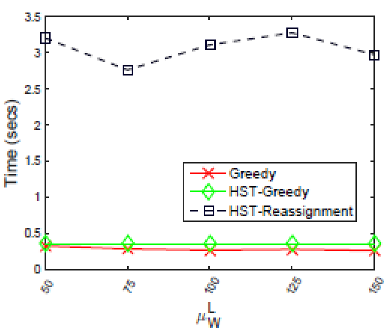
(b) Cost of varied Normal σ_W^L



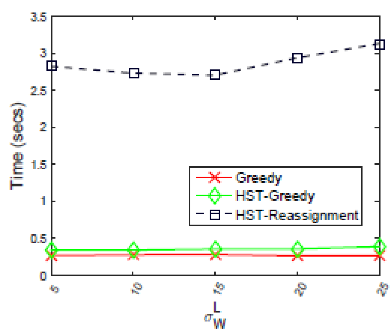
(c) Cost of varied Power α_W^L



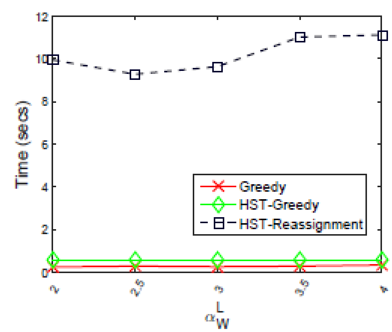
(d) Cost of varied Exp λ_W^L



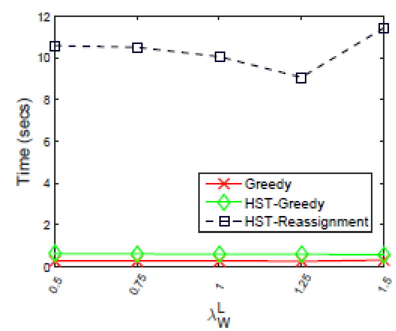
(e) Time of varied Normal μ_W^L



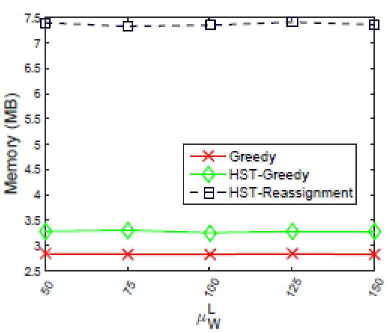
(f) Time of varied Normal σ_W^L



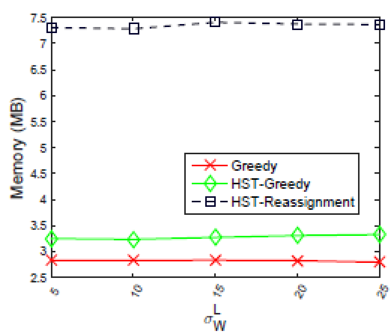
(g) Time of varied Power α_W^L



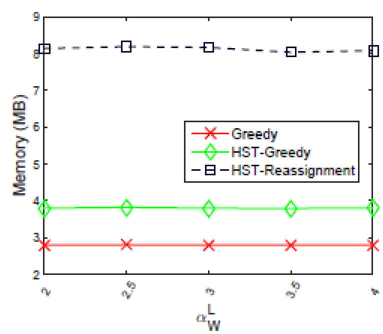
(h) Time of varied Exp λ_W^L



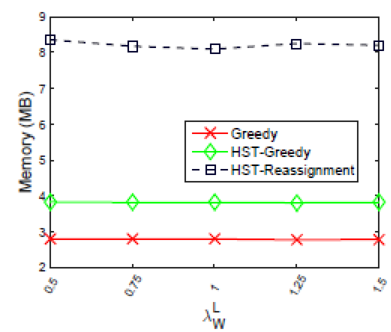
(i) Memory of varied Normal μ_W^L



(j) Memory of varied Normal σ_W^L

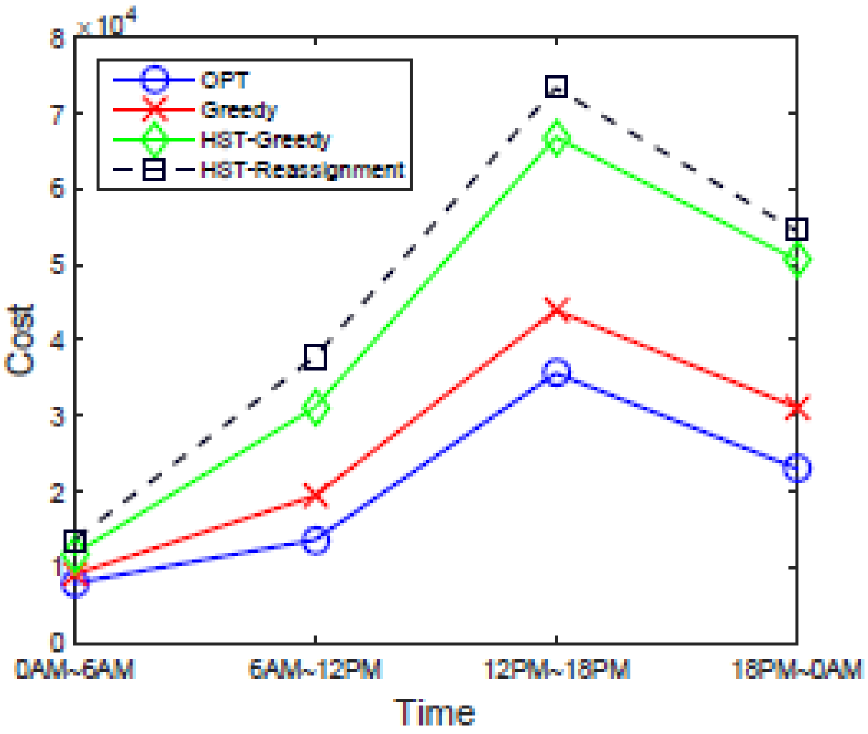


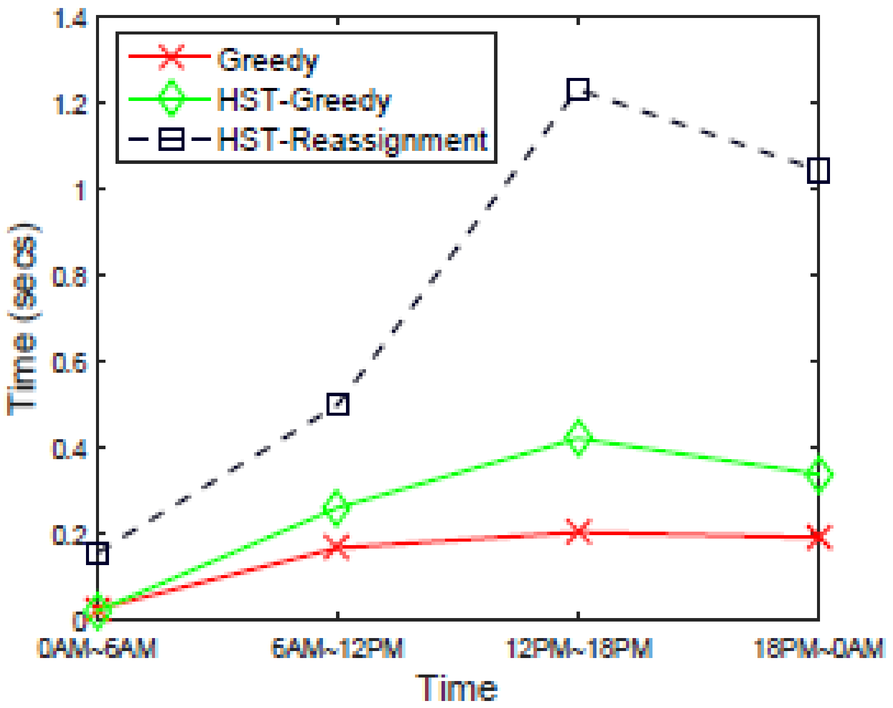
(k) Memory of varied Power α_W^L



(l) Memory of varied Exp λ_W^L

Experiments: Real Datasets





For total distance, Greedy is almost 2 times better than the other two online algorithms!

Conclusion

- **Our most important experimental finding is that both the efficiency and the effectiveness of *Greedy significantly outperforms the other algorithms in almost all practical cases.***
- ***We clarify the 25-year misunderstanding* towards Greedy for the OMBM problem through the experimental study!**
- **According to extensive random experiments, we propose the a open question: the average-case competitive ratio of Greedy for the OMBM problem should be constant.**

Outline

- Human Computation in a Nutshell
- State-of-art
  - A Review of Crowdsourced Databases
  - **Task Assignment**
    - Online Mobile Micro-Task Assignment [ICDE’2016]
    - Online Matching in Real-Time Spatial Data [VLDB’2016]
  - **Quality Control**
    - **Whom to Ask [VLDB’2012]**
- Crowdsourcing in a Vision

Whom to Ask?

- "Which venue held the latest International Film Festival in Hong Kong?"

Andy Lau

"HK Coliseum"

Cecilia Cheung

"? ? ?"

Nicholas Tse

"Hong Kong Cultural Centre"

Jackie Chan

"HK Coliseum"

Whom to Ask?

- “What’s the next breakthrough in Big Data”

Andy Lau

“? ? ?”

Cecilia Cheung

“? ? ?”

Nicholas Tse

“? ? ?”

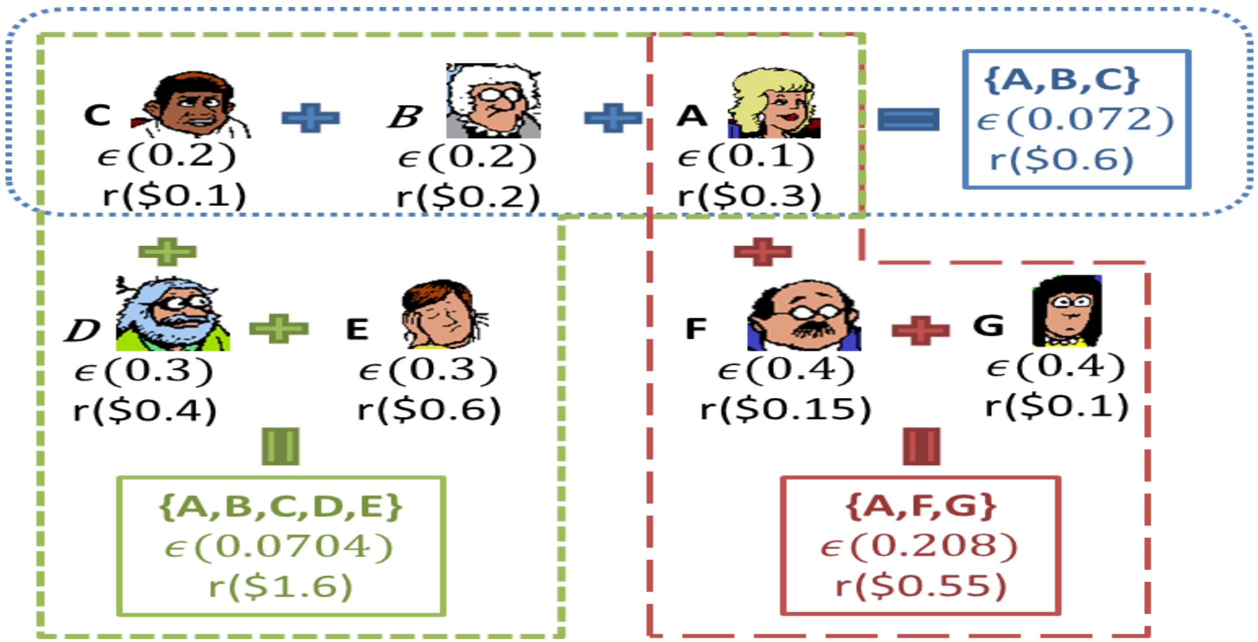
Jackie Chan

“? ? ?”

Running Example

- "Is it possible to swim in the Silverstrand Beach in August?"

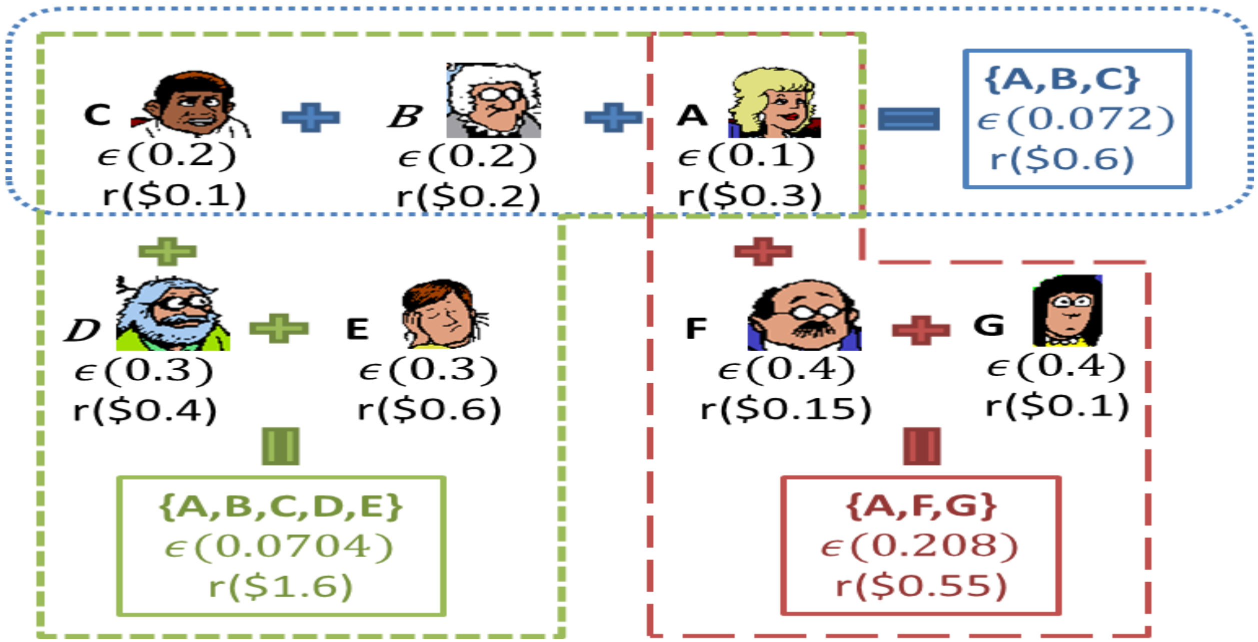
Motivation – Jury Selection Problem Running Case(1)



"Is it possible to swim in the Silverstrand Beach in August?"

- Given a decision making problem, with budget $1, **whom should we ask**?

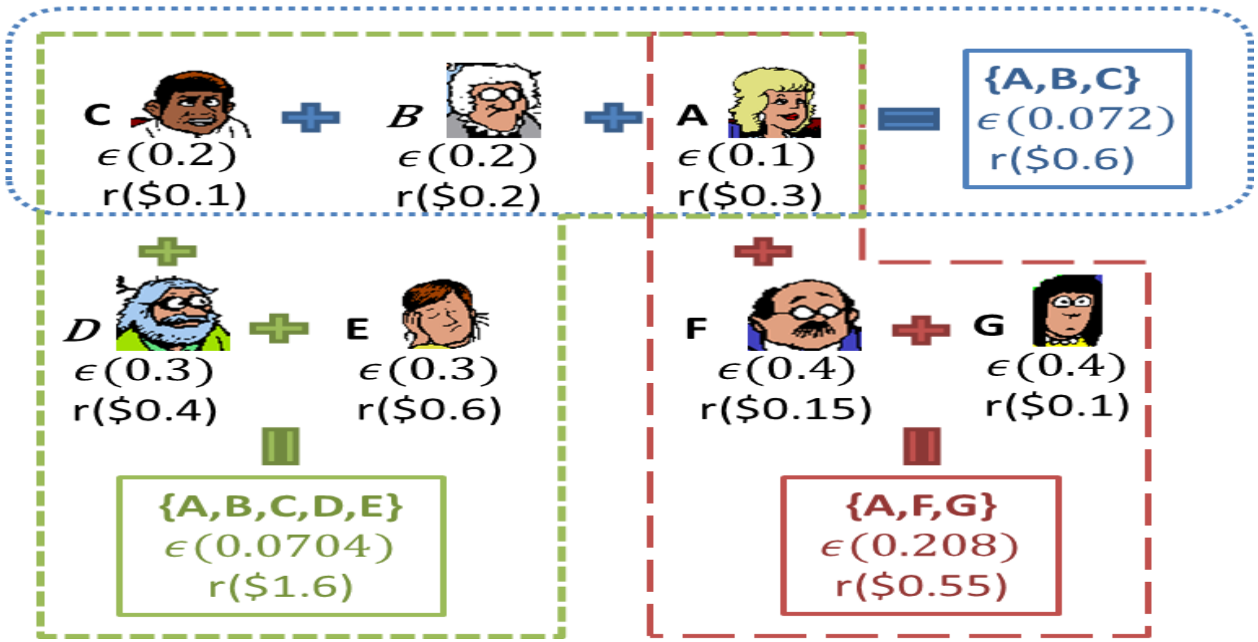
Motivation – Jury Selection Problem Running Case(2)



"Is it possible to swim in the Silverstrand Beach in August?"

- ε: error rate of an individual
- r: requirement of an individual, can be virtual
- Majority voting to achieve the final answer

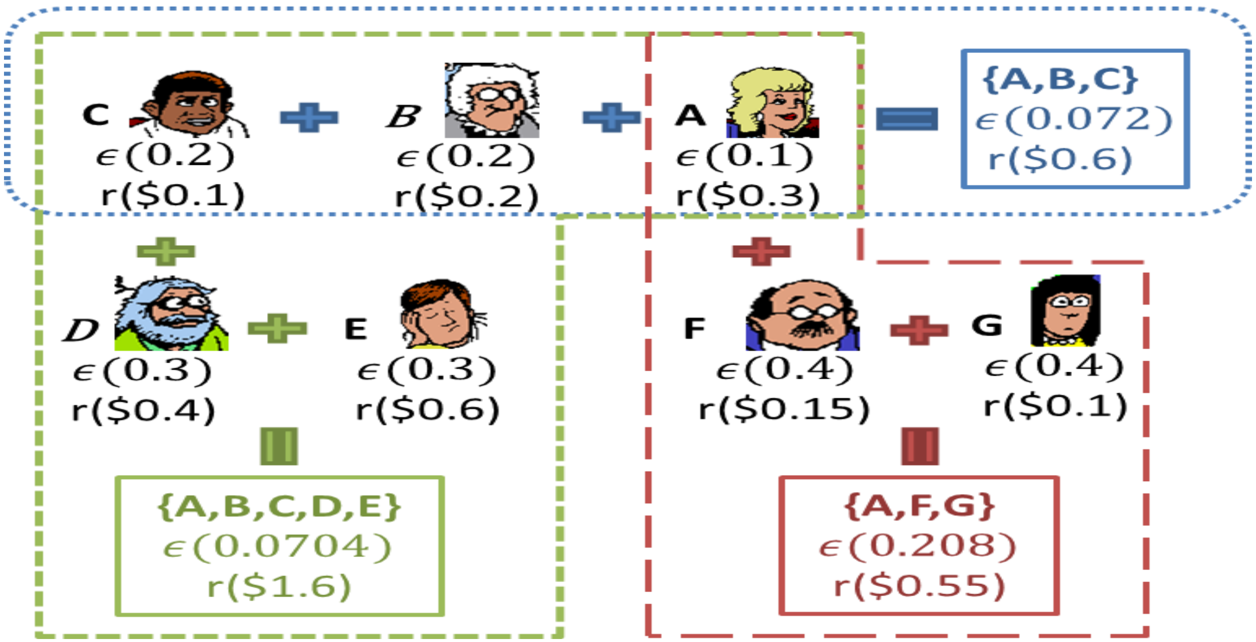
Motivation – Jury Selection Problem Running Case(3)



"Is it possible to swim in the Silverstrand Beach in August?"

- Worker : Juror
- Crowds : Jury
- Data Quality : Jury Error Rate

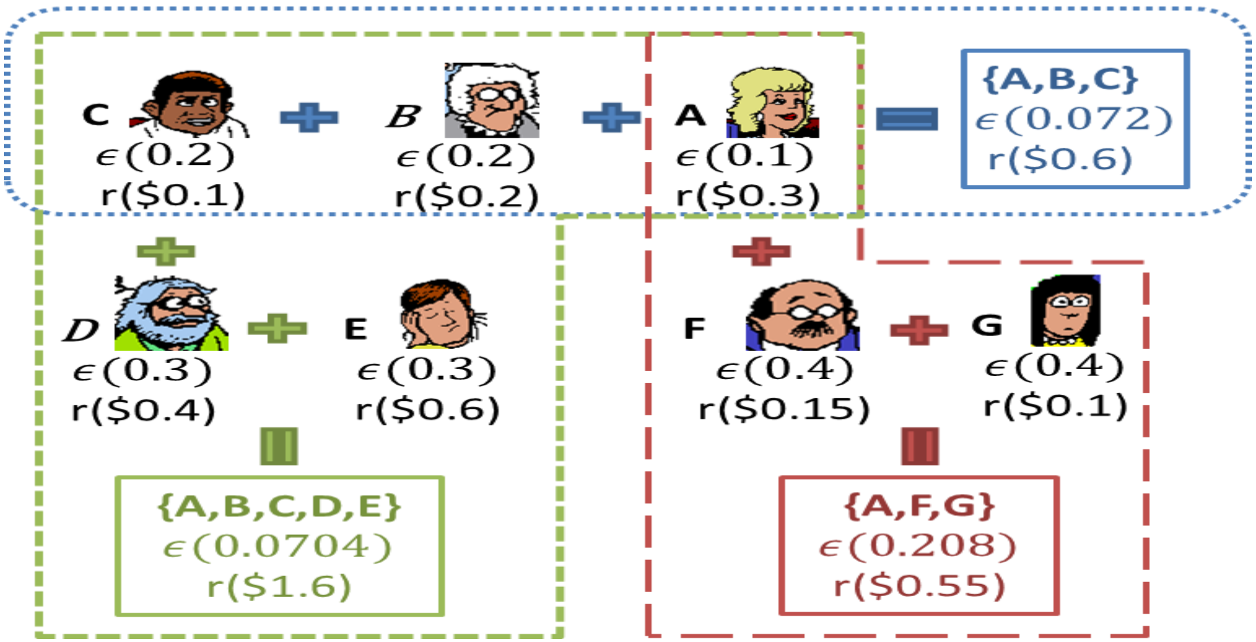
Motivation – Jury Selection Problem Running Case(4)



"Is it possible to swim in the Silverstrand Beach in August?"

- If (A, B, C) are chosen(Majority Voting)
  - JER(A,B,C) = 0.1*0.2*0.2 + (1 - 0.1)*0.2*0.2 + 0.1* (1 - 0.2)*0.2 + 0.1*0.2*(1 - 0.2) = 0.072
  - Better than A(0.1), B(0.2) or C(0.2) individually

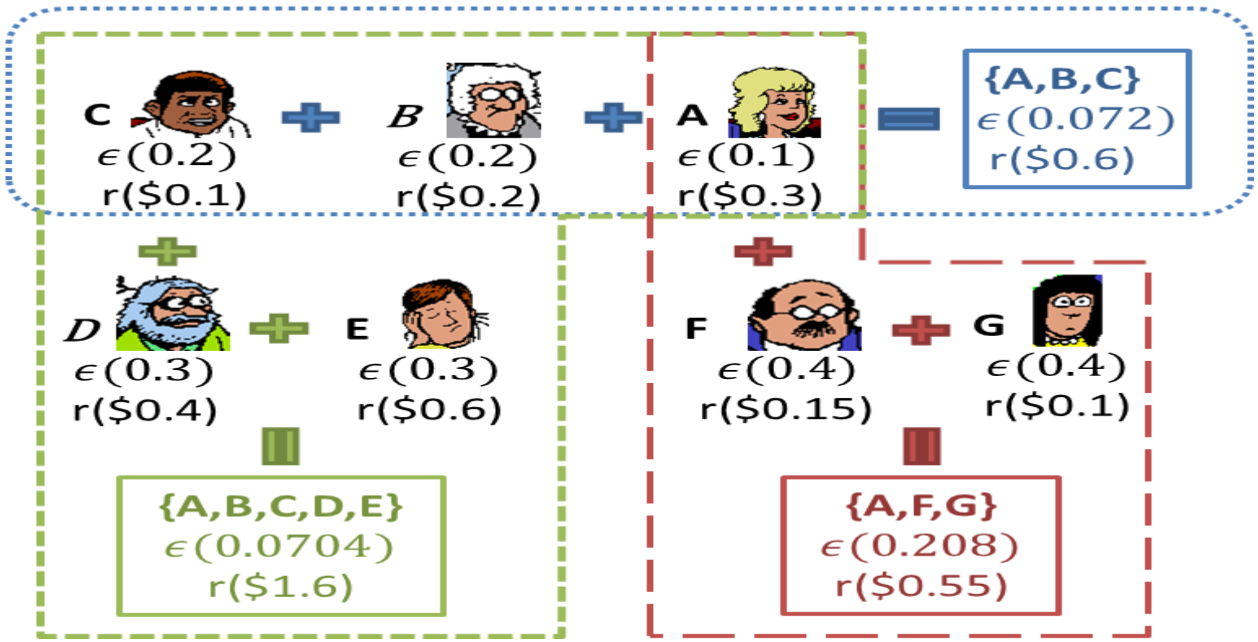
Motivation – Jury Selection Problem Running Case(5)



"Is it possible to swim in the Silverstrand Beach in August?"

- What if we enroll more
  - JER(A,B,C,D,E) = 0.0704 < JER(A,B,C)
  - The more the better?

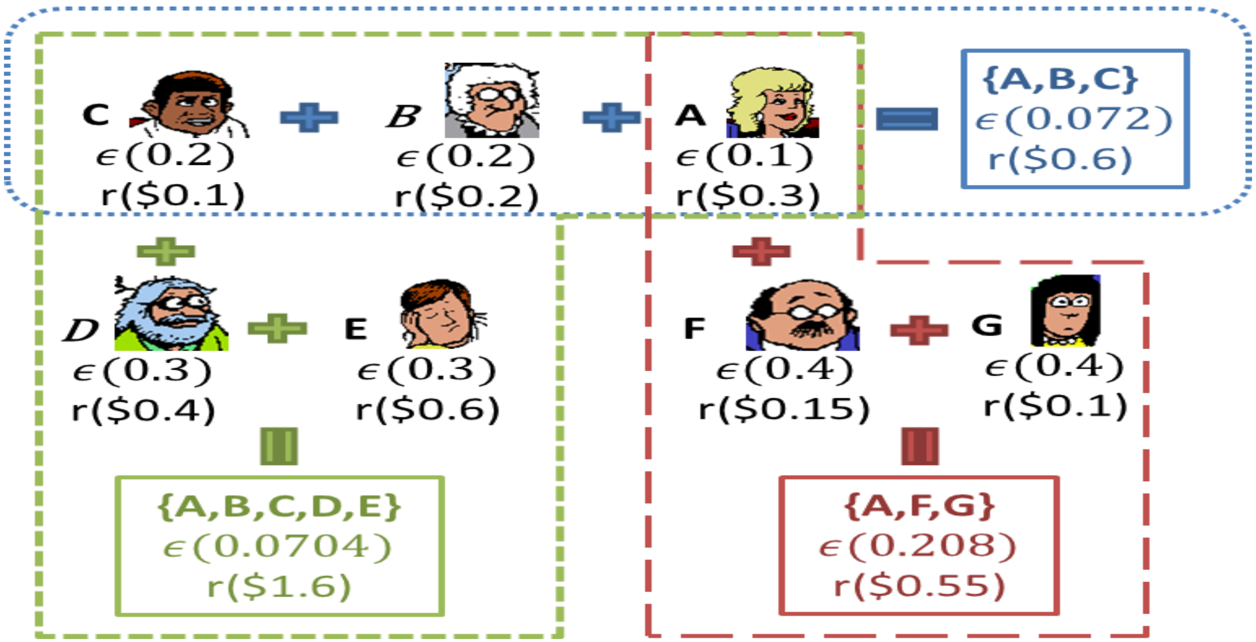
Motivation – Jury Selection Problem Running Case(6)



"Is it possible to swim in the Silverstrand Beach in August?"

- What if we enroll even more?
  - JER(A,B,C,D,E,F,G) = 0.0805 > JER(A,B,C,D,E)
  - Hard to calculate JER

Motivation – Jury Selection Problem Running Case(7)



"Is it possible to swim in the Silverstrand Beach in August?"

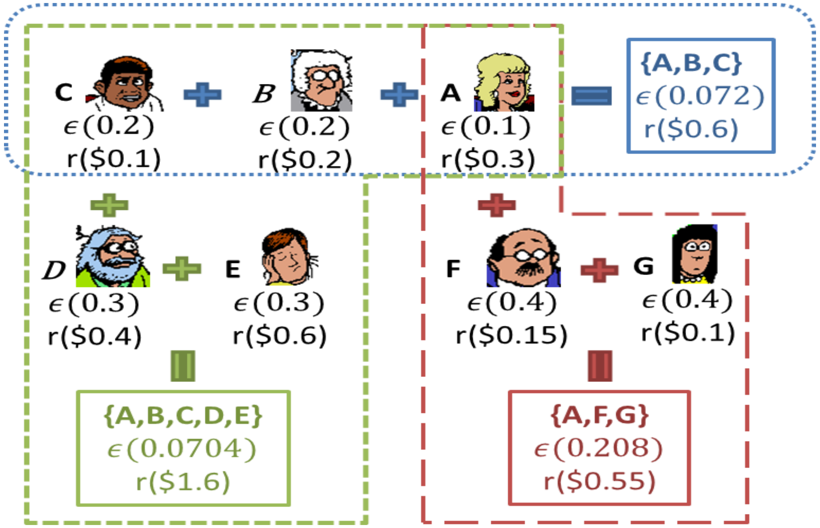
- So just pick up the best combination?
  - JER(A,B,C,D,E)=0.0704
  - R(A,B,C,D,E) = $1.6 > budget($1.0)

Motivation – Jury Selection Problem Running Case(8)

Crowd	Individual Error-rate	Jury Error-rate
C	0.2	0.2
A	0.1	0.1
C,D,E	0.2,0.2,0.3	0.174
A,B,C	0.1,0.2,0.2	0.072
A,B,C,D,E	0.1,0.2,0.2,0.3,0.3	0.0703
A,B,C,D,E,F,G	0.1,0.2,0.2,0.3,0.3,0.4,0.4	0.0805

Worker selection for maximize the quality of a particular type of product:

The reliability of voting.



Outline

- Human Computation in a Nutshell
- State-of-art
  - A Review of Crowdsourced Databases
  - Task Assignment
    - Online Mobile Micro-Task Assignment [ICDE'2016]
    - Online Matching in Real-Time Spatial Data [VLDB'2016]
  - Quality Control
    - Whom to Ask [VLDB'2012]
- **Crowdsourcing in a Vision**

Research in Crowdsourcing

- Crowdsourced Science
  - Traditional Science that enhanced by crowdsourcing

- Science of Crowdsourcing
  - The characteristics of Human Computation as new hardware

Science of Crowdsourcing

- Privacy Preserving
  - How to preserve the privacy of crowd workers?
- Incentive Mechanism
  - Didi Taxi vs. Uber
- Gamification & Quality Control
  - How to make it fun?

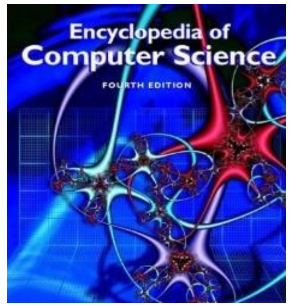
Science of Crowdsourcing

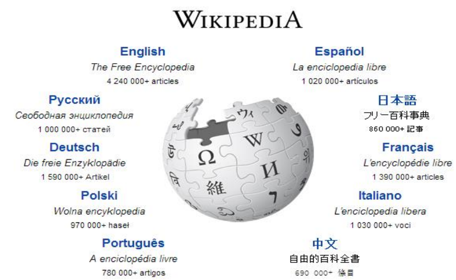
- Computer Architecture based on HPU
  - **Parallel** Computing based on HPU?
  - **Cache** design, **Memory** design?
  - **Index** design for Crowdsourced Spatial Database?

- Algorithms Design based on HPU
  - **Complexity** of human algorithms?
  - Is there **NP-hard** theory based on HPU?

Conclusions

- Scientific Domain：Wikipedia vs. Encyclopedia





- A Novel Computation Paradigm in The Era of Sharing Economy

衣 食 住 行

Reading List

- **Task Assignment**
  - **Y. Tong,** J. She, B. Ding, L. Chen, T. Wo, and K. Xu. Online Minimum Matching in Real-Time Spatial Data: Experiments and Analysis. **In VLDB 2016. (CCF A)**
  - **Y. Tong,** J. She, B. Ding, L. Wang, and L. Chen. Online Mobile Micro-Task Allocation in Spatial Crowdsourcing**. In ICDE 2016. (CCF A)**
  - J. She, **Y. Tong**, L. Chen, and C. Cao. Conflict-Aware Event-Participant Arrangement and its Variant for Online Setting. **In TKDE 2016. (CCF A)**
  - J. She, **Y. Tong,** and L. Chen. Utility-Aware Event-Participant Planning. **In SIGMOD 2015. (CCF A)**
  - J. She, **Y. Tong**, L. Chen, and C. Cao. Conflict-Aware Event-Participant Arrangement. **In ICDE 2015. (CCF A)**
- **Quality Control**
  - **Y. Tong,** C. Cao, and L. Chen. TCS: Efficient Topic Discovery over Crowd-oriented Service Data. **In SIGKDD 2014. (CCF A)**
  - **Y. Tong**, C. Cao, C. Zhang, Y. Li, and L. Chen. CrowdCleaner: Data Cleaning for Multi-Version Data on the Web via Crowdsourcing. **In ICDE 2014. (CCF A)**
  - C. Zhang, **Y. Tong,** and L. Chen. Where To: Crowd-Aided Path Selection. **In VLDB 2015. (CCF A)**
  - C. Cao, **Y. Tong,** L. Chen, and H. V. Jagadish. WiseMarket: A New Paradigm for Managing Wisdom of Online Social. **In SIGKDD 2013. (CCF A)**
  - C. Cao, J. She, **Y. Tong,** and L. Chen. Whom to Ask? Jury Selection for Decision Making Tasks on Micro-blog Services. **In VLDB 2012. (CCF A)**

致谢


dank u
ju faleminderit
Tack
Asante
谢谢
Tak
mulţumesc
kiitos
Salamat!
Gracias
Terima kasih
Aliquam
Merci
Dankie
Obrigado
ありがとう
köszönöm
grazie
Aliquam
Go raibh maith agat
děkuji
Thank you
gam