GeoCrowd: Maximum Task Scheduling

Dingxiong Deng, Ugur Demiryurek, Cyrus Shahabi
Integrated Media Systems Center
University of Southern California

iCampus/MediaQ✓ iWatch IT✓

Introduction

- **Introduction:** the significant growth in the number of smart phone users, the phone's hardware and software features, and the bandwidth enables the platform of spatial crowdsourcing

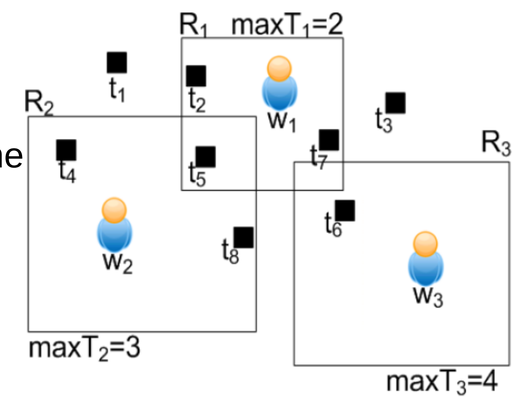
- **Spatial Crowdsourcing**
  + Crowdsourcing a set of spatial tasks to a set of workers
  + Spatial task is related to a location
  + Workers performing the spatial tasks by physically travelling to those locations

- **Spatial Task Publishing Mode**
  + Worker selected tasks (WST)
  + Server assigned tasks (SAT)

Previous Model

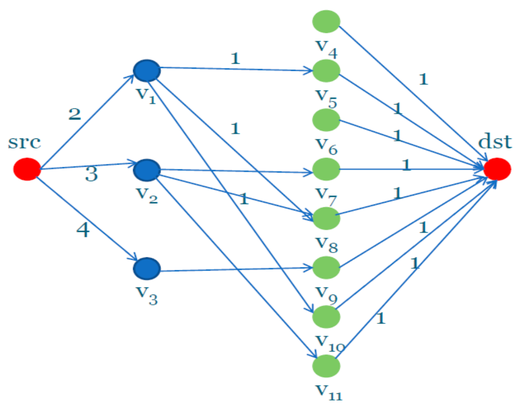
- **Server Assigned Mode**
  + Server assigns to each worker the tasks close to him
  + Server can globally optimize the number of assigned tasks

- **Maximum Task Assignment**
  + Maximize the number of tasks assigned to the workers
  + Satisfying the workers' constraints



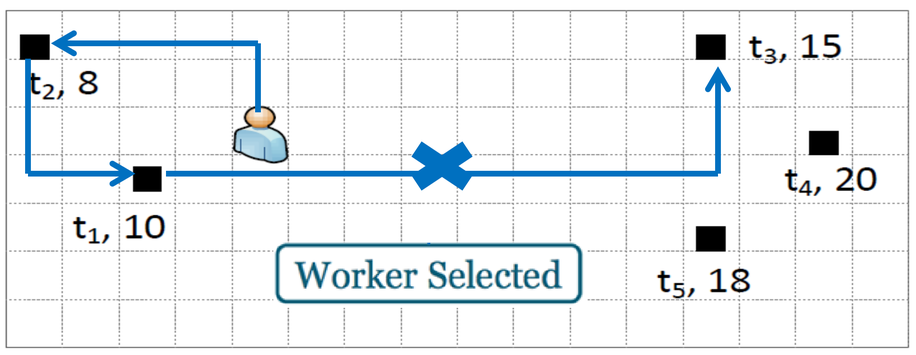
Example of W_i and T_i

- **Maximum Flow Problem**
  + Greedy (GR) Strategy
  + Least Location Entropy Priority (LLEP) Strategy
  +Nearest Neighbor (NNP) Strategy



Motivation

- **Limitation of Previous Model**
  + It is assumed that one worker can finish all the tasks assigned to him
  + It doesn't consider the travel cost from one location to another
  + It doesn't consider tasks' expiration time (deadline) , i.e. if tasks cannot be started before their deadlines, it will expire

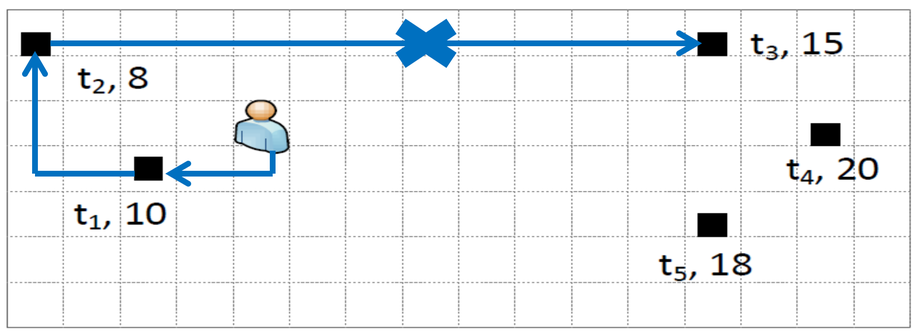


- **GeoCrowd: Worker Selected Mode**
  + Considering the above two constraints
  + A worker should have a scheduling about how to finish these tasks

Problem Definition

- **Task *t* and Worker *w***
  + Task *t* and worker *w* have location information *l(t)* and *l(w)*
  + Task *t* has expiration time *d(t)*

- **Task Sequence *R***
  + Given a worker *w* and a set of tasks $T = \{t_1, t_2, \ldots, t_n\}$
  + Task sequence *R* is a subset of tasks: $R = <t_{p1}, t_{p2}, \ldots, t_{pr}>$
  + E.g. $R = <t_1, t_2, t_3, t_4, t_5>$

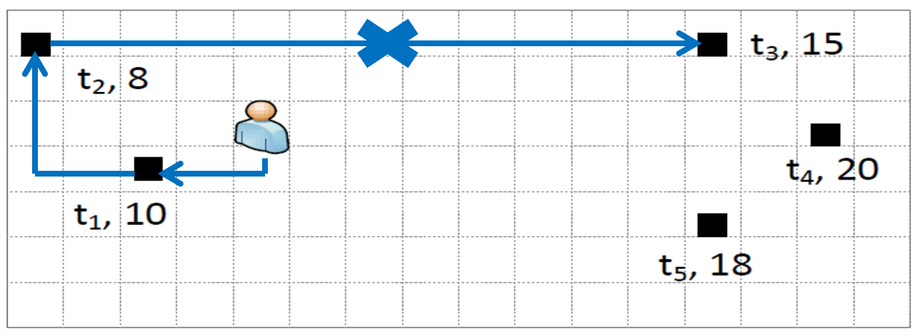


- **Valid Number α_R for a task sequence R**
  + α_R is the number of tasks which can be finished on time
  + E.g. $\alpha(< t_1, t_2, t_3, t_4, t_5 >) = 2$

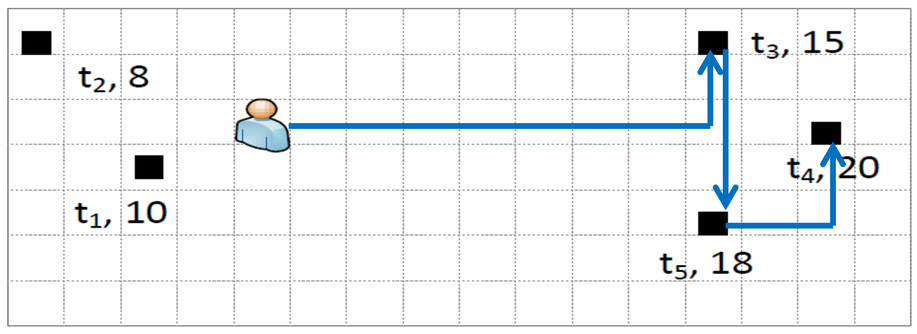
- **Maximum Task Scheduling (MTS)**
  + Given a worker *w* and a set of tasks *T*
  + MTS: find a valid task sequence *R* which has the maximum valid number

Case Study and Problem Complexity

- **Greedy Choice**
  + Greedy chose the task nearest to the worker or with least expiration time?



  + Optimal solution



- **Problem Complexity**
  + MTS problem can be proved as NP-hard by reduction from Travel Salesmen Problem (TSP)
  + Approximation Algorithm

Conclusion and Future Work

- Develop exact and approximate algorithm for MST problem
- Extend MTS problem from WST TO SAT mode