Detour Planning Problem on Mobile Crowdsensing Systems

Chen-Chih Liao

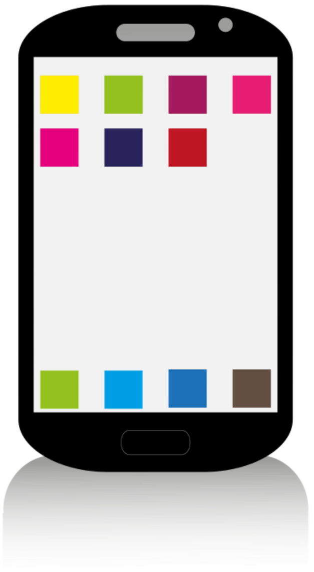
Slides: 39 pages

Outline

1. Introduction
    - Motivation
    - Crowdsensing
    - Research Problems
2. Related Work
3. Detour Planning Problem
    - Formulation
    - Solution
    - Evaluation
4. Multi-user Detour Planning Problem
    - Formulation
    - Solution
    - Evaluation
5. Conclusion and Future Work

Motivation

- Smartphone users are ubiquitous, and smartphones provide powerful computing and sensing abilities
  - Smartphones are capable to perform some tasks (e.g.: shooting photos/videos, reading sensory data, and etc.)
- Ideas to make good use of smartphones and let smartphone users contribute their effort for some rewards
- Crowdsensing



What is Crowdsensing?

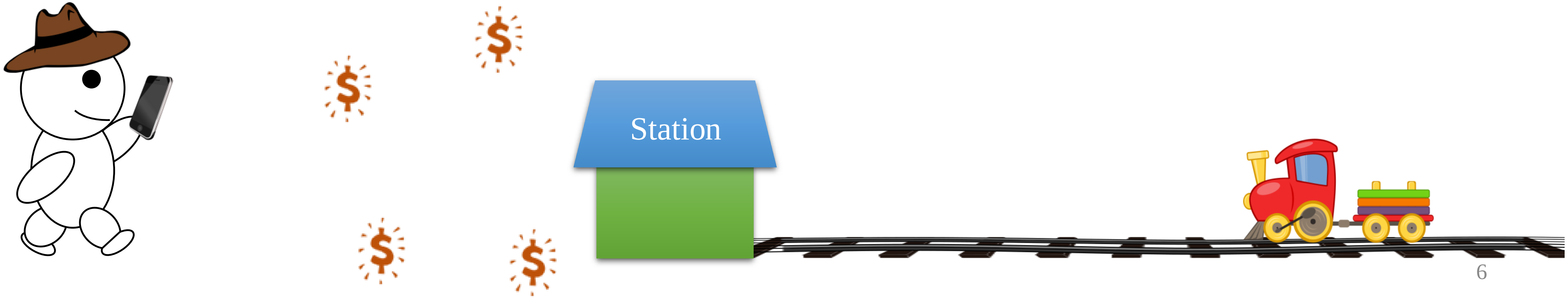
- Mobile sensing
  - Opportunistic sensing
  - Participatory sensing
- Limitation
  - Mobile sensing could not serve a large number of sensing tasks
- Crowdsensing
  - Human-in-the-loop
    - Similar to crowdsourcing

Geospatial Information Gathering

- **A new class of crowdsensing systems**
- Requesters: companies and organizations
  - Submit geospatial and temporal-dependent tasks (specific time and location)
  - Task: capturing videos/pictures or collecting sensor readings
- Workers: smartphone users
  - Report their destination and deadline
  - They wouldn't mind to take some detour routes for small rewards

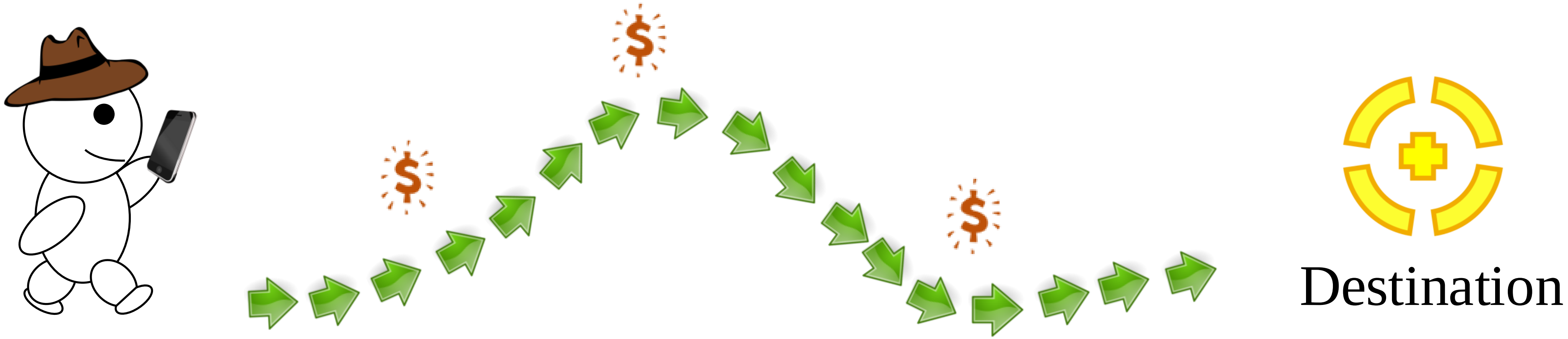
Usage Scenario #1

- When the smartphone user is at the train station, and he is waiting the train which will arrive at 1 to 2 hours
- What can he do in this free time?
  - Perform some tasks which are near the station



Usage Scenario #2

- When the smartphone user is traveling, and he expect that he will arrive at his destination in 2 hours
- What can he do in this traveling time?
  - Perform some tasks which are near the expected route

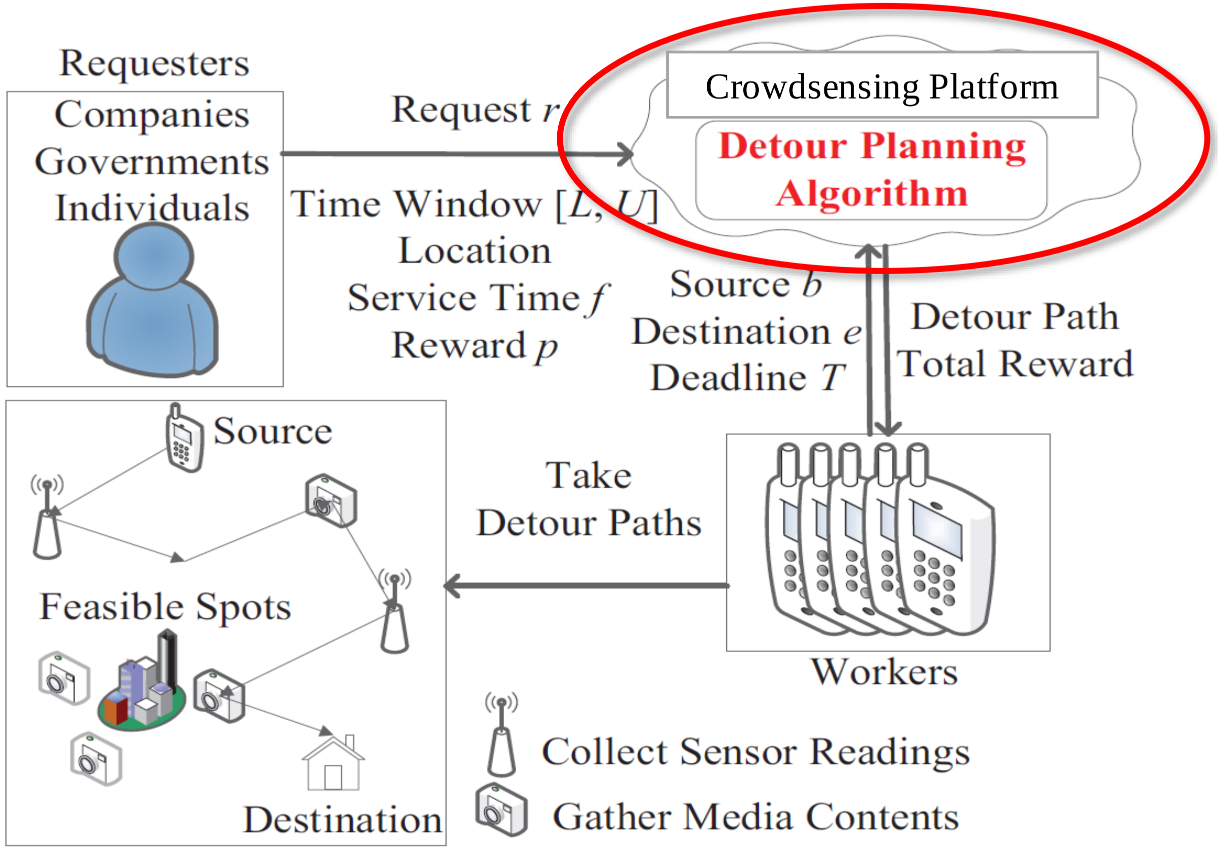


Key Research Problem: Detour Planning Problem

- Problem: How to make a good use of the abilities of ubiquitous smartphone users?
- Goal: We plan to let smartphone users well utilize their smartphones and earn the maximum rewards in their available time
- Solutions:
  - (Single-user) Detour Planning Algorithm [MoVid'13]
  - Multi-user Detour Planning Algorithm

[MoVid'13] C. Liao and C. Hsu. A detour planning algorithm in crowdsourcing systems for multimedia content gathering. In *Proc. of Workshop on Mobile Video (MoVid'13)*

System Architecture



Contribution

1. The systems produce a detour path for each new worker.
2. The systems compute the detour paths to maximize total worker profit.
3. The systems simultaneously consider multiple users to make a good use of all users.
4. The systems concern the energy consumption and sensor accuracy when assigns requests to workers.

Outline

1. Introduction
   - Motivation
   - Crowdsensing
   - Research Problems
2. Related Work
3. Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
4. Multi-user Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
5. Conclusion and Future Work

Related Work

- Crodwsourcing
  1. [36] Task matching in crowdsourcing
  2. [1] Mechanism design for spatio-temporal request satisfaction in mobile networks
  3. [3] On task assignment for real-time reliable crowdsourcing
  - ➢We consider mobile multimedia/sensing, and our solution gives optimal paths
- Crowdsensing
  1. [12] A location-based incentive mechanism for participatory sensing systems with budget constraints
  2. [5] Truthful auction for location-aware collaborative sensing in mobile crowdsourcing
  3. [23] Toward optimal allocation of location dependent tasks in crowdsensing
  - ➢We consider more time constraints of multimedia/sensing requests and workers, and we also take account of reward, traveling/energy costs, and accuracy.

Outline

1. Introduction
   - Motivation
   - Crowdsensing
   - Research Problems
2. Related Work
3. Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
4. Multi-user Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
5. Conclusion and Future Work

Problem Formulation

$$\max \sum_{i=1}^{N-1}\sum_{j=2}^{N}[p_i - \sum_{a=1}^{z_i}\sum_{b=1}^{z_j} c_{i_a,j_b} u_{i,a} u_{j,b}] x_{i,j}$$

Maximize overall profits

$$s.t. \sum_{j=2}^{N} x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1$$

Start and end points

$$\sum_{i=1}^{N-1} x_{i,k} = \sum_{j=2}^{N} x_{k,j} \leq 1, \forall k = 2, ..., N-1$$

No rep. each request

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N}(\sum_{a=1}^{z_i}\sum_{b=1}^{z_j} m_{i_a,j_b} u_{i,a} u_{j,b} + f_i) x_{i,j} \leq T_{max}$$

Arrive destination in time

$$\sum_{j=1}^{z_i} u_{i,j} \leq 1, \forall i = 1, ..., N$$

Visit one feasible spot
of each request

$$s_i + f_i + \sum_{a=1}^{z_i}\sum_{b=1}^{z_j} m_{i_a,j_b} u_{i,a} u_{j,b} - s_j \leq M(1 - x_{i,j}), \forall i, j = 1, ..., N$$

Timeline of each request

$$L_i \leq s_i, \forall i = 1, ..., N$$

Start time of each request

$$s_i + f_i \leq U_i, \forall i = 1, ..., N$$

Finish time of each request

$$x_{i,j}, u_{i,j} \in \{0, 1\}. \quad (3.9)$$

Orienteering Problem with Time Window (OPTW)

- A similar problem
  - Goal: maximize the score
  - Game: players go to specific spots, and finish the predetermined job for a score
  - Not exactly the same:
    (1) multiple feasible spots and (2) traveling cost (gas and car depreciation)
- We enhanced a dynamic programming based OPTW algorithm [RS09] for an optimal Detour Planning (DP) algorithm
  - Complexity: $O(NZ2^{NZ})$
- We propose DP Approximation (DPA) algorithm to improve the complexity time of DP by a user selected parameter ϵ [LG06]

[RS09] Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. Computers and Operations Research, 36(4):1191–1203, April 2009.
[LG06] K. Lai and M. Goemans. The knapsack problem and fully polynomial time approximation schemes (FPTAS). http://math.mit.edu/~goemans/18434S06/knapsack-katherine.pdf,

Collecting Feasible Spots

- Find 25 landmarks in Taipei (http://taipeitravel.net) and Vancouver (http://hotels.com)
- Use Flickr API to download the pictures tagged with each landmark, and retrieve the longitude/latitude
- Use hierarchical clustering algorithm to group these photos at the granularity of blocks (~100 m) ← gives us the feasible spots
- Employ Google map to compute the distance between any two feasible spots

Simulator Implementation

- We implement a trace-driven simulator in C
- It supports five algorithms
  - The proposed DP algorithm
  - Four heuristic algorithms
    - Highest-Reward (HR) ← mimic human behavior
    - Closest-Request (CR) ← mimic human behavior
    - Highest-Reward with Ontime (HROT)
    - Closest-Request with Ontime (CROT)

Simulation Design

- Parameters
  - N: number of requests: {5, 10, 15, 20, 25}
  - T: deadline: {1, 2, 4, 8, 16} (hr)
  - C: travel cost: {0, 0.06, 0.12, 0.24, 0.48} ($/km)
- Metrics
  - Total profit
  - Running-time
  - Ontime-ratio

Ontime Ratio

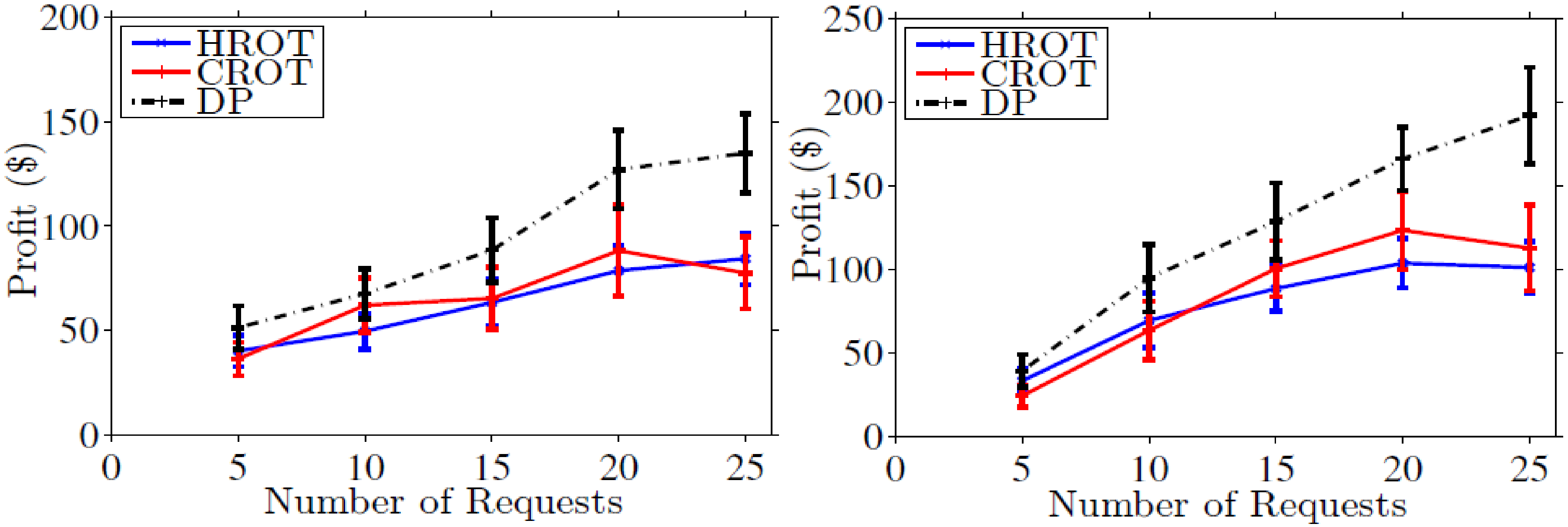
Table 1: Ontime Ratio (%) of Various Algorithms

City	Taipei			Vancouver		
Algorithm	HR	CR	DP	HR	CR	DP
Deadline $T = 1$	0	0	100	0	0	100
2	4.1	4.1	100	4.1	0	100
4	0	0	100	0	0	100
8	0	0	100	0	4.1	100
16	29.1	58.3	100	33.3	41.6	100

City	Taipei			Vancouver		
Algorithm	HR	CR	DP	HR	CR	DP
No. Requests $N = 5$	12.5	8.3	100	0	0	100
10	0	0	100	0	4.1	100
15	0	8.3	100	0	0	100
20	0	0	100	0	0	100
25	0	4.1	100	0	0	100

HR and CR (mimicking humans) → low ontime ratios!

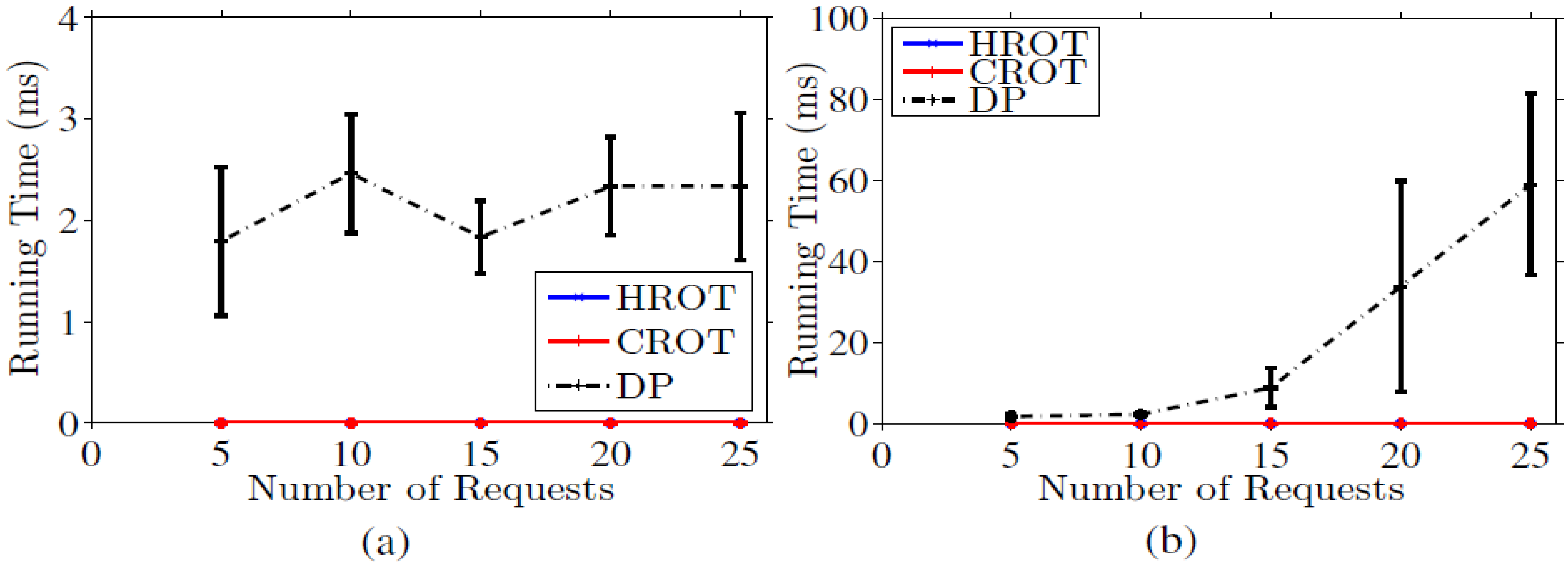
Total Profits



(a) (b)

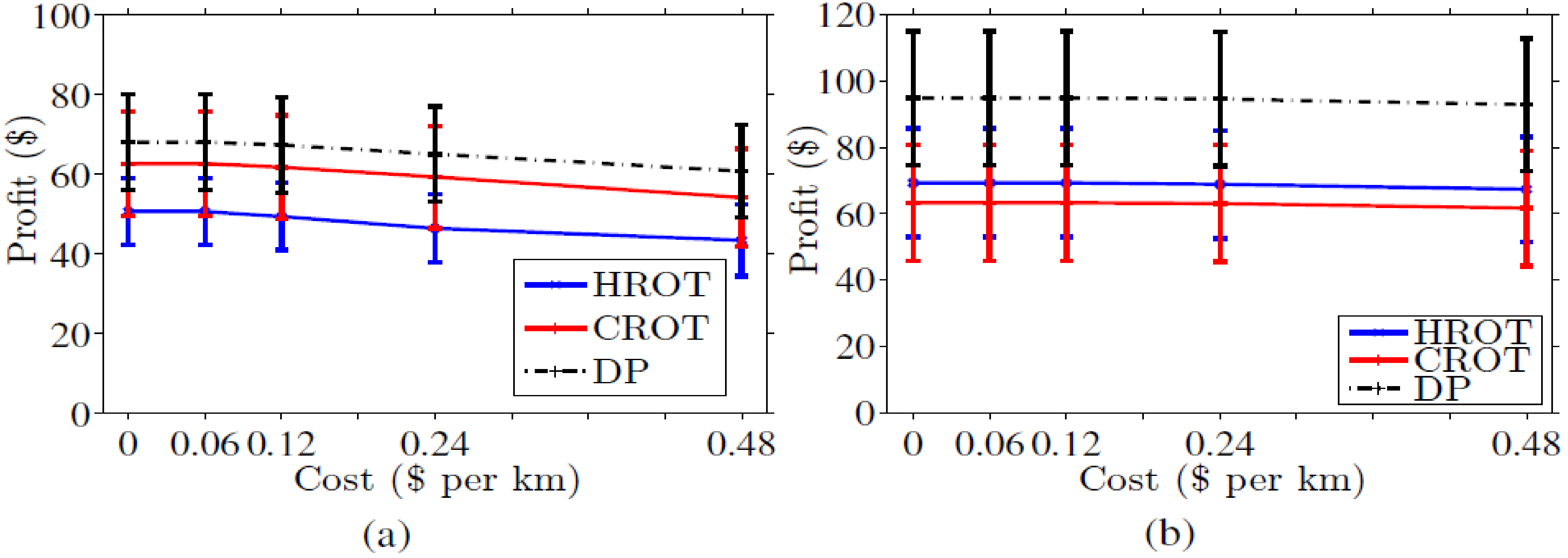
- Although HROT and CROT guarantee ontime arrival, they suffer from low profits
- Compared to HROT and CROT, DP doubles the profit with 25 requests
  - More requests → larger gap!

DP is Efficient



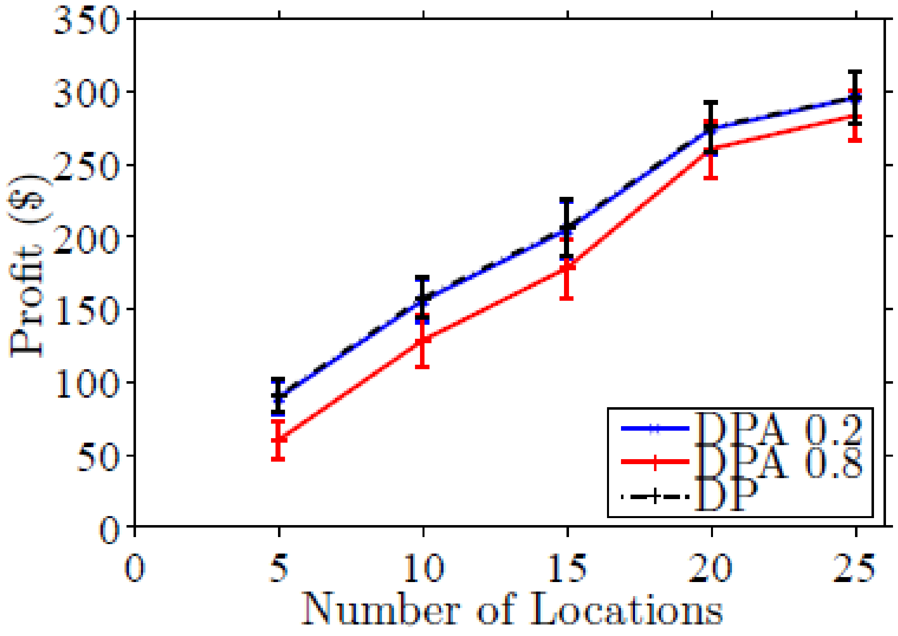
- Terminates in less than 60 ms
- Slower for Vancouver (right) ← up to total 162 feasible spots
  - Taipei (left): 49

Implication of Travel Cost

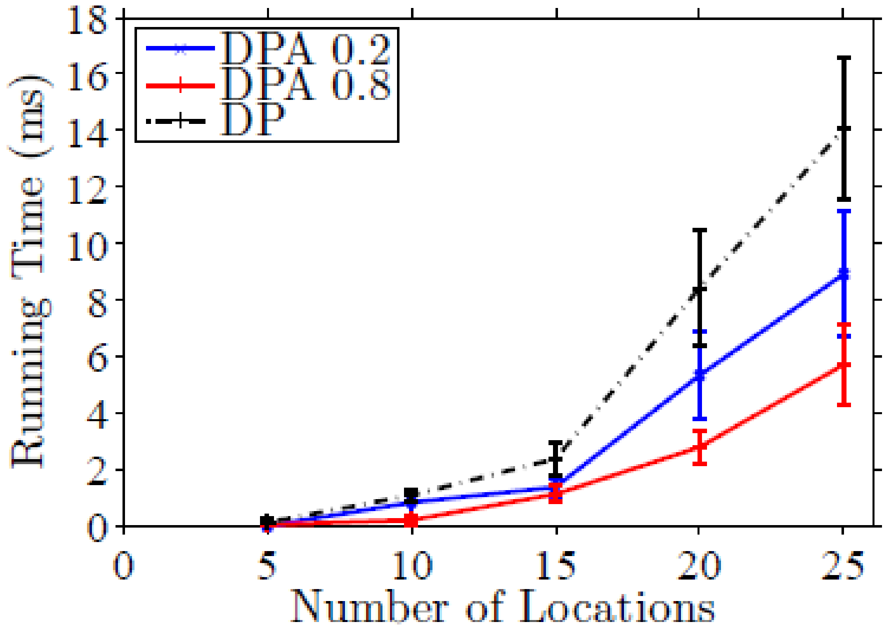


- Higher profits when per-km cost is lower

DPA Improves the Running Time with Near-optimal Results



(a)



(b)

- When $\epsilon = 0.8$, DPA achieves near-optimal profits and 3X speed ups
  - ϵ is a user selected parameter
  - higher ϵ leads to both higher approximation gap and lower complexity

Discussion

Algorithm		Contribution
DP	MDP	
V	V	1. The systems produce a detour path for each new worker.
V	V	2. The systems compute the detour paths to maximize total worker profit.
	V	3. The systems should simultaneously consider multiple users to make a good use of all users.
	V	4. The systems should concern the energy consumption and sensor accuracy when assigns requests to workers.

DP: Detour planning algorithm

MDP: Multi-user detour planning algorithm

Outline

1. Introduction
    - Motivation
    - Crowdsensing
    - Research Problems
2. Related Work
3. Detour Planning Problem
    - Formulation
    - Solution
    - Evaluation
4. Multi-user Detour Planning Problem
    - Formulation
    - Solution
    - Evaluation
5. Conclusion and Future Work

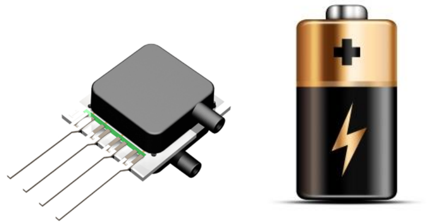
Multiple Detour Planning Problem

- The real system must simultaneously consider all workers
  - DP computes a detour path for a worker at a time
  - Workers can balance or reduce the traveling cost
- We further consider the energy cost (battery level) and the accuracy of sensory data [EMASC'14]
  - Energy model
  - Accuracy model

[EMASC'14] C. Liao, T. Hou, T. Lin, Y. Cheng, A. Erbad, C. Hsu, and N. Venkatasubramanian. Smartphone augmented infrastructure sensing for public safety and sustainability in smart cities. In *Proc. of Workshop on Emerging Multimedia Applications and Services for Smart Cities (EMASC'14)*

Models

- Energy model
  - Compute the total consumption of smartphone sensors



- Accuracy model
  - Perform how much times for achieving the accuracy of tasks
  - Decide how many workers for collecting enough result to achieve the quality of tasks

MDP Formulation

$$\max \sum_{i=1}^{N-1}\sum_{j=2}^{N}[p_i - \sum_{a=1}^{z_i}\sum_{b=1}^{z_j} c_{i_a,j_b} u_{i,a} u_{j,b}] x_{w,i,j}$$

A new dimension

Maximize overall profits

$$s.t. \sum_{w=1}^{W}\sum_{j=1,j!=o_w}^{N} x_{w,o_w,j} = \sum_{w=1}^{W}\sum_{i=1,i!=d_w}^{N} x_{w,i,d_w} = 1$$

Start and end points

$$\sum_{i=1}^{N-1} x_{w,i,k} = \sum_{j=2}^{N} x_{w,k,j} \le a(q_k), \forall w = 1, ..., W, \forall k = 2, ..., N-1$$

Assign many times to different workers for achieving the required quality

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N}(\sum_{a=1}^{z_i}\sum_{b=1}^{z_j} m_{i_a,j_b} u_{i,a} u_{j,b} + f_i) x_{w,i,j} \le T_w, \forall w = 1, ..., W$$

Arrive destination in time

$$\sum_{j=1}^{z_i} u_{i,j} \le 1, \forall i = 1, ..., N$$

No rep. feasible spots

$$s_i + f_i + \sum_{a=1}^{z_i}\sum_{b=1}^{z_j} m_{i_a,j_b} u_{i,a} u_{j,b} - s_j \le M(1 - x_{w,i,j}), \quad \forall i, j = 1, ..., N, \forall w = 1, ..., W$$

Timeline of each request

$$B_i \le s_i, \forall i = 1, ..., N$$

Start time of each request

$$s_i + f_i \le U_i, \forall i = 1, ..., N$$

Finish time of each request

$$x_{w,i,j}, u_{i,j} \in \{0, 1\}. \quad (4.9)$$

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N} \delta_i x_{w,i,j} \le g_w, \forall w = 1, ..., W$$

Satisfy the battery level

Proposed Solutions

- Multiple detour planning algorithm (MDP)
  - We design a utility function $u_{w,j} = \frac{p_j + \sum_{i \neq j}^{I} \frac{p_i}{d_{i,j}}}{d_{w,j}}$
    - Worker *w*, request *j*, profit *p*, and distance *d*
- Steps
  1. Compute all utility $u_{w,n}, \forall w = 1 \sim W, n = 1 \sim N$
  2. Choose the maximal utility $u_{i,j}$
  3. If it satisfies constraints, request j is assigned to worker i
  4. If Idle workers and requests still exist, go back to step 1. Or go to step 5
  5. Return all detour paths

Collecting real trace data

- Find 5700 posts from PTT in 10 days (4/11~4/20, 2014)
  - Contents include title, IP, and posted time
- Transfer IPs to locations
  - Filter out the IPs which are not in Taiwan by IPInfoDB
  - Hire three servers to ping the IP
    - Check network latency ⇨ Estimate the distance
  - Partition Taiwan into 1 km^2 grids
    - Compute the Mean-Square-Error (MSE) of each grid's and the IP's distances to servers
  - The precise locations are then randomly assigned within the grid

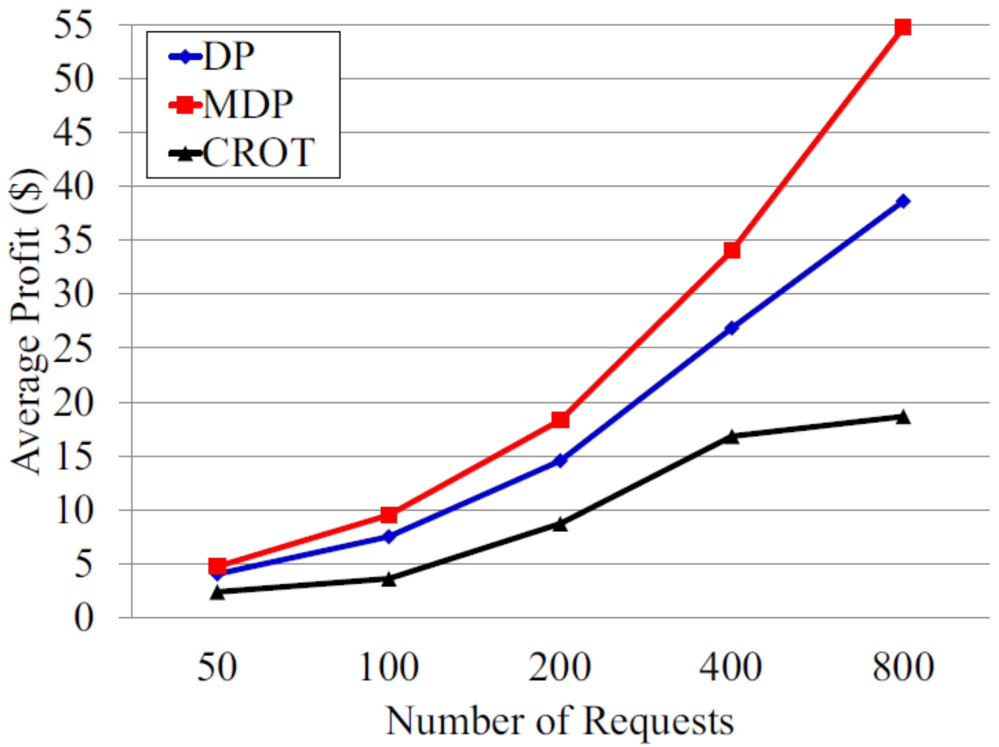
Simulator Implementation

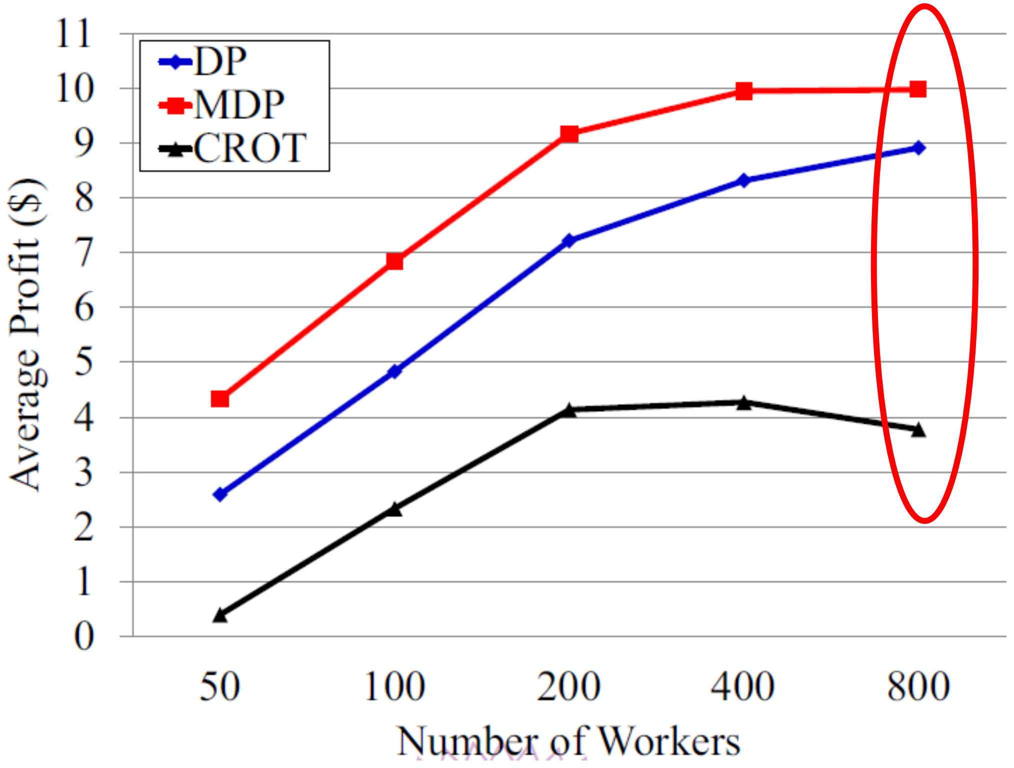
- The following trace-driven simulators are implemented in JAVA
- It supports three algorithms
  - MDP algorithm
  - DP algorithm
  - A baseline algorithm – CROT

Simulation Design

- Parameters
  - *N*: number of requests: {50, 100, 200, 400, 800}
  - *W*: number of workers: {50, 100, 200, 400, 800}
- Metrics
  - Average profit
  - Average traveling cost
  - Completed requests ratio

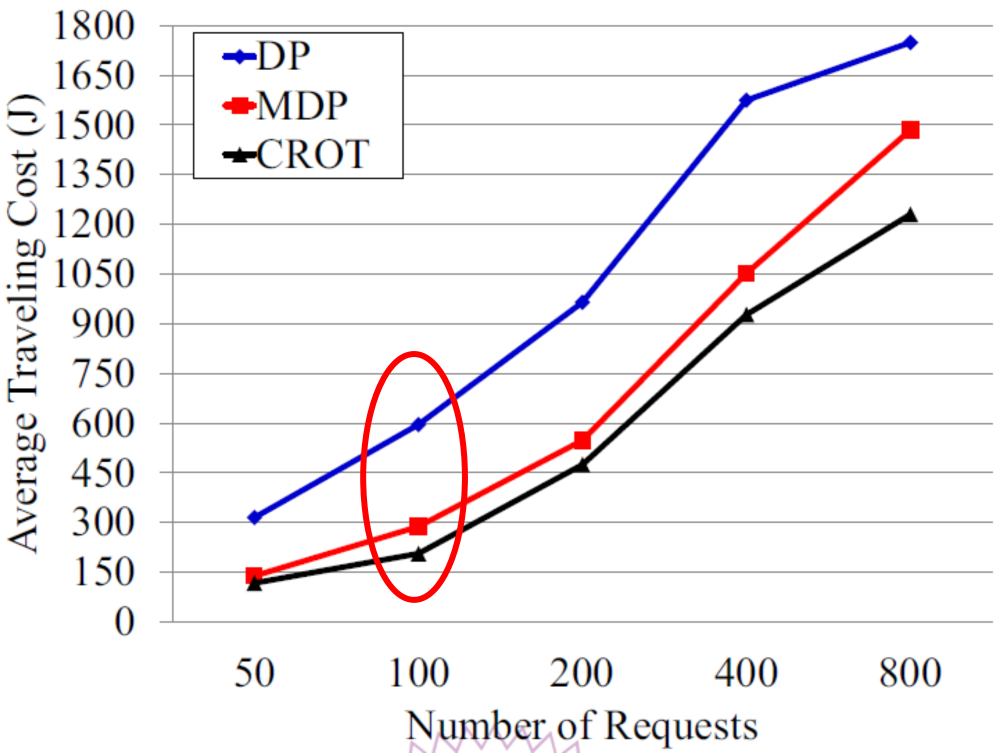
Average Profit

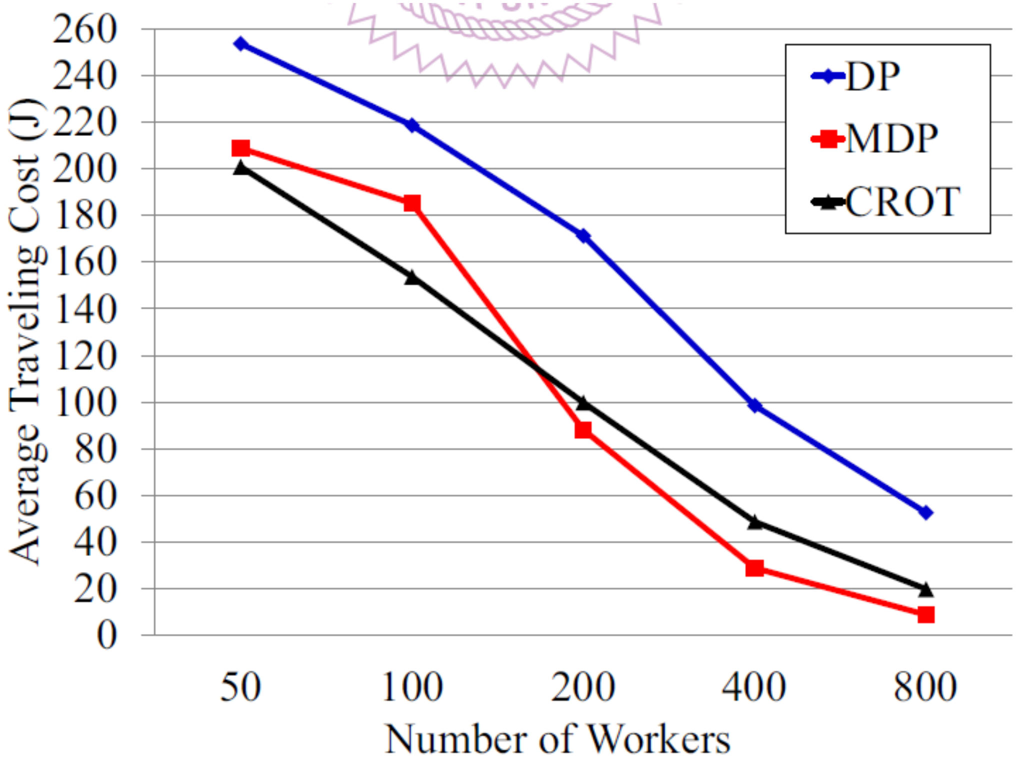




- MDP achieves 2.64 times the profit of the baseline

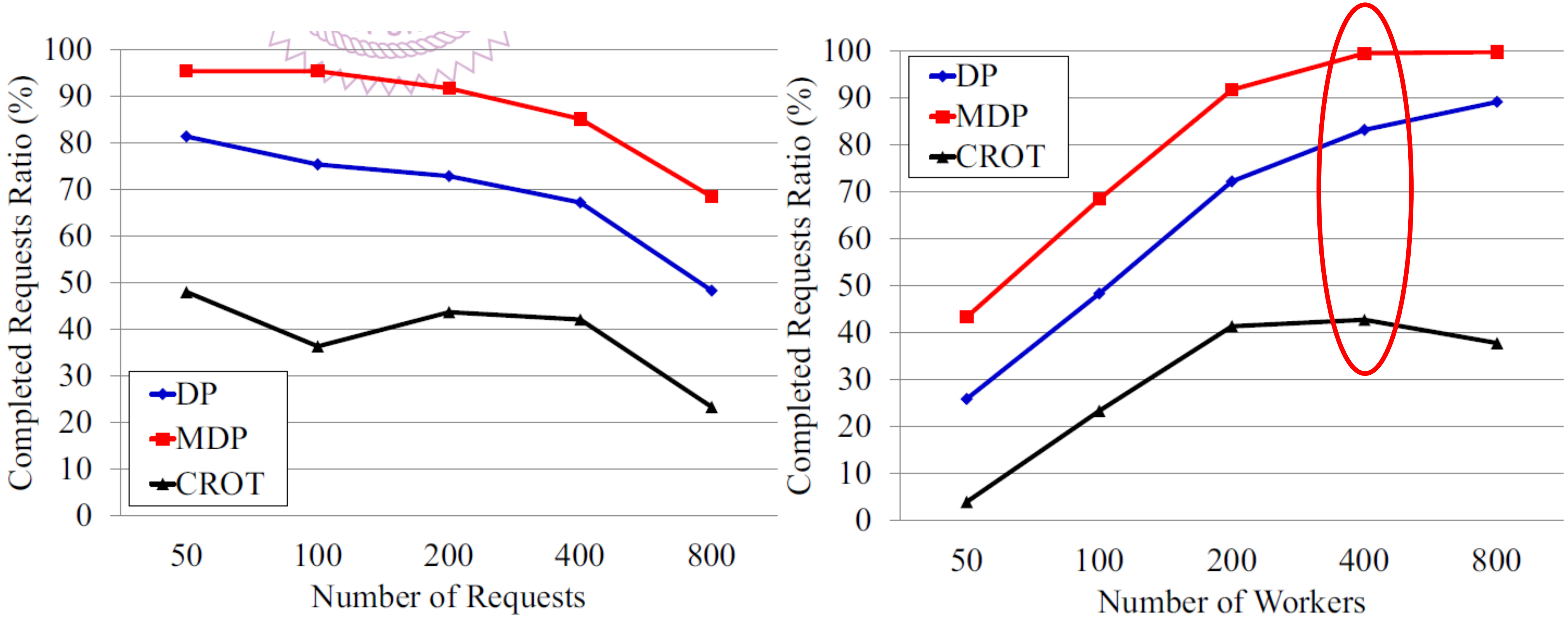
Average Traveling Cost





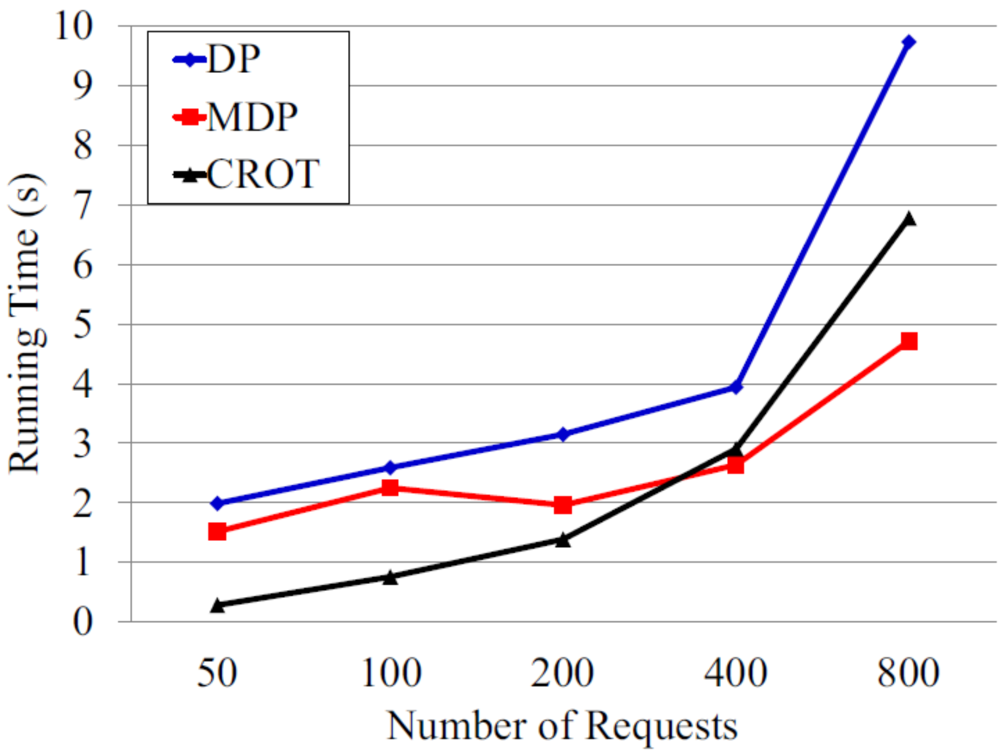
- MDP saves 51% traveling cost compared to DP

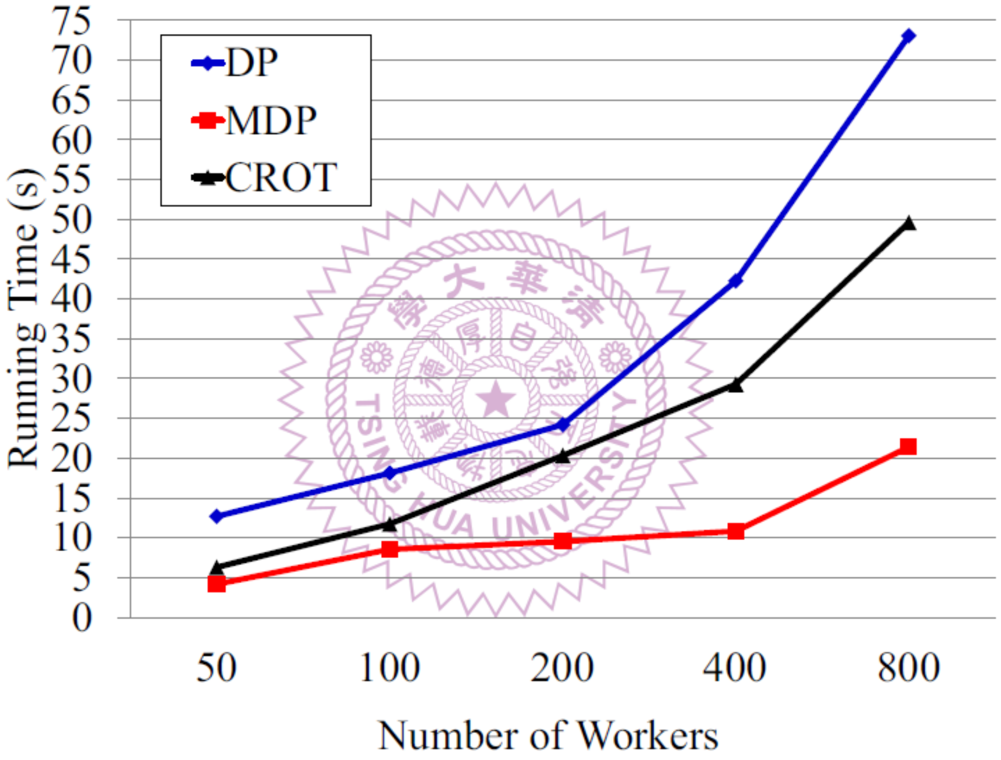
Completed Requests Ratio



- MDP achieves almost 100% completed requests ratio at 400 workers

Running Time





- MDP outperforms others at every cases

Outline

1. Introduction
   - Motivation
   - Crowdsensing
   - Research Problems
2. Related Work
3. Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
4. Multi-user Detour Planning Problem
   - Formulation
   - Solution
   - Evaluation
5. Conclusion and Future Work

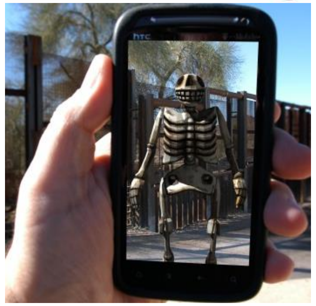
Conclusion

- We propose a mobile crowdsensing system (MCS), and we discuss and formulate detour planning problem and multi-users detour planning problem.
- We address detour planning algorithm (DP), approximation detour planning algorithm (DPA) and multi-users detour planning algorithm (MDP) for solving the proposed problems.
- We implemented traced-driven simulators, and the results show:
  (1) DP achieves optimal profit, and DPA runs efficiently
  (2) MDP outperforms other algorithms in average profit and traveling cost

Contribution	1	2	3	4
Detour planning algorithm (DP)	V	V		
Multi-user detour planning algorithm (MDP)	V	V	V	V

Future Works

- Gamified crowdsensing
  - Combine games and requests for attracting workers to play and earn rewards
  - Challenges:
    (1) Unify games to the system, and players play games smoothly
    (2) Use augmented reality to let players shoot photos to trigger a new game event



- Design the result upload mechanism and verification for better performance
  - Challenges:
    (1) How to efficiently upload results in different network conditions (e.g. 3G, and WiFi)
    (2) Whether the results are real or fake (e.g. timestamp, and GPS)
    (3) There are privacy issues about tracking locations of workers
  - Apply to Urban Computing

Demo, and Q & A

- http://youtu.be/9WFfQjq8pTs

END