Online Task Assignment in Spatial Crowdsourcing

Mohammad Asghari, Liyue Fan, Cyrus Shahabi
Integrated Media Systems Center
University of Southern California

Introduction

- **Crowdsourcing:**
  + Outsourcing a set of tasks to a set of workers.

- **Spatial Crowdsoursing:**
  + Crowdsourcing a set of *spatial tasks* to a set of workers.
  + Workers can perform the spatial task by physically being at the location of the task.
    E.g., Taking a picture of Tommy Trojan before the UCLA game.

Task Assignment in Spatial Crowdsourcing

- ***Dynamism*** in multiple dimensions:
  + New *tasks* and *workers* can become available without the server having a priori knowledge of their ***release time***.
  + The ***duration*** within which a task (worker) is available, is known only after its release.
  + The ***location*** of the task (worker) is known only after its release.

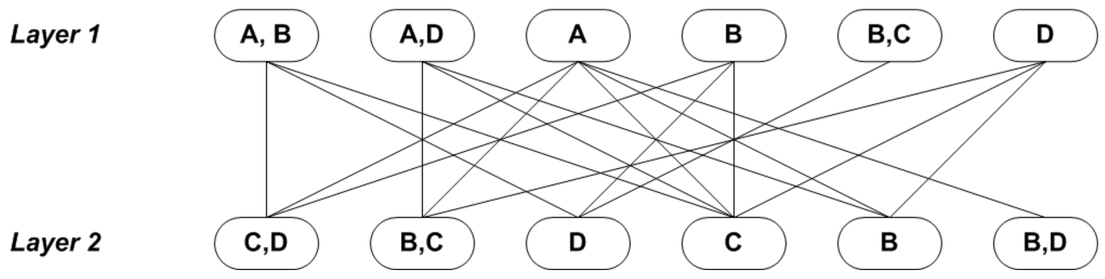
- ***Matching*** and ***Scheduling*** simultaneously.

***Task Assignment in Spatial Crowdsourcing (TASC)* finds a matching between the tasks and the workers such that each worker will be able to complete all the tasks assigned to it within the time constraints of both tasks and workers.**

Clairvoyant Algorithms

- What if there exists a Clairvoyant which has global knowledge?
  + Using a straight forward variation of the *Hamiltonian Path Problem*, we show the TASC problem is ***NP-Complete***.

- **Optimal Clairvoyant Algorithm:**
  1- For each worker, find all potential task subsets (PTS) that the worker can perform.
  2- Generate a graph by connecting each PTS of worker *i*, to every PTS of worker *j*, where the intersection of the two PTSs is the empty set.



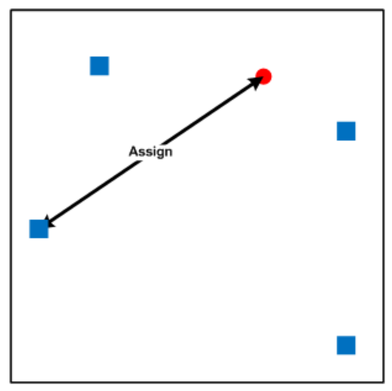
  3- Find the **maximum clique** in this graph.

- **Approximate Clairvoyant Algorithm:**
  As long as you have workers:
  1- Find the *worker* with the *bestPTS* using the remaining tasks.
  2- Assign tasks in *bestPTS* to *worker.*
  3- Remove the tasks in *bestPTS* from the list of tasks.
  4- Remove the *worker* from list of available workers.

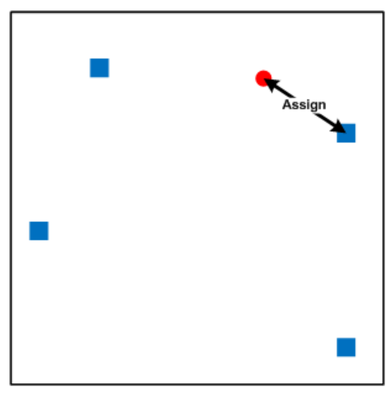
Online Algorithms

In Online Algorithms, upon arrival of a new task, the SC server makes an ir*revocable* decision whether to assign the task to a specific worker or not. Different heuristics can be assumed in order to make such decision:

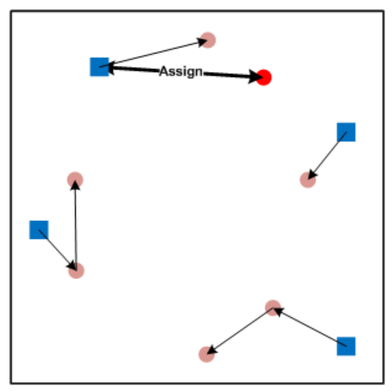
- **Random:**
  + Among workers available to perform the task, randomly select one and assign the task to it.



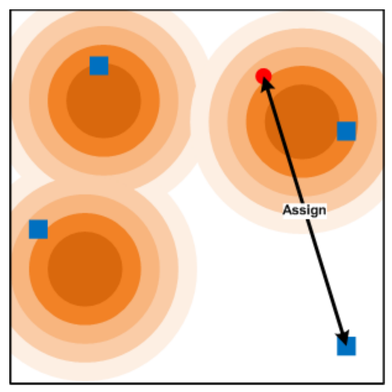
- **Nearest Neighbor:**
  + Assign the task to the nearest worker that will be able to perform the task on time.



- **Best Insertion:**
  + Assign the task to the worker who can schedule the task with minimum additional cost.



- **Best Distribution:**
  + Assuming we have the spatial distribution of tasks, assign the task to worker, which will cause the spatial distribution of workers get as close as possible to the distribution of tasks.



Related Research

- The Assignment Problem
- Task Scheduling
- The Vehicle Routing Problem
- Online Algorithms and Competitive Analysis

Conclusion and Future Work

- Test the performance of different heuristic approaches with each other and also with Clairvoyant algorithms under different scenarios.

- Propose an online algorithm with a theoretical bound with respect to the optimal offline (clairvoyant) solution.