Smart Bundling for Crowdsourced Package Deliveries

Larry Lin, Chung-Kyun Han, Shih-Fen Cheng, Hoong Chuin Lau, and Archan Misra
School of Information Systems, Singapore Management University

1 Introduction

Mobile crowdsourcing, which involves the use of a pool of crowd-workers who visit different locations to perform a variety of location-specific tasks, has emerged as a key paradigm for executing many urban services. While *on-demand* crowdsourced transportation (e.g. Uber) has arguably received the greatest attention, *last mile* logistics and package delivery is another service that is rapidly adopting this crowdsourcing paradigm. Crowdsourced package delivery has several key advantages: (1) Logistics companies no longer have to maintain a large dedicated fleet and worker pool, thus reducing their capital expenses; (2) They can tap on a flexible workforce that can handle seasonal demand fluctuations. Despite avid interests in adopting the crowdsourcing paradigm, most current practices suffer from the inefficiency caused by decentralized task allocation. This is so since crowd-workers need to independently browse and choose their preferred tasks, which in most cases are cognitively demanding and rarely optimal.

In this paper, we aim to ease the burden of crowd-workers in task selection and increase the effectiveness of crowdsourced logistics by creating *flow-aware bundles*. A flow-aware bundle is designed to include multiple tasks in anticipation of the participation of anonymous crowd-workers, whose participations are assumed to follow known origin-destination distribution. This is significantly different from the past state-of-the-art [1], since our tasks contain both pick-up and drop-off locations (instead of just a single task location) and we do not know exactly which crowd-workers would show up (implying that we cannot create bundles to cater to individual movement patterns, but have to work with *aggregated* travel patterns).

2 The Model

Our problem can be viewed as a variant of the Pickup and Delivery Problem [2], with the following two major differences: 1) Paths are considered instead of individual vehicles, i.e., bundles are evaluated against aggregated travel patterns (defined by path weights), and not against trajectories of known crowd-workers; 2) Crowd-workers are considered homogeneous, a crowd-worker can perform any task, as long as time permits.

We denote $P = \{1, ..., n\}$ as the set of pickup nodes, and $D = \{n + 1, ..., 2n\}$ as the set of delivery nodes, and let $N = P \cup D$. To come up with definition for paths, we let $\Omega = \{2n + 1, ..., 2n + m\}$ be the set of origin nodes, and $\Pi = \{2n + m + 1, ..., 2n + 2m\}$ be the set of destination nodes. The network graph is defined as $G = (V, A)$, where $V = N \cup \Omega \cup \Pi$ and $A = V \times V$. Let K be the set for all paths, and f_k and w_k be the flow and weight of path k. The objective function of our formulation is:

$$\max_{x} \sum_{b \in B} \sum_{k \in K} w_k \cdot y^{b,k} \cdot \mathcal{R}^b, \tag{1}$$

where B is the set of all bundles, $y^{b,k}$ is the binary variable indicating whether bundle b is feasible for path k, and $\mathcal{R}^b$ is the reward for bundle b. Bundle-related configurations are determined by:

$$\sum_{b \in B} z_i^b = 1, \forall i \in P; \;\; \mathcal{R}^b = \sum_{i \in P} z_i^b \cdot r_i, \forall b \in B; \;\; \text{and} \sum_{i \in P} z_i^b \cdot v_i \leq \lambda, \forall b \in B, \tag{2}$$

where z_i^b indicates whether task i is in b, r_i and v_i are the reward and the volume of task i, and λ is the volume limit for a bundle. The flow conservation is standard, and can be ensured by the following constraints:

$$\sum_{j \in P \cup \{\pi_k\}} x_{\omega_k, j}^{b,k} = 1, \;\; \sum_{i \in D \cup \{\omega_k\}} x_{i, \pi_k}^{b,k} = 1, \forall k, b; \;\; \sum_{j \in N} x_{i,j}^{b,k} = \sum_{j \in N} x_{j,i+n}^{b,k}, \;\; \sum_{j \in N} x_{i,j}^{b,k} = z_i^b, \forall i \in P, k, b; \tag{3}$$

$$\sum_{i \in \{\omega_k\} \cup N} x_{i,j}^{b,k} = \sum_{i \in N \cup \{\pi_k\}} x_{j,i}^{b,k}, \forall j \in N, k, b; \;\; x_{i,i}^{b,k} = 0, \forall i \in N, k, b; \;\; x_{i+n,i}^{b,k} = 0, \forall i \in P, k, b. \tag{4}$$

For the above, ω_k and π_k represent origin and destination of path k, $x_{i,j}^{b,k}$ is the binary decision variable that is set to 1 if the edge between nodes i and j is used by path k when trying to satisfy the bundle b.

Finally, for each bundle-path combination, we need to determine the required travel distance, and use this distance to decide whether the bundle-path pair is feasible:

$$d^{b,k}_{\omega_k} = 0, (d^{b,k}_{\pi_k} - t_{\omega_k,\pi_k} - \delta) \le M(1 - y^{b,k}), \forall k, b; \;\; d^{b,k}_j = \sum_{i \in N \cup \omega_k} x^{b,k}_{i,j} \cdot \left(d^{b,k}_i + t_{i,j}\right), \forall j \in N \cup \{\pi_k\}, k, b. \quad (5)$$

For the above, $d^{b,k}_j$ is the accumulated travel distance for path k to satisfy b, $t_{i,j}$ is the travel distance between i and j, and δ is the maximal detour distance.

To actually solve the above formulation, we need to insert the subtour elimination constraint and linearize several non-linear constraints. These steps are standard and are skipped in the interest of space.

3 The Greedy Heuristic

Although we could obtain optimal solutions by solving the exact formulation, it is not scalable to even moderate-size problems. As a result, we will have to rely on heuristics to solve instances of meaningful sizes. In our study, we design a simple yet effective greedy heuristic approach to generate task bundles.

The design of the heuristic is based on quantifying the attractiveness of a bundle, which is formally defined as: $Att_b = \sum_{k \in K} w_k \cdot y^{b,k} \cdot \mathcal{R}_b$, for bundle b. With this function, we will iteratively add tasks to each bundle that would give the highest bundle attractiveness. This step continues until we reach the volume limit. Also, for every *bundle-path* combination, the evaluation of a task is done by insertion of task pickup and delivery nodes to previous best found sequences via the greedy approach, while is then used to derive the bundle-path feasibility.

4 Experiment Setup and Results

To evaluate the effectiveness of our task-bundling approaches, we made use of a real transportation dataset of Singapore's public train network (`http://www.ezlink.com.sg`), consisting of 90 train stations. We derive the flow on each route by counting commuters traveling from any origin station to any destination station. These flows are then normalized to derive the path flow weights. For these instances, we can only solve them using the greedy heuristic, as the execution time grows exponentially in number of nodes.

Our experiments are designed to provide the following insights in the crowd-sourced logistics planning: 1) the effectiveness of bundling based on path flow weights versus trajectories, 2) flow-sensitive bundling versus flow-insensitive bundling, 3) bundled versus non-bundled (assuming workers are myopic).

Our findings are summarized below: 1) as expected, trajectory-based bundle generation outperforms flow-based bundle generation, allowing almost 100% of task completion (compared to 80% for the flow-based approach); however, from the worker's perspective, the saving in terms of the detour time is only around 3 minutes, which is quite minor; 2) considering flows bring about 10% improvement in task completion, which is significant; 3) compared to the bundled approach, myopic workers can finish 100% tasks, but at the expense of mobilizing almost 4 times more workers, while each worker is given only half the task amount.

For future work, we are exploring alternatives to improve the performance and the realism. In particular, we would want to incorporate more realistic worker behaviors, such as probabilistic acceptance of task bundles and worker-specific physical limits.

References

[1] S.-F. Cheng, C. Chen, T. Kandappu, H. C. Lau, A. Misra, N. Jaiman, R. Tandriansyah, and D. Koh. Scalable urban mobile crowdsourcing: Handling uncertainty in worker movement. *ACM Transactions on Intelligent Systems and Technology*, to appear, 2017.

[2] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.