Multi-Agent Task Assignment for Mobile Crowdsourcing under Trajectory Uncertainties

(Extended Abstract)

Cen Chen Shih-Fen Cheng Archan Misra Hoong Chuin Lau
School of Information Systems, Singapore Management University
80 Stamford Road, Singapore 178902
{cenchen.2012, sfcheng, archanm, hclau}@smu.edu.sg

ABSTRACT

In this work, we investigate the problem of mobile crowdsourcing, where workers are financially motivated to perform location-based tasks physically. Unlike current industry practice that relies on workers to manually browse and filter tasks to perform, we intend to automatically make task recommendations based on workers' historical trajectories and desired time budgets. However, predicting workers' trajectories is inevitably faced with uncertainties, as no one will take exactly the same route every day; yet such uncertainties are oftentimes abstracted away in the known literature. In this work, we depart from the deterministic modeling and study the stochastic task recommendation problem where each worker is associated with several predicted routine routes with probabilities. We formulate this problem as a stochastic integer linear program whose goal is to maximize the expected total utility achieved by all workers. We further exploit the separable structure of the formulation and apply the Lagrangian relaxation technique to scale up the solution approach. Experiments have been performed over the instances generated using the real Singapore transportation network. The results show that we can find significantly better solutions than the deterministic formulation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Human Factors

Keywords

crowdsourcing, mobile crowdsourcing, multiagent planning

1. INTRODUCTION

Mobile crowdsourcing is a rapidly growing extension to the traditional crowdsourcing paradigm, characterized by mobile workers financially motivated to perform location-based tasks physically. Examples of mobile crowdsourcing tasks include citizen sensing (ask participants to contribute sensor readings such as pollution, congestion, noise level), store audits (e.g., checking shelfs, store displays), logistics (package pickup and delivery), to name a few.

Most research and deployments of mobile crowdsourcing presently employ a *pull-based* model in which individual crowdworker independently select from the list of available tasks. As pointed out by Musthag and Ganesan [2], such pull-based embodiments of mobile crowdsourcing, suffer from the phenomenon of *super agents*; i.e., a small percentage of crowdworkers who perform the majority of tasks. Such phenomenon is undesirable, as many ordinary crowdworkers might drop out as a result of not having enough tasks. By examining empirical data, they conclude that the major difference between ordinary worker agents and super agents is the latter's ability in planning better routes and choosing tasks that fit their routes best. Build upon this insight, Chen et al. [1] investigate an alternative *push-based* model, where the crowdsourcing platform centrally plan for task assignment (it's called *assignment* in their paper, but it's really a *recommendation*, as agents still make decisions independently), by assuming that each mobile crowdworker has only one *deterministic* routine route. Our work addresses more realistic scenarios where an individual worker's trajectory has inherent uncertainty, assuming that each agent's list of possible routine trajectories is finite, and governed by a known probabilistic distribution.

2. THE MODEL

This problem can be viewed as a special routing problem with time budget and stochastic routine route constraints. We denote N as the set containing both the routine and task nodes (denoted as N_t), and for all pairs (i, j), where $i, j \in N$, let t_{ij} be the travel time from i to j. Let K be the set of agents, and let M_k be the set of agent k's routine routes. For each route $m \in M_k$, let β_k^m be the probability that agent k would use route m, R_k^m be the collection of all nodes in route m, o_k^m be the origin, d_k^m be the destination, and p_{ik}^m be the visit order for node $i \in R_k^m$. For each task $i \in N_t$, let s_i be its reward, and e_i be its required execution time. The total time budget for agent k's route m is b_k^m.

We define the following decision variables. $y_{ik} \in \{0, 1\}$ is set to 1 when task i is assigned to agent k. $x_{ijk}^m \in \{0, 1\}$ is set to 1 when agent k moves from nodes i to j when the realized routine route is m. $u_{ik}^m \in \{0, \ldots, N\}$ indicates the visit order of node i for agent k, when the realized routine route is m. δ_{ik}^m is set to 1 if task i is assigned to agent k, yet cannot be completed when the realized routine route is m.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

We formulate the problem as an integer linear programming (ILP) model, whose objective is to maximize the expected total rewards earned by all agents, considering uncertainties over their routine routes.

$$\max \sum_{i \in N_t} s_i \sum_{k \in K} (y_{ik} - \sum_{m \in M_k} \beta_k^m \cdot \delta_{ik}^m). \quad (1)$$

$$\sum_{k \in K} y_{ik} \leqslant 1, \qquad \forall i \in N_t. \quad (2)$$

$$\delta_{ik}^m \geqslant y_{ik} - \sum_{j \in N} x_{ijk}^m, \qquad \forall i \in N_t. \quad (3)$$

Constraint (2) ensures that each task is assigned to at most one agent. Constraint (3) extracts whether a task i node is bypassed ($\delta_{ik}^m = 1$) from the flow decision (x_{ijk}^m) when it's assigned to this agent k's route m.

The rest constraints are at agent-route level, i.e., for each pair of $(k, m) \in (K, M_k)$, same set of constraints applies.

$$\sum_{i \in N} x_{idk}^m = \sum_{j \in N} x_{djk}^m, \qquad \forall d \in N \backslash \{o_k^m, d_k^m\}, \quad (4)$$

$$\begin{cases} \sum_{j \in N} x_{djk}^m \leqslant 1, & \forall d \in N \backslash R_k^m, \\ \sum_{j \in N} x_{djk}^m = 1, & \forall d \in R_k^m \backslash \{d_k^m\}, \\ \sum_{i \in N} x_{idk}^m = 1, & d = d_k^m, \end{cases} \quad (5)$$

$$\sum_{i \in N} \sum_{j \in N} (t_{ij} + e_i) \cdot x_{ijk}^m \leqslant b_k^m, \quad (6)$$

$$u_{ik}^m = 1, \qquad i = o_k^m, \quad (7)$$

$$(u_{ik}^m + 1) - u_{jk}^m \leqslant N(1 - x_{ijk}^m), \qquad \forall i, j \in N, \quad (8)$$

$$u_{ik}^m - u_{jk}^m \geqslant p_{ik}^m - p_{jk}^m, \qquad \forall i, j \in R_k^m : p_{ik}^m > p_{jk}^m. \quad (9)$$

Constraints (4)–(5) ensure that flows are consistent at all nodes. The time budget constraint for each routine route is enforced in (6). Constraints (7)–(9) produce visit orders (u_{ik}^m) from flows (x_{ijk}^m), and ensure that all nodes in the routine route m are visited in correct partial order (it's partial since additional nodes can be in-between two routine nodes). This ILP model can be solved using standard solver such as CPLEX. But given the complexity of the model, this is only feasible for very small problem instances.

To scale up the solution approach, we adopt Lagrangian relaxation by moving the complicating constraint (3) into the objective function. We define $\boldsymbol{\lambda} = \{\ldots, \lambda_{ik}^m, \ldots\}$ as the vector of Lagrangian multipliers associated with all constraints in (3), and convert the maximization problem to be a minimization problem $L(\boldsymbol{\lambda})$. Observing the problem structure, we further decompose $L(\boldsymbol{\lambda})$ into the following two classes of subproblems. Firstly, the *assignment subproblem* decides how tasks should be assigned to individual agents:

$$\min \sum_{i \in N_t} s_i \sum_{k \in K} (-y_{ik} + \sum_{m \in M_k} \beta_k^m \cdot \delta_{ik}^m)$$
$$+ \sum_{i \in N_t} \sum_{k \in K} \sum_{m \in M_k} \lambda_{ik}^m \cdot (y_{ik} - \delta_{ik}^m), \quad (10)$$

$$\delta_{ik}^m \leqslant y_{ik}, \qquad \forall i \in N_t, m \in M, k \in K, \quad (11)$$

with constraint (2). Secondly, the agent-route level *routing subproblems* decide the exact node visit sequence for each agent k and each realized routine route m. For each $k \in K$ and $m \in M_k$, we have:

$$\min - \sum_{i \in N_t} (\lambda_{ik}^m \cdot \sum_{j \in N} x_{ij}), \quad (12)$$

with constraints (4)–(9). The dual solution can be calculated by summing up the objective values of all subproblems. The primal solution can be extracted by projecting the routing policy, $\{x_{ijk}^m\}$, into the original optimization problem and optimize the same objective function (1), subject to constraints (2)–(3). We iteratively update $\boldsymbol{\lambda}$ and solve the problem by using standard subgradient descent algorithm.

3. EXPERIMENT RESULTS

The performance of our LR heuristic is evaluated using 40 different synthetic and real-world inspired task scenarios by two metrics: 1) solution quality, and 2) stochastic improvement against baselines. For each task scenario, both tasks and agent's routine route distributions are generated randomly, either on a synthetic or realistic network topology.

We first present the results on solution quality in Table 1. All reported task scenarios are characterized by (k, N_t, N), and due to the scalability issue of the ILP formulation, the largest scenario we can solve is just $(8, 16, 80)$.

(k,N_t,N)	ILP		LR	
	Optimum	Runtime]	Gap	Runtime
(2,4,40)	400	0.8s	0%	0.09s
(4,8,80)	630.2	22.9s	0%	0.2s
(8,16,80)	1560.7*	6558.6s	0.06%	14.8s

Table 1: Solution quality and runtime for ILP and LR.
(*) Terminates early at the optimality gap of 0.06%.

The stochastic improvement over deterministic baseline is shown in Table 2. Task scenarios are generated similarly, but for the deterministic baseline, route with highest probability is selected for each agent as the routine route. For each instance evaluated, 1000 agent route realizations are sampled, and task completion rate is calculated the average of these 1000 samples. The upper bound (UB) percentages reported in Table 2 are the task completion rates achieved with *perfect information*. The gap values reported for LR and Det are compared against reported UB.

Detour	Upper Bound	LR (Gap)	Det (Gap)
10%	22.5%	0.4%	13.8%
20%	40.2%	0.3%	10.9%

Table 2: Stochastic improvement.

4. ACKNOWLEDGMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority(MDA).

REFERENCES

[1] C. Chen, S.-F. Cheng, A. Gunawan, A. Misra, K. Dasgupta, and D. Chander. TRACCS: Trajectory-aware coordinated urban crowd-sourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*, pages 30–40, 2014.

[2] M. Musthag and D. Ganesan. Labor dynamics in a mobile micro-task market. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 641–650, 2013.