Cost-Efficient Worker Trajectory Planning Optimization in Spatial Crowdsourcing Platforms

Ning Wang and Jie Wu
# Research Background

- **Crowdsourcing and Spatial Crowdsourcing**
  - Crowdsourcing: organizing the crowd (workers) to do tasks which are hard for machines but easy for human.
  - Spatial crowdsourcing: Organizing the crowd (mobile workers) to do spatial tasks by physically moving to other locations.
Research Background

Tasks

- General Spatial Task
  - Inventory identification
  - Placement checking
  - Data collection
  - ...

- Specific spatial task
  - Taxi calling service
  - Food delivery service
  - ...

"Spatial Crowdsourcing: Challenges, Techniques, and Applications", in Proceedings of the 43rd International Conference on Very Large Databases (VLDB 2017), Munich, Germany
Research Background

- Management Mode
  - Worker Selected Tasks (WST)
    - workers actively select tasks
  - Server Assigned Tasks (SAT)
    - workers passively wait for the platform to assign tasks

Worker Selected Tasks

Server Assigned Tasks
Task Assignment: Challenges

- **Quality-control**
  - Different sensors (sampling frequency, reading-accuracy)
  - Different behaviors (e.g., following the instruction strictly or careless)

- **Crowdsourcing Cost**
  - Workers have to go the crowdsourcing locations from their current locations.
  - Different workers have different movement distances.
Network Model

- Multiple workers and crowdsourcing locations
  - Each worker has a certain quality for finishing crowdsourcing tasks.
  - The cost of a worker is proportional to the movement distance, e.g., ridesharing.
  - Each recruited worker generates a round crowdsourcing tour.
Cost-efficient Worker Recruitment Problem

- How to recruit a set of proper workers?
  - Maximize the worker recruitment efficiency
    - different crowdsourcing qualities for different workers
    - different crowdsourcing costs for different workers

  \[ \text{System efficiency} = \frac{\sum \text{quality}}{\sum \text{cost}} \]

- Coverage Constraint
  - All the crowdsourcing locations should be covered/reached, e.g., traffic/environment monitoring, route navigation, etc.

- NP-complete in general scenario
  - Reduce to the TSP problem
Cost-efficient Worker Recruitment Problem

- A motivation example
  - Three algorithms:
    - Nearest: each location is assigned to the closest worker
    - Min-Distance: overall crowdsourcing distance is minimized
    - Max-Quality: each location is assigned to the worker with the highest quality

<table>
<thead>
<tr>
<th>Schedule</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>Efficiency Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest</td>
<td>${l_1}$</td>
<td>${l_2, l_3}$</td>
<td>$\frac{3+2}{5+4} = 0.56$</td>
</tr>
<tr>
<td>Min dist.</td>
<td>${}$</td>
<td>${l_1, l_2, l_3}$</td>
<td>$\frac{4.5}{8} = 0.56$</td>
</tr>
<tr>
<td>Max quality</td>
<td>${l_1, l_2, l_3}$</td>
<td>${}$</td>
<td>$\frac{6}{10} = 0.60$</td>
</tr>
<tr>
<td>Optimal</td>
<td>${l_1, l_2}$</td>
<td>${l_3}$</td>
<td>$\frac{1.5 + 4}{2 + 7} = 0.61$</td>
</tr>
</tbody>
</table>
Proposed Problem in 1-D Scenario

- All workers and tasks can be reached via a line, e.g., people/vehicles in highway or main street.

- An example
  - two workers and three crowdsourcing locations
Proposed Solution: Dynamic Programming

Algorithm

- Sort the worker locations and crowdsourcing location separately from one side to another side, e.g., from left to right.
- Define $\text{opt}[i,j]$ as the maximum ratio between first $i$ workers with first $j$ crowdsourcing locations.
  
  The $\text{opt}[i,j].c$ and $\text{opt}[i,j].q$ are the corresponding total tour(s) length and the total quality.

$$\text{opt}[i,j] = \max \begin{cases} \text{opt}[i-1,j], & \text{if } i > 1, \\ \frac{\text{opt}[i',j',q]}{\text{opt}[i',j',c] + 2(\max\{l_{i'},l_{j'}\} - \min\{l_i,l_j\})} & \forall i',j',i' < i,j' < j. \end{cases}$$
Proposed Solution: Dynamic Programming

❖ A toy example

➢ Dynamic programming (An illustration example: \( q_1 = 0.5 \) and \( q_2 = 1 \))

Calculate \( \text{opt}[2,3] \)

\[
\text{opt}[2,3] = \max \left\{ \frac{\text{opt}[1,0].q + 3 + 1}{\text{opt}[1,0].c + 7 + 2}, \frac{\text{opt}[1,1].q + 2 + 1}{\text{opt}[1,1].c + 4 + 2}, \frac{\text{opt}[1,2].q + 1 + 1}{\text{opt}[1,2].c + 1 + 2}, \text{opt}[1,3] \right\}
\]

\( w_2: \{l_1, l_2, l_3\} \)
\( w_1: \{l_1\} \)
\( w_1: \{l_1, l_2\} \)
\( w_1: \{l_1, l_2, l_3\} \)
\( w_2: \{l_2, l_3\} \)
\( w_2: \{l_3\} \)
Proposed Problem in 2-D Scenario

- Homogenous 2-D scenario (all workers have the same quality)
  - Objective: minimize the overall tour(s) length

- A simple nearest assignment solution
  - Voronoi graph partition

![Diagrams showing nearest and optimal assignment with worker locations and tour paths.](image-url)
Proposed Problem in 2-D Scenario

- Homogenous 2-D scenario
  - Performance Analysis: to minimize the total tour length, the nearest assignment can be as bad as $n$ times of the optimal solution, where $n$ is the total number of workers in the network.
  - an extreme example

![Diagram of Nearest and Optimal Assignment]
Proposed Solution in Homogenous 2-D scenario

- A Minimum-Spanning Tree (MST) based approach
  - Transfer the network into a graph where links are shortest distance between them.
  - Add a dummy node and it has links (zero-weight) with all workers
Proposed Solution in Homogenous 2-D scenario

- A Minimum-Spanning Tree (MST) based approach
  - Find the MST in the new graph
  - Got a spanning forest by removing the dummy nodes and the corresponding link
  - Find the best visiting tour for each selected workers based on the generated spanning tree(s)
Proposed Solution: Analysis

- **Homogenous 2-D scenario**
  - MST can be calculated optimally based on the matroid theory.
  - The MST to the shortest tour transfer has an approximation ratio of 1.5 through greedy algorithm in the metric space.
  - The best shortest tour algorithm achieves an approximation of $1 + \epsilon$ through Fully Polynomial-time approximation scheme (FPTAS) in the Euclidean space.

- **Heterogeneous 2-D scenario**
  - Apply the same solution, further bounded by the maximum quality ratio between workers in the network
    - further optimization is our future work
Performance Evaluation

- Uber pick-up trace from the NYC
  - April 2014, which has 564,516 records.
  - Worker and crowdsourcing locations are randomly generated.
  - 7 different worker qualities

![Trace visualization](image)

Trace visualization  Broadway, Manhattan.
Performance Evaluation

- Time complexity (logarithmic axis)
  - The proposed approaches have similar running-time in different scales
Performance Evaluation

- Effectiveness (1-D scenario)
  - DP: Dynamic Programming, NA: Nearest Assignment, ST: Shortest Tour(s), and MQ: Max-Quality

![Graphs showing effectiveness vs task and worker number]
Performance Evaluation

- 2-D scenario
  - MST: proposed approach, NA: Nearest Assignment, and MQ: Max-Quality
Summary

- Work recruitment problem in spatial crowdsourcing is still not well-solved by considering heterogeneous worker qualities.

- We proposed the concept of the System efficiency and proposed solutions in 1-D and 2-D scenario.
  - Optimal solution in 1-D scenario
  - Approximation solution in 2-D scenario

- We demonstrated proposed approaches in Uber NYC traces.
Thanks!

- **Contact**
  - wangn@rowan.edu