TEMPLE UNIVERSITY

中国科学技术大学
University of Science and Technology of China

Deadline-Sensitive User Recruitment for Mobile Crowdsensing with Probabilistic Collaboration

Mingjun Xiao[a], Jie Wu[b], He Huang[c], Liusheng Huang[a] , and Chang Hu[a]

[a] University of Science and Technology of China,
[b]Temple University, [c]Soochow University

Outline

- Motivation
- Model & Problem
- Solution
- Extension
- Simulation
- Conclusion

Motivation

- **Mobile Crowdsensing**
  - A group of mobile users are coordinated to perform a large-scale sensing job over urban environments through their smartphones.

Sensing job

Results

Platform

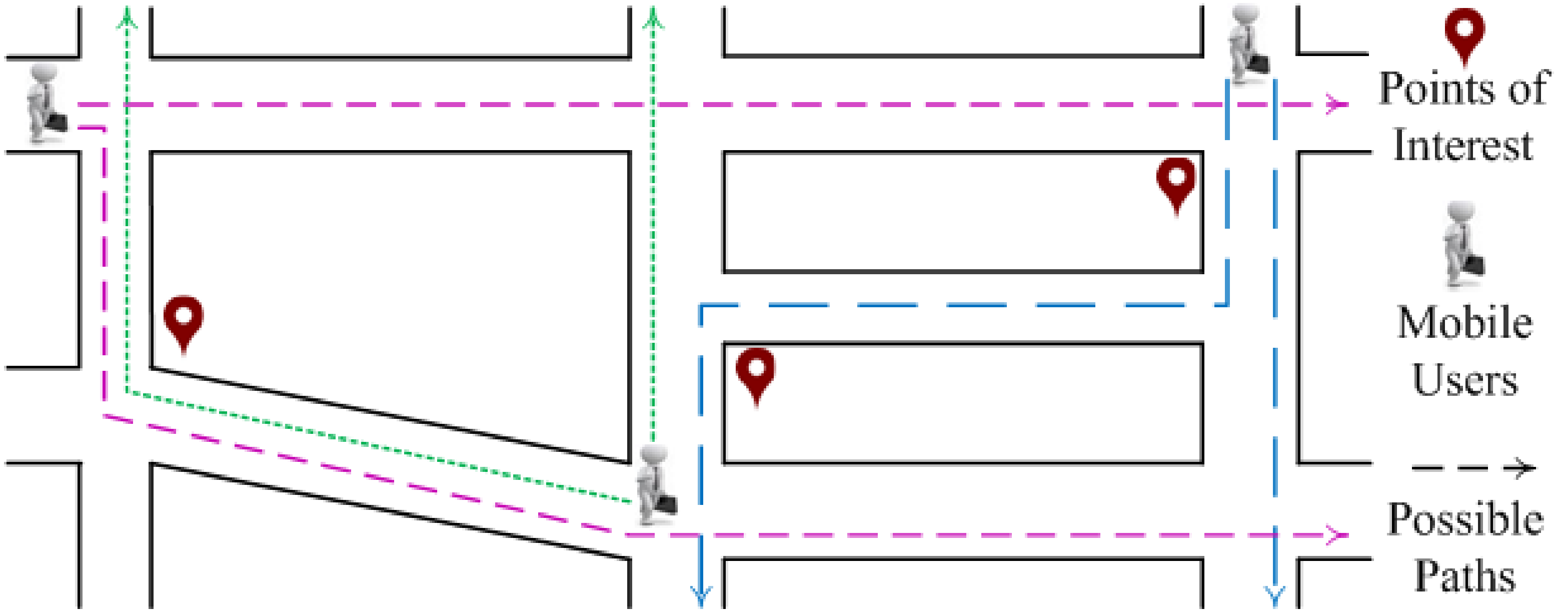
Urban area

Motivation

- **Mobile Crowdsensing**
  - Applications: urban WiFi characterization, traffic information mapping, wireless indoor localization, and so on.
  - User recruitment or task allocation is one of the most important topics
  - Existing works mainly focus on deterministic mobile crowdsensing

Motivation

- **Probabilistically collaborative crowdsensing**
  – Example:

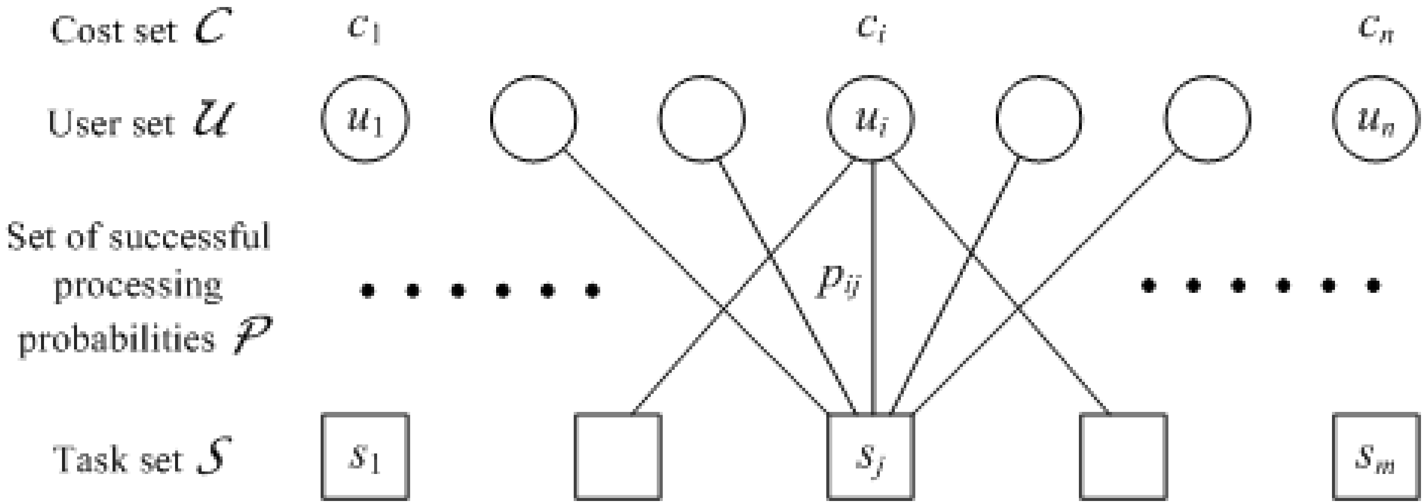


Model

- **Model**
  - Location-related sensing tasks: $S = \{s_1, \ldots, s_m\}$
  - Mobile users: $U = \{u_1, \ldots, u_n\}$
  - Time is divided into many equal-length sensing cycles: τ
  - Each user can perform one or more tasks in each sensing cycle with some probabilities: p_{ij}
  - Each user will also charge a cost from the requester as the reward for participating in crowdsensing: c_i

Model

- **Model**



Cost set $\mathcal{C}$
c_1
c_i
c_n
User set $\mathcal{U}$
u_1
u_i
u_n
Set of successful processing probabilities $\mathcal{P}$
p_{ij}
Task set $\mathcal{S}$
s_1
s_j
s_m

Problem

- **Deadline-sensitive User Recruitment (DUR)**
  - The objective is to determine which users should be recruited, so that the requester can minimize the total cost, while ensuring that the expected completion time of the crowdsensing is no larger than a given deadline T.

Problem

- **Problem Formalization**
  - Joint processing probability ρ_j^{Φ}

$$\rho_j^{\Phi} = 1 - \prod_{u_j \in \Phi} \left(1 - p_{ij}\right)$$

  - The DUR problem

$$Min \; : \; C(\Phi) = \sum_{u_i \in \Phi} c_i$$

$$s.t. \; : \; \Phi \subseteq U$$

$$\frac{\tau}{\rho_j^{\Phi}} \leq \mathrm{T}, \quad 1 \leq j \leq m$$

Solution

- **Problem Hardness Analysis**
  - **Theorem 1**: The DUR problem is NP-hard
- **Utility function $f(\Phi)$**

$$f(\Phi) = \theta \sum_{j=1}^{m} \min\{\rho_j^{\Phi}, \frac{\tau}{T}\},\ \theta = \max\{\theta_1, \theta_2\},\ \theta_1 = \frac{T \sum_{i=1}^{n} c_i}{m\tau}$$

$$\theta_2 = \max\left\{ \frac{c_i \middle| 1 \le i \le n}{\frac{\tau}{T} - \rho_j^{\Phi} \middle| 1 \le j \le m, \rho_j^{\Phi} < \frac{\tau}{T}, \Phi \subset U} \right\}$$

Solution

- **Problem Re-formalization**
  - **Theorem 2:** 1) $f(\emptyset)=0$; 2) $f(\Phi)$ is an increasing function.
  - **Theorem 4:** $f(\Phi)$ is a submodular function.
  - **Theorem 5:** $f(\Phi)$ is a polymatroid function on 2^U.

Solution

- **Problem Re-formalization**
  - **Theorem 6:** $C(\Phi)$ is a modular function as well as a polymatroid function on 2^U.
  - **Corollary 1:** The DUR problem can be equivalently re-formalized as a Minimum Submodular Cover with Submodular Cost (MSC/SC) problem :

$$Minimize\left\{ C(\Phi) \mid f(\Phi) = f(U),\ \Phi \subseteq U \right\}$$

Solution

- **The gDur Algorithm**
  - **The greedy strategy:** the user who can improve the utility mostly with the least cost is recruited first.

Algorithm 1 The *g*DUR Algorithm

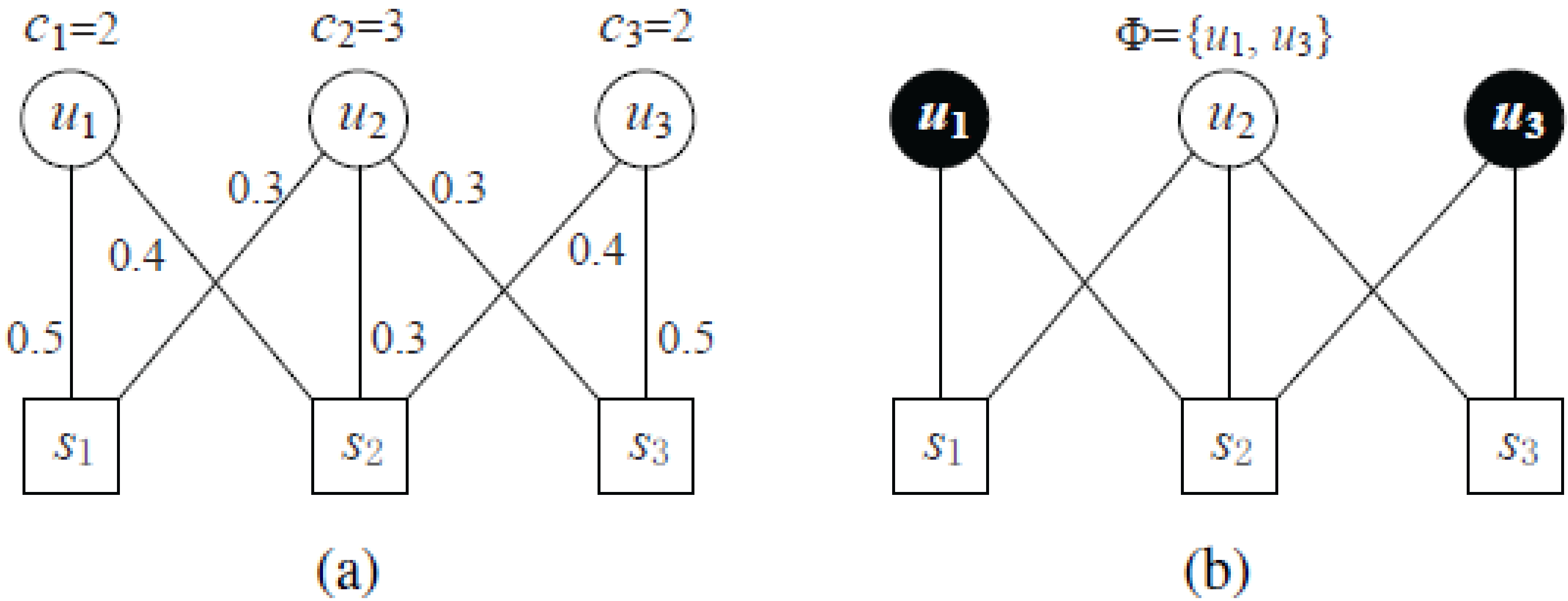
Require: $\mathcal{U}, \mathcal{S}, \mathcal{P}, \mathcal{C}, \tau, \mathcal{T}$

Ensure: Φ

1: $\Phi = \emptyset$;
2: **while** $f(\Phi) < \frac{m\tau\theta}{\mathcal{T}}$ **do**
3: Select a user $u_i \in \mathcal{U} \setminus \Phi$ to maximize $\frac{f(\Phi \cup \{u_i\}) - f(\Phi)}{c_i}$;
4: $\Phi = \Phi \cup \{u_i\}$;
5: **return** Φ;

Solution

- **An Example**



Solution

- **Correctness and Performance Analysis**
  - **Theorem 3:** The gDur algorithm 1 is correct. That is, it will produce a feasible solution of the DUR problem, as long as the problem is solvable.
  - **Theorem 8:** The proposed gDUR algorithm can achieve a $\left(1+\ln\frac{m\tau\theta}{opt\cdot T}\right)$-approximation solution, where opt is the cost of the optimal solution for the DUR problem.

Extension

- **The Extended Problem**
  - When a user performs a sensing task, there is a sensing duration *d*. The total expected duration of each task *σ* is no less than a given threshold *D*.

$$Min\ :\ \ C(\Phi)=\sum\nolimits_{u_i\in\Phi}c_i$$

$$s.t.\ \ :\ \ \Phi\subseteq U$$

$$\sigma_j^{\Phi}\geq D,\quad 1\leq j\leq m$$

$$\frac{\tau}{\rho_j^{\Phi}}\leq \mathrm{T},\quad 1\leq j\leq m$$

Extension

- **Solution**
  - **Utility function *g* (Φ)**

$$g(\Phi) = \frac{\vartheta}{mD}\sum_{j=1}^{m}\min\{\sigma_j^{\Phi}, D\},$$

$$\vartheta = \theta \text{ if } D > 0; \vartheta = 0 \text{ and } g\left(\Phi\right) = 0, \text{ if } D = 0$$

  - **Combinational utility function *h* (Φ)**

$$h(\Phi) = f(\Phi) + g(\Phi)$$

Extension

– **The dDur algorithm:** the user who can improve the combinational utility mostly with the least cost is recruited first.

Algorithm 2 The dDUR Algorithm

Require: $\mathcal{U}, \mathcal{S}, \mathcal{P}=\{p_{ij}|u_i \in \mathcal{U}, s_j \in \mathcal{S}\}, \mathcal{C}, \tau, \mathcal{T}, \mathcal{D}$

Ensure: Φ

1: $\Phi = \emptyset$;

2: **while** $h(\Phi) < \frac{m\tau\theta}{\mathcal{T}} + \vartheta$ **do**

3: Select a user $u_i \in \mathcal{U} \backslash \Phi$ to maximize $\frac{h(\Phi \cup \{u_i\}) - h(\Phi)}{c_i}$;

4: $\Phi = \Phi \cup \{u_i\}$;

5: **return** Φ;

Extension

- **Correctness and Performance Analysis**
  - **Theorem 9**: 1) h (Φ) is an increasing function with $h(\emptyset)=0$; 2) h (Φ) = $m\tau\theta/T + \vartheta$ if and only if Φ is a feasible solution of the extended DUR problem.
  - **Theorem 11**: The propsed dDUR algorithm can achieve a $\left(1+\ln\frac{m\tau\theta+\vartheta T}{opt\cdot T}\right)$-approximation solution, where *opt* is the cost of the optimal solution for the extended DUR problem.

Simulation

- **Trace**
  - Cambridge Haggle Trace Set
  - Synthetic traces
- **Settings**

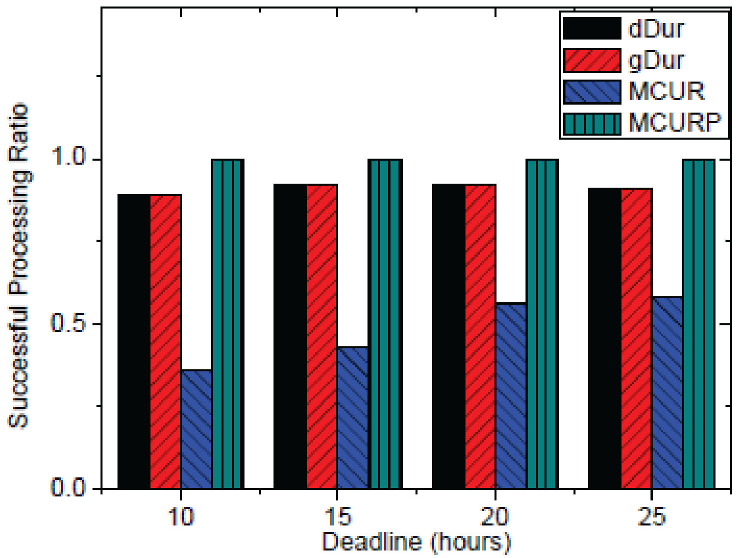
Parameter name	Default Value	Range
Number of users n	100	100-400
Number of tasks m	20	20-80
Threshold of sensing duration D	0min	0min,4min
Deadline T	10hours	{10, 15, 20, 25}
Probabilities of users P	[0,0.1]	0-0.4

Simulation

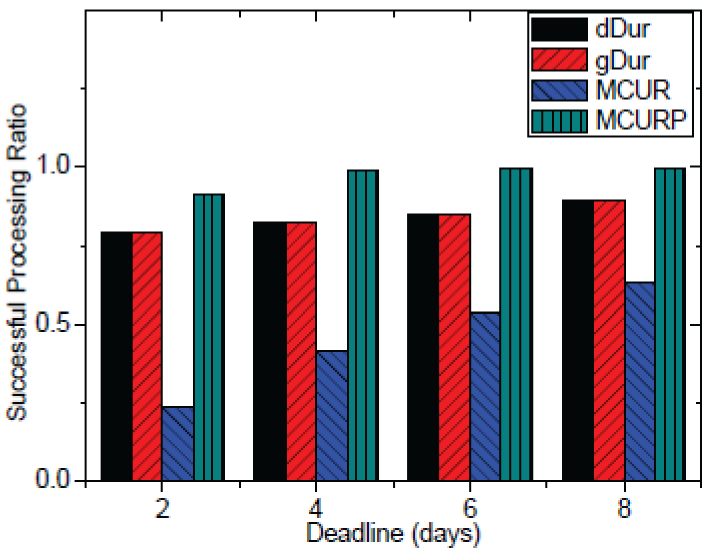
- Algorithms in comparison
  - gDUR
  - dDUR
  - Minimum Cost User Recruitment (MCUR)
  - MCUR with Probabilistic mobility (MCURP)
- Metrics
  - The total cost
  - The successful processing ratio

Simulation

- Results
  - Successful processing ratio vs. deadline



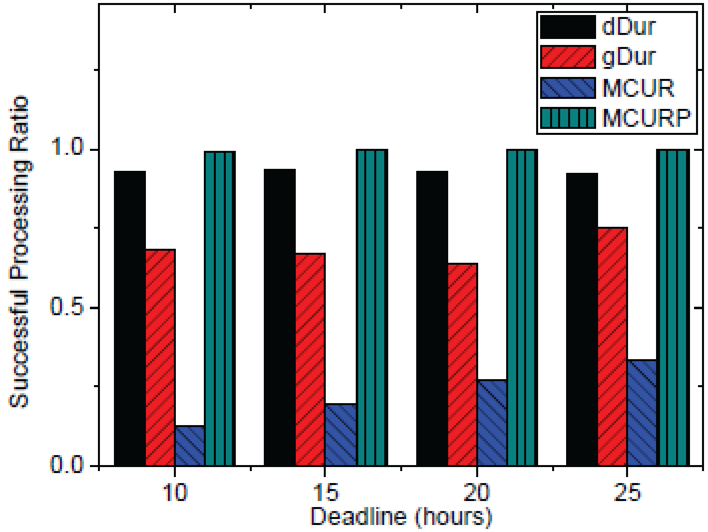
(a) $\mathcal{D}=0$ minute



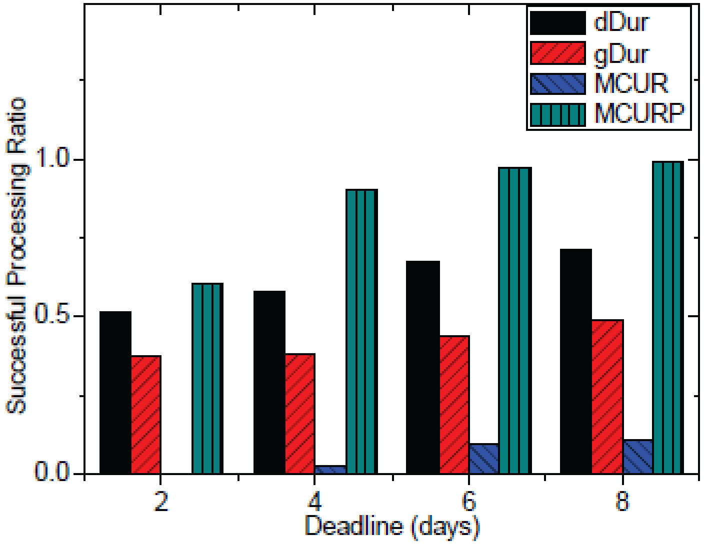
(a) $\mathcal{D}=0$ minute

Simulation

- Results
  - Successful processing ratio vs. deadline



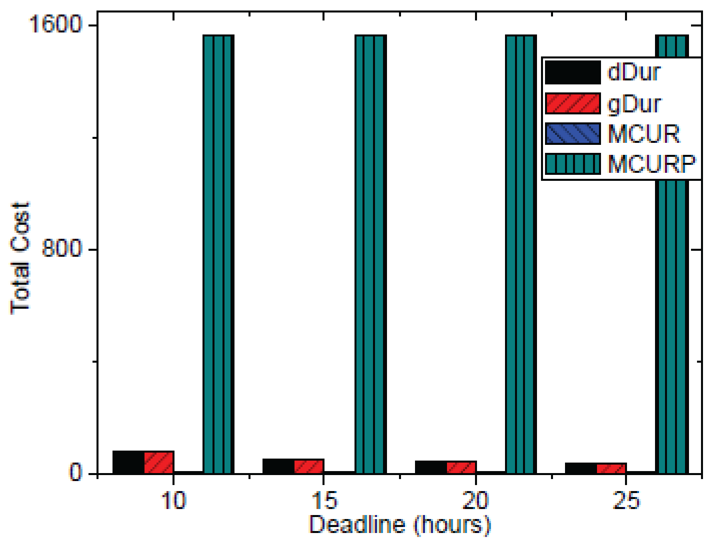
(c) $\mathcal{D}=4$ minutes



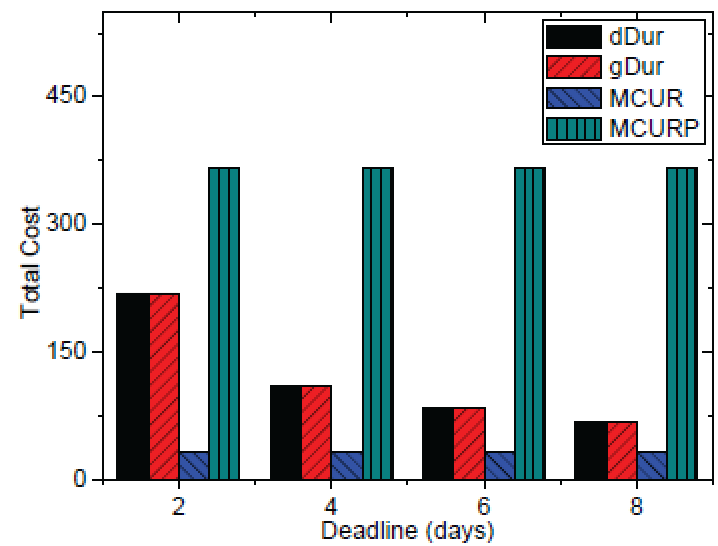
(c) $\mathcal{D}=4$ minutes

Simulation

- Results
  - Total cost vs. deadline



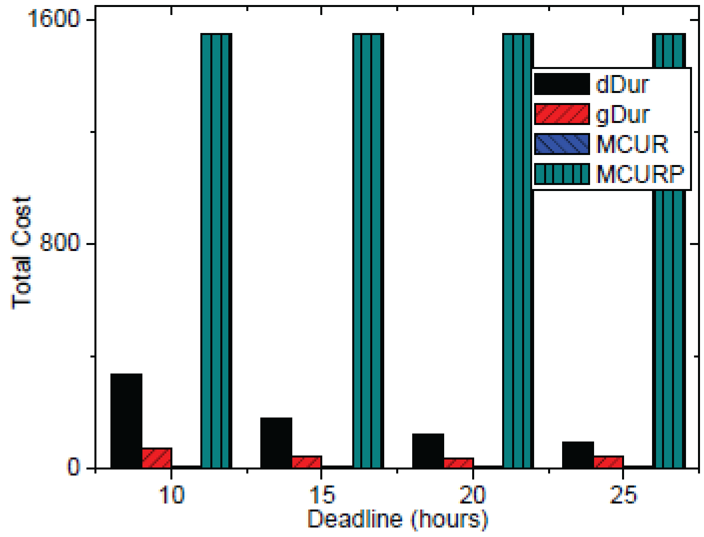
(b) $\mathcal{D}=0$ minute



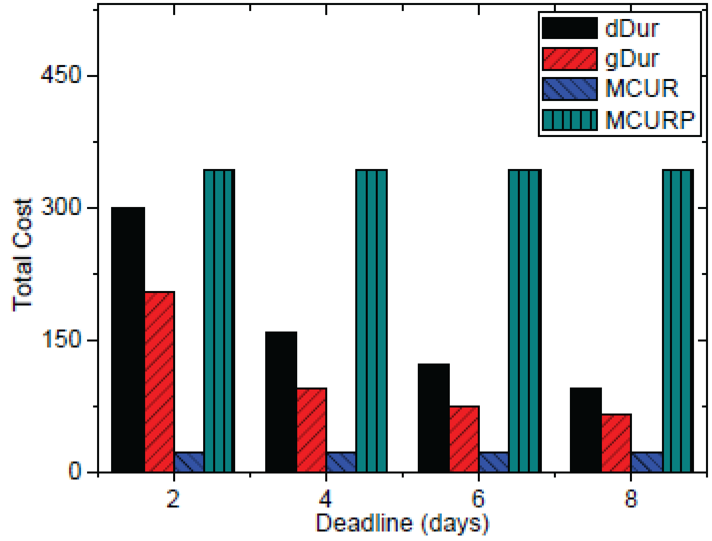
(b) $\mathcal{D}=0$ minute

Simulation

- Results
  - Total cost vs. deadline



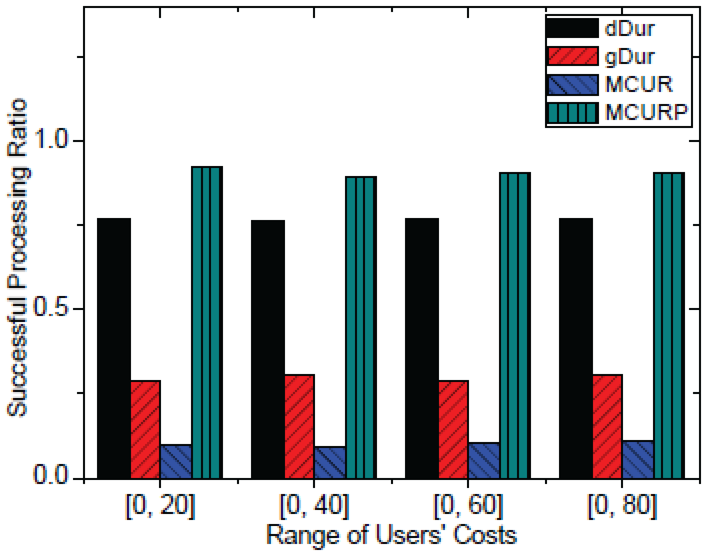
(d) $\mathcal{D}=4$ minutes



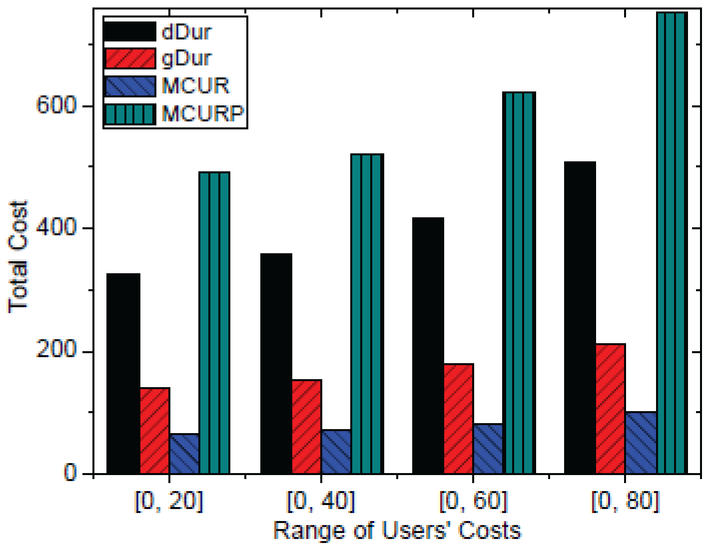
(d) $\mathcal{D}=4$ minutes

Simulation

- Results
  - Changing *costs C*



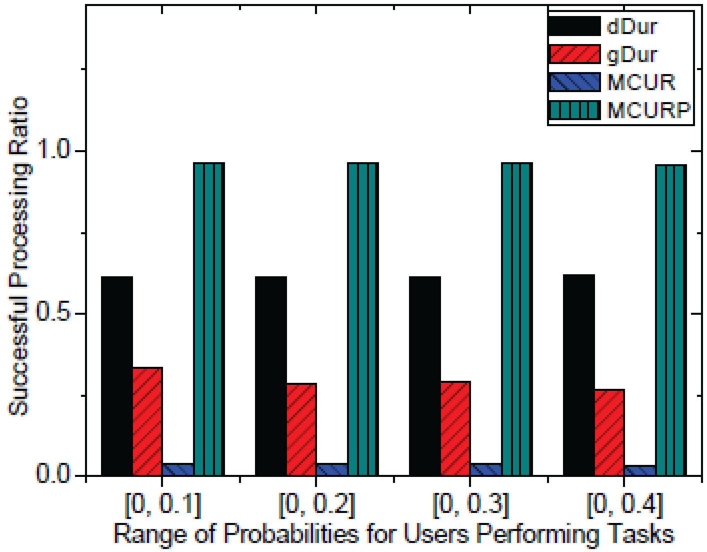
(a) Ratio vs. Costs of Users



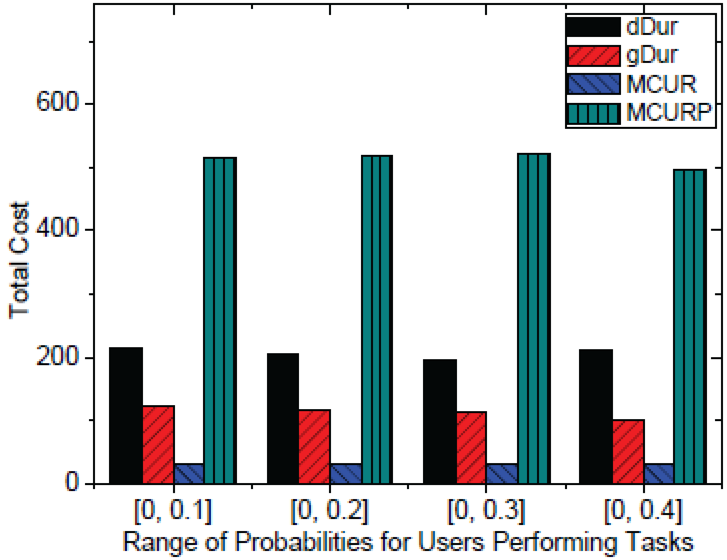
(b) Total Cost vs. Costs of Users

Simulation

- Results
  - Changing *costs P*



(c) Ratio vs. Probabilities of Users



(d) Total Cost vs. Probabilities of Users

Conclusion

- When the sensing duration is ignored, gDUR and dDUR achieve the same results.
- When the sensing duration is considered, dDUR will recruit more users than gDUR and resulting in larger total costs as well as higher successful processing ratios.
- MCUR algorithm recruits fewer users than our algorithms, however results in very low successful processing ratios.
- MCURP algorithm achieves higher successful processing ratios than our algorithms while resulting in larger total costs.
- Both gDUR and dDUR demonstrate much better integrative performances than the two compared algorithms.

Thanks!

Q&A