Distributed Game-Theoretical Route Navigation for Vehicular Crowdsensing

En Wang\textsuperscript{1}, Dongming Luan\textsuperscript{1}, Yongjian Yang\textsuperscript{1}, Zihe Wang\textsuperscript{2}, Pengmin Dong\textsuperscript{1}, Dawei Li\textsuperscript{3}, Wenbin Liu\textsuperscript{1}, and Jie Wu\textsuperscript{4}

\textsuperscript{1}Jinlin University, \textsuperscript{2}Renmin University of China, \textsuperscript{3}Montclair State University, \textsuperscript{4}Temple University
Outline

I.  Motivation and Problem

II. Challenges

III. Contributions

IV. System Model

V. Strategy

VI. Theoretical Analysis

VII. Performance Evaluation
Motivation

Mobile Crowdsensing (MCS)

- Vehicular crowdsensing
- The existing task allocation strategies:
  - A heavy computation complexity
  - Fail to satisfy the preferences of users and the system.

Mobile Crowdsensing

Traffic monitoring

Noise monitoring

Distributed task allocation with the route navigation
Problem

How to find an equilibrium state?

<table>
<thead>
<tr>
<th>Approach</th>
<th>Solution</th>
<th>Profit</th>
<th>Equilibrium</th>
</tr>
</thead>
</table>
| Maximum profit        | $u_1: r_1$
|                       | $u_2: r_3$
|                       | $u_3: r_4$
|                       | $u_1: 6/3=2$
|                       | $u_2: 6/3=2$
|                       | $u_3: 6/3=2$
|                       | 6        | No       |
| Distributed equilibrium| $u_1: r_1$
|                       | $u_2: r_3$
|                       | $u_3: r_4$
|                       | $u_1: 5$
|                       | $u_2: 6/2=3$
|                       | $u_3: 6/2=3$
|                       | 11       | Yes      |
| Centralized optimal   | $u_1: r_1$
|                       | $u_2: r_3$
|                       | $u_3: r_5$
|                       | $u_1: 5$
|                       | $u_2: 6$  
|                       | $u_3: 1$  | 12       | No          |

$u_3$ can select $r_4$ to get more profit.
Challenges

▪ How to construct a distributed model to achieve the equilibrium while guaranteeing the profit performance?

▪ How to design a unified distributed algorithm such that it could take the requirements of both the platform and users into consideration?

▪ How to guarantee a lower performance bound with respect to the centralized optimal solution?
System model

Profit of user $i$ under strategy profile $s$: $s = (s_i, s_{-i})$

$$P_i(s) = \alpha_i \cdot \sum_{k \in L_{s_i}} \frac{w_k(n_k(s))}{n_k(s)} - \beta_i \cdot d(s_i) - \gamma_i \cdot b(s_i)$$

- the cost incurred by traveling the detour distance
- the cost incurred by the congestion

User parameters: $\alpha_i, \beta_i, \gamma_i$

System parameters: $\varphi, \theta$

An illustrative example of the influence of $\varphi$ and $\theta$

User parameters:
- $\alpha_i$ (user's preference)
- $\beta_i$ (relative cost of detour)
- $\gamma_i$ (relative cost of congestion)

System parameters:
- $\varphi$ (relative cost of detour)
- $\theta$ (relative cost of congestion)

Profit function for $u_i$: $P_i(r_j) = \frac{w(r_j)}{n(r_i)} + \varphi \cdot h(r_j) + \theta \cdot c(r_j)$

<table>
<thead>
<tr>
<th>$h(r_i)$</th>
<th>$c(r_i)$</th>
<th>$r_1$</th>
<th>$r_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Achieve different purposes by adjusting the values of $\varphi$ and $\theta$.

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$\theta$</th>
<th>Solution</th>
<th>Task #</th>
<th>Detour</th>
<th>Congestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>$u_1: r_1$ $u_2: r_2$</td>
<td>2</td>
<td>0+2=2</td>
<td>3+1=4</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>$u_1: r_1$ $u_2: r_1$</td>
<td>1</td>
<td>0+0=0</td>
<td>3+3=6</td>
</tr>
<tr>
<td>0.1</td>
<td>1</td>
<td>$u_1: r_2$ $u_2: r_2$</td>
<td>1</td>
<td>2+2=4</td>
<td>1+1=2</td>
</tr>
</tbody>
</table>

Maximize task # | Minimize detour | Minimize congestion
Theoretical Analysis

- NP-hardness of The Centralized Problem

**Theorem 1.** The problem of finding the solution with the maximum total profit in a centralized manner is NP-hard.

- Nash equilibrium
  
  No user can improve the profit by altering the strategy unilaterally in a Nash equilibrium.

- Potential game
  
  ✓ Nash equilibrium existence  ✓ Finite improvement property

- Potential game proof

**Theorem 2.** The multi-user route navigation game is a weighted potential game and has a Nash equilibrium and finite improvement property.
Strategies

For user

**Initialization Phase**

1. Input $\alpha_i, \beta_i, \lambda_i$, the initial location and the destination.
2. Receive the recommended routes $R_i$.
3. Initialize $s_i(0) = r$ by randomly selecting a route $r \in R_i$.
4. Send $s_i(0)$ to the platform.
5. Receive $n_k$ for each task $k$ that is covered by $s_i(0)$.
6. Calculate the profit $P_i$.
7. Receive $d(r)$ and $b(r)$ for each route $r$ in $R_i$.
8. **repeat** for each decision slot $t$
   - Obtain $n_k$ for each task $k$ that is covered by $R_i$.
   - Compute the best route set $\Delta_i(t)$.
   - **if** $\Delta_i(t) \neq \emptyset$ **then**
     - Send the request to contend the opportunity for updating decision.
     - **if** Win the opportunity **then**
       - Update the route selection decision $s_i(t)$ by selecting a route $r \in \Delta_i(t)$.
       - Report $s_i(t)$ to the platform.
     - **else**
       - Choose the original decision $s_i(t) = s_i(t - 1)$.
   - **until** The termination message is received.
9. **repeat** for each decision slot $t$
10. Obtain $n_k$ for each task $k$ that is covered by $R_i$.
11. Compute the best route set $\Delta_i(t)$.
12. **if** $\Delta_i(t) \neq \emptyset$ **then**
13. Send the request to contend the opportunity for updating decision.
14. **if** Win the opportunity **then**
15. Update the route selection decision $s_i(t)$ by selecting a route $r \in \Delta_i(t)$.
16. Report $s_i(t)$ to the platform.
17. **else**
18. Choose the original decision $s_i(t) = s_i(t - 1)$.

For platform

**Algorithm 2 Information Update Algorithm for the platform.**

1. Send the recommended route set $R_i$ to the user $i \in U$.
2. Receive $s_i(0)$ from each user $i \in U$.
3. Calculate $n_k$ for each task $k \in L$.
4. Send $n_k, d(r)$ and $b(r)$ to the corresponding user.
5. **repeat**
6. Receive the request from the users and let $U'$ denote the set of users that send the request.
7. **if** $U' \neq \emptyset$ **then**
8. Select a set of users $\mu$ by SUU or PUU algorithm.
9. Inform the users in $\mu$ to update the decisions.
10. Receive $s_i(t)$ from user $i \in \mu$ and update $n_k$ for each task $k \in L$.
11. **until** No request is received from the user.
12. Send the termination message to all users.
Performance Evaluation

- Convergence for Nash equilibrium

![Graphs showing profit vs. decision slot for Shanghai, Roma, and Epfl.](image)

Figure 3: User profit vs. decision slot.

![Graphs showing potential function value and total profit vs. decision slot for Shanghai, Roma, and Epfl.](image)

Figure 6: Potential function and total profit vs. decision slot.
Performance Evaluation

- **Coverage and reward**

![Bar charts showing coverage and reward for different user numbers in Shanghai, Roma, and Epfl.](chart)

**Figure 8: Coverage vs. user number.**

![Bar charts showing average reward for different task numbers in Shanghai, Roma, and Epfl.](chart)

**Figure 9: Average reward vs. task number.**
Performance Evaluation

- The influence of user and system parameters

![Graphs showing the influence of system parameters](image)

**Figure 12: The influence of system parameters.**

**Table 5: The influence of the user parameters.**

<table>
<thead>
<tr>
<th>$\alpha_i$</th>
<th>reward</th>
<th>$\beta_i$</th>
<th>detour</th>
<th>$\gamma_i$</th>
<th>congestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>7.74</td>
<td>0.1</td>
<td>12.24</td>
<td>0.1</td>
<td>12.03</td>
</tr>
<tr>
<td>0.2</td>
<td>7.85</td>
<td>0.2</td>
<td>10.97</td>
<td>0.2</td>
<td>10.48</td>
</tr>
<tr>
<td>0.3</td>
<td>7.94</td>
<td>0.3</td>
<td>9.88</td>
<td>0.3</td>
<td>9.52</td>
</tr>
<tr>
<td>0.4</td>
<td>7.96</td>
<td>0.4</td>
<td>9.38</td>
<td>0.4</td>
<td>8.75</td>
</tr>
<tr>
<td>0.5</td>
<td>7.98</td>
<td>0.5</td>
<td>8.84</td>
<td>0.5</td>
<td>8.48</td>
</tr>
<tr>
<td>0.6</td>
<td>8.08</td>
<td>0.6</td>
<td>8.38</td>
<td>0.6</td>
<td>8.20</td>
</tr>
<tr>
<td>0.7</td>
<td>8.10</td>
<td>0.7</td>
<td>8.07</td>
<td>0.7</td>
<td>8.05</td>
</tr>
<tr>
<td>0.8</td>
<td>8.16</td>
<td>0.8</td>
<td>7.99</td>
<td>0.8</td>
<td>7.97</td>
</tr>
</tbody>
</table>
Thanks for listening