Maximizing the Utility in Location-Based Mobile Advertising

Peng Cheng
HKUST
Hong Kong, China
pchengaa@cse.ust.hk

Xiang Lian
Kent State University
Ohio, USA
xlian@kent.edu

Lei Chen
HKUST
Hong Kong, China
leichen@cse.ust.hk

Siyuan Liu
Pennsylvania State University
Pennsylvania, USA
siyuan@psu.edu

***Abstract*—Nowadays, the locations and contexts of users are easily accessed by mobile advertising brokers, and the brokers can send customers related location-based advertisement. In this paper, we consider a location-based advertising problem, namely maximum utility advertisement assignment (MUAA) problem, with the estimation of the interests of customers and the contexts of the vendors, we want to maximize the overall utility of ads by determining the ads sent to each customer subject to the constraints of the capacities of customers, the distance ranges and the budgets of vendors. We prove that the MUAA problem is NP-hard and intractable. Thus, we propose one offline approach, namely the reconciliation approach, which has an approximation ratio of $(1-\epsilon)\cdot\theta$, where $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$, and n_i^c is the larger value between the number of valid vendors and the capacity a_i of customer u_i. Experiments on real data sets confirm the efficiency and effectiveness of our proposed approach.**

I. Introduction

Nowadays, *location-based mobile advertising* (LBA) can pinpoint the potential customers' locations and provide location-based advertisement on their mobile devices. LBA has drawn much attention from industry (e.g., MobileAds [3]). Specifically, in LBA, vendors create campaigns on the broker system with the specified information of ads and budgets to cover the ad fee of the broker. Then, the broker system sends LBA ads to potential customers based on their current locations, profiles (e.g., occupation) and preferences (e.g., interests) with a goal to increase the influence of the vendors and lure in interested customers. However, it is difficult to push ads to the most suitable customers with the limited budget such that the overall utility of the ads is maximized as the inherent complexity of the problem and the hardness of estimating the utility of each ad.

Under this background, in this paper, we will consider a practical problem, namely *maximum utility advertisement assignment* (MUAA), which pushes ads to suitable customers to maximize the overall utility subject to the constraints of distance ranges, capacities of customers and the limited budget. Here, the utility of an ad sent to a customer is the measure of its effect on attracting the customer to visit the shops of the ad owner.

Existing studies in LBA focus on investigating the customer attitudes [5], analyzing the business models [6] and proposing approaches to solve the continuous vendor selection problem [12]. However, no existing works studied designing a good LBA ad strategy to match the vendors and customers having budget-constrained vendors with multiple ad types to select.

In this paper, we first prove that our MUAA problem is NP-hard. As a result, the MUAA problem is not tractable. Therefore, in order to efficiently handle the MUAA problem, we proposed an approximate offline algorithm, namely the reconciliation approach, in Sections III. We conducted experiments on real data sets to show the efficiency and effectiveness of our approach in Section IV.

II. Problem Definition

Assume that customers and vendors are depicted with a set of tags, such as "fast foods", "sport shoes" and "electronic devices". Let $\Psi = \{g_1, g_2, ..., g_w\}$ be a universe of w tags.

Definition 1. (Spatial Customers) Let $U_\varphi = \{u_1, u_2, ..., u_m\}$ be a set of m customers at timestamp φ. Each customer, u_i, is at location $l(u_i, \varphi)$ at timestamp φ with a limited number a_i of the received ads. Moreover, customer u_i has a vector ψ_i of his/her interests on the tags.

Definition 2. (Spatial Vendors) Let $V_\varphi = \{v_1, v_2, ..., v_n\}$ be a set of n spatial vendors at timestamp φ. Each spatial vendor v_j is located at $l(v_j)$, specifies a circular area of radius r_j centered at $l(v_j)$, and provides a limited budget B_j to support its ads. Also, a tag vector ψ_j is specified for depicting the characteristics of the vendor.

The vector ψ_j reflects the similarities of the vendor on the tags. Any element $\psi_j^{(k)} \in \psi_j$ is in the range $[0, 1]$ and depicts the relevance degree of vendor v_j for the tag g_k.

Definition 3. (Ad Types) Let $T = \{\tau_1, \tau_2, ..., \tau_q\}$ be a set of q types of ads. For each ad type τ_k, it costs c_k and has a utility effectiveness β_k.

Definition 4. (Ad Assignment Instance Set) An ad assignment instance set, donated by $\mathbb{I}$, is a set of triples in the form $\langle u_i, v_j, \tau_k\rangle$, where each customer $u_i \in U$ is assigned with an ad of a vendor $v_j \in V$ in a type $\tau_k \in T$.

For an ad assignment instance $\langle u_i, v_j, \tau_k\rangle$, we evaluate its utility λ_{ijk} with a similar method in [11] to derive the equation as below:

$$\lambda_{ijk} = \beta_k \cdot \frac{s(u_i, v_j, \varphi)}{d(u_i, v_i, \varphi)}, \tag{1}$$

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00158

where β_k is the utility effectiveness of ads in type τ_k. $s(u_i, v_j, \varphi)$ indicates the temporal preference of u_i towards v_j at timestamp φ and $d(u_i, v_i, \varphi)$ represents the distance between u_i and v_j at timestamp φ.

Since the shops/services of a tag may be in different active status from time to time, we use $\alpha_x(\varphi)$ to reflect the active level of tag g_x at timestamp φ. We use the *weighted Pearson correlation coefficient* [8] to define $s(u_i, v_j, \varphi)$ as below:

$$\begin{aligned} \mathrm{m}(\psi_i, \varphi) &= \frac{\sum_x \alpha_x(\varphi) \cdot \psi_i^{(x)}}{\sum_x \alpha_x(\varphi)}, \\ \mathrm{cov}(\psi_i, \psi_j, \varphi) &= \frac{\sum_x \alpha_x(\varphi)(\psi_i^{(x)} - \mathrm{m}(\psi_i, \varphi))(\psi_j^{(x)} - \mathrm{m}(\psi_j, \varphi))}{\sum_x \alpha_x(\varphi)}, \\ \mathrm{s}(u_i, v_j, \varphi) &= \frac{\mathrm{cov}(u_i, v_j, \varphi)}{\sqrt{\mathrm{cov}(u_i, u_i, \varphi)\mathrm{cov}(v_j, v_j, \varphi)}}, \end{aligned} \tag{2}$$

where, $\mathrm{m}(\psi_i, \varphi)$ indicates the weighted temporal mean of vector ψ_i and $\mathrm{cov}(\psi_i, \psi_j, \varphi)$ means the weighted temporal covariance of two vector ψ_i and ψ_j.

Definition 5. (Maximum Utility Advertisement Assignment Problem, MUAA) Given a set of spatial customers U and a set of spatial vendors V, the problem of MUAA is to obtain an ad assignment instance set, such that:

1) for any instance $\langle u_i, v_j, \tau_k \rangle$, u_i is located in the specified area of v_j, that is, $d(u_i, v_j) \leq r_j$;
2) the number of assigned ads for each u_i is not more than one's ad number limit u_i;
3) for each vendor v_j, the total cost of the assigned ads does not exceed its budget B_j, that is, $\sum_{\forall \langle u_i, v_j, \tau_k \rangle \in \mathbb{I}} c_k \leq B_j$; and
4) the overall utility, $\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}} \lambda_{ijk}$, of the ad assignment instance sets $\mathbb{I}$ is maximized.

Theorem II.1. *(The Hardness of the MUAA Problem) The maximum utility ad assignment (MUAA) problem given in Definition 5 is NP-hard.*

Proof. We prove the theorem by a reduction from the 0-1 knapsack problem. A 0-1 knapsack problem can be described as follows: Given a set, C, of n items i numbered from 1 up to n, each with a weight w_i and a value x_i, along with a maximum weight capacity W, then find a subset C' of C that maximizes $\sum_{i \in C'} x_i$ subjected to $\sum_{i \in C'} w_i \leq W$.

For a given 0-1 knapsack problem instance, we can transform it to an instance of MUAA as follows: we generate a MUAA problem with only one customer u_0 and one vendor v_0. Then, we give n valid ad assignment instances, such that for each instance tuple $\langle u_0, v_0, \tau_i \rangle$, the ad cost $c_i = w_i$ and the evaluated utility $\lambda_{00i} = x_i$. Also, we set the budget $B_0 = W$. Then, for this MUAA instance, we want to achieve an ad assignment instance set $\mathbb{I}$ that maximizes the overall utility $\sum_{\forall \langle u_0, v_0, \tau_i \rangle \in \mathbb{I}} \lambda_{00i}$ subjected to $\sum_{\forall \langle u_0, v_0, \tau_i \rangle \in \mathbb{I}} c_i \leq B_0$. This way, we can reduce the 0-1 knapsack problem to the MUAA problem. Since the 0-1 knapsack problem is known to be NP-hard [10], MUAA is also NP-hard. □

III. The Reconciliation Approach

In this section, we propose an efficient reconciliation algorithm to handle the MUAA problem, which first solves the single-vendor problems for each vendor separately (i.e., without considering the conflicts from other vendors) with existing algorithms of multi-choice knapsack problem [7], [9], and then reconciles the conflicts on the limited numbers of receiving ads of the customers to provide a global assignment strategy with the accuracy guarantee.

A. The Single-Vendor problem

We discuss the single-vendor problem, where only one vendor exists. Specifically, we assume that if there is only one vendor v_o existing, we can obtain a single-vendor problem $\mathbb{M}_o$, which is presented in a linear programming problem:

$$\begin{aligned} \max \quad & \sum \lambda_{iok} \cdot x_{iok} \\ \text{s.t.} \quad & d(u_i, v_o) \cdot x_{iok} \leq r_o, \qquad i = 1, \ldots, m; k = 1, \ldots, q, \\ & \sum_{i=1}^{m} \sum_{k=1}^{q} c_k \cdot x_{iok} \leq B_o, \\ & \sum_{k=1}^{q} x_{iok} \leq 1, \qquad i = 1, \ldots, m, \end{aligned} \tag{3}$$

where, x_{ijk} is an indicator, if an ad in type τ_k of vendor v_j is sent to customer u_i, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$. This single-vendor problem $\mathbb{M}_o$ is a variant of the multi-choice knapsack problem [7], which can be solved with ε-approximate LP-relaxation algorithm. That is, the utility value of the solution obtained with the ε-approximate LP-relaxation algorithm is at least (1-ε) of that of the optimal solution. In this paper, we use the LP-relaxation algorithm [7] to solve the single-vendor problems.

B. The Reconciliation Algorithm

The MUAA problem generally has multiple vendors who compete for suitable customers, and thus the existing single-vendor algorithms [7] cannot be used directly. Then, we introduce the reconciliation algorithm to reconcile the violations on the limitation of the received number of ads, which iteratively picks a random customer from the set of the customers having ads limitation violations, then resolves the violations by replacing the low-utility customer-and-vendor pairs with other available pairs, until all limitation constraints are satisfied.

Algorithm 1 illustrates the reconciliation algorithm, namely ViolationReconcile, to take advantage of the existing single-vendor algorithms and to satisfy the limitation of the received number of ads, which first solves the single-vendor problems, then resolves the violations of ads limitation on customers, and returns a global ad assignment instance set without ad limitation violations.

First, we initialize the global ad assignment instance set $\mathbb{I}$ to an empty set, as no instances exist (line 1). Next, for each vendor v_j, we obtain a set U_j of its valid customers, who are located in the effective range of vendor v_j (i.e., the distance between each valid customer and the vendor v_j is less than radius r_j) (line 3). For each constructed single-vendor problem $\mathbb{M}_j$, we solve it with the Linear Programming solver [2] and obtain the result $\mathbb{I}_j$ (lines 4-5). As there may exist ad limitation violations on customers, we first retrieve a set $\hat{U}$ of customers with violations (i.e., the number of assigned ads in all the results of single-vendor problems is larger than the limited

Algorithm 1: Reconcile Algorithm

Input: A set U_φ of m customers and a set V_φ of n vendors at timestamp φ

Output: An ad assignment instance set $\mathbb{I}$

1 $\mathbb{I} \leftarrow \emptyset$
2 **foreach** $v_j \in V_\varphi$ **do**
3   obtain a set of valid customers U_j
4   construct a single-vendor problem $\mathbb{M}_j$ with v_j and U_j
5   solve $\mathbb{M}_j$ to get the result $\mathbb{I}_j$
6 obtain a set $\hat{U}$ of customers violating their ad limitations
7 **foreach** $u_i \in \hat{U}$ **do**
8   sort the instances of u_i based on their utility values
9   **while** *ad limitation violations of* u_i *exist* **do**
10   delete one ad assignment instance $\langle u_i, v_j, \tau_k \rangle$ with the lowest utility value λ_{ijk} from $\mathbb{I}_j$
11   greedily assign new valid customers to v_j and add the assignment instance to $\mathbb{I}_j$
12 **return** $\mathbb{I} = \bigcup_{j=1}^{n} \mathbb{I}_j$

number a_i of receiving ad of customer u_i) (line 6). Then, we randomly pick one violated customer u_i in each iteration and replace the low-utility customer-and-vendor pairs with other valid pairs to resolve the violations (lines 7-11). Specifically, for the randomly selected customer u_i, we greedily delete its ad assignment instance $\langle u_i, v_j, \tau_k \rangle$ with the lowest utility value (calculated with Equation 1) in $\mathbb{I}_j$ (line 10) and reassign one new valid customer to v_j subject to the constraints of the budget of the vendor and the limitation number of the customer (line 11). We iteratively reduce the number of assigned ads of customer u_i until the number of his/her assigned ads is less than a_i. Finally, we return a union set of violation-free ad assignment instance sets.

C. Performance Analysis

The Approximation Ratio. We present the approximation ratio of Algorithm 1, ViolationReconcile, with the following theorem.

Theorem III.1. ViolationReconcile *has an approximation ratio of* $(1-\varepsilon)\cdot\theta$*, where* $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$*, and* n_i^c *is the larger value between the number of valid vendors and the capacity* a_i *of customer* u_i*.*

Proof. For a given MUAA problem $\mathbb{M}$ with a set U of m customers and a set V of n vendors, we donate its optimal solution as $\mathbb{I}^*$ and the solution achieved by ViolationReconcile as $\mathbb{I}$. Let $\lambda(\mathbb{I})$ be the overall influence value of the instances in $\mathbb{I}$. Then, we have:

$$\lambda(\mathbb{I}) = \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}} \lambda_{ijk} = \sum_{u_i \in U} \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}(u_i)} \lambda_{ijk}.$$

Let U' be the set of customers that are selected in any single-vendor problem's solution $\mathbb{I}_j$, then $U' \subseteq U$. Thus, we have:

$$\lambda(\mathbb{I}) \geq \quad \sum_{u_i \in U'} \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}(u_i)} \lambda_{ijk}. \tag{4}$$

For each customer u_i, let n_i^s be the number of ads sent among all the solutions of the single-vendor problems, that is $n_i^s = \sum_{j=1}^{n} |\mathbb{I}_j(u_i)|$, where $\mathbb{I}_j(u_i)$ is the set of instances associated with u_i in $\mathbb{I}_j$ and $|\mathbb{I}_j(u_i)|$ is its size. Let $\mathbb{I}(u_i)$ be the set of instances associated with u_i in $\mathbb{I}$. If $n_i^s \leq a_i$, $\frac{|\mathbb{I}(u_i)|}{n_i^s} = 1$; if $n_i^s > a_i$, as we only retain $|\mathbb{I}(u_i)| (= a_i)$ ad assignment instances having higher influences compared with the $n_i^s - |\mathbb{I}(u_i)|$ ad assignment instances that are replaced, thus $\frac{|\mathbb{I}(u_i)|}{n_i^s} < 1$. Thus, we have $\frac{|\mathbb{I}(u_i)|}{n_i^s} \leq 1$. Then, the Inequality (4) can be written as:

$$\lambda(\mathbb{I}) \geq \sum_{u_i \in U'} \frac{|\mathbb{I}(u_i)|}{n_i^s} \cdot \sum_{j=1}^{n} \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j(u_i)} \lambda_{ijk}.$$

If a customer u_i we have $n_i^s \leq a_i$, $\frac{a_i}{n_i^c} = 1 = \frac{|\mathbb{I}(u_i)|}{n_i^s}$; if a customer u_i has $n_i^s > a_i$, then $|\mathbb{I}(u_i)| = a_i$. As $n_i^c \geq n_i^s$, $\frac{a_i}{n_i^c} < 1 = \frac{|\mathbb{I}(u_i)|}{n_i^s}$. Then, we have $\frac{|\mathbb{I}(u_i)|}{n_i^s} \geq \min_{u_i \in U'}(\frac{a_i}{n_i^c}), \forall u_i \in U'$. As for any customer $u_k \in U - U'$, $\frac{a_k}{n_k^c} = 1 \geq \min_{u_i \in U'}(\frac{a_i}{n_i^c})$, thus $\min_{u_i \in U}(\frac{a_i}{n_i^c}) = \min_{u_i \in U'}(\frac{a_i}{n_i^c})$. Next, we can have the inequality as below:

$$\begin{aligned} \lambda(\mathbb{I}) \quad &\geq \quad \min_{u_i \in U'}(\frac{a_i}{n_i^c}) \cdot \sum_{u_i \in U'} \sum_{j=1}^{n} \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j(u_i)} \lambda_{ijk} \\ &= \quad \min_{u_i \in U}(\frac{a_i}{n_i^c}) \cdot \sum_{j=1}^{n} \lambda(\mathbb{I}_j) \end{aligned} \tag{5}$$

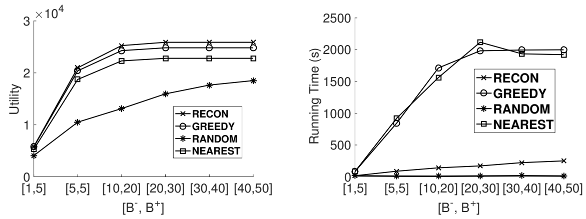
For each created single-vendor problem $\mathbb{M}_j$, we donate its optimal solution as $\mathbb{I}_j^*$ and the solution achieved by the LP-relaxation algorithm in [7] as $\mathbb{I}_j$. As the LP-relaxation algorithm for the multi-choice knapsack problem is a ε-approximate algorithm, we have $\lambda(\mathbb{I}_j) \geq (1-\varepsilon) \cdot \lambda(\mathbb{I}_j^*)$. Then, based on the Inequality (5), we have:

$$\lambda(\mathbb{I}) \geq (1-\varepsilon) \cdot \min_{u_i \in U}(\frac{a_i}{n_i^c}) \cdot \sum_{j=1}^{n} \lambda(\mathbb{I}_j^*) \geq (1-\varepsilon) \cdot \min_{u_i \in U}(\frac{a_i}{n_i^c}) \cdot \lambda(\mathbb{I}^*).$$

Therefore, we have $\lambda(\mathbb{I}) \geq (1-\varepsilon) \cdot \theta \cdot \lambda(\mathbb{I}^*)$, where $\theta = \min_{u_i \in U}(\frac{a_i}{n_i^c})$. Thus, $\mathbb{I}$ is a $(1-\varepsilon) \cdot \theta$-approximate solution for the MUAA problem instance $\mathbb{M}$ w.r.t. the influence of the selected ad instances, which completes the proof. □

IV. Experimental Study

Data Sets. For real data sets, we used one check-in data set of Foursquare [13], which includes 573,703 check-ins from 2,293 users towards 61,858 venues in Tokyo from 12 April 2012 to 16 February 2013 extracted from the Foursquare application through the public API. Each check-in record contains the timestamp, the ID of the user, and the ID, category, and location of the venue. In addition, we use the locations of the venues to initialize the locations of the vendors, and use the locations and timestamps of the check-ins to initialize the corresponding information of the customers. In the experiments on the real data set, we only use the check-ins related to the venues having at least 10 check-ins, which is 441,060 check-ins of 7,222 venues. In other words, we have 441,060 chances to post ads to 2,293 customers (here one chance to post ads to customers means one customer appears in the system) and 7,222 vendors in the real data. For simplicity, we first linearly map check-in locations from Foursquare into a $[0,1]^2$ data



(a) Utility Value (b) Running Time

Fig. 1. Effect of the Range $[B^-, B^+]$ of Budgets.

space, and then modulo the arrival times of customers into 24 hours in the real data accordingly (i.e., ignore the date information of the timestamps of check-ins). According to a report [4] of the statistics information of the AdWords system [1] from 2006 to 2016, we use the average cost per click (the amount of money to pay for one ad click) and the average click through rate (i.e., the number of clicks that an ad receives divided by the number of times the ad is shown) to initialize the prices and the utility effectiveness of ad types, respectively. We simulate the budget B_j of each vendor v_i with Gaussian distribution $\mathcal{N}(\frac{B^-+B^+}{2}, (B^+-B^-)^2)$ within range $[B^-, B^+]$, for $0 < B^- \leq B^+ < 1$. Similarly, we can generate the radius values of the range where vendor v_j wants to post ads and the capacities of customers with Gaussian distributions within the range of the values of $[r^-, r^+]$ and $[a^-, a^+]$. The default values are: $[2, 3]$ for $[r^-, r^+]$ and $[1, 4]$ for $[a^-, a^+]$

Measures and Competitors. We evaluate the effectiveness and efficiency of our MUAA processing approach, in terms of the overall utility score and the CPU time. Specifically, the CPU time is given by the average time cost of performing MUAA assignment for a single customer. We compare the *reconciliation* (RECON) algorithm with a random (RANDOM) method (which randomly assigns vendors' ads to valid customers under the budget constraint), a greedy (GREEDY) method (which iteratively selects one "current best ad instance" that has the current highest budget efficiency), and a nearest-neighbor (NEAREST) method (iteratively selects one "current nearest ad instance" that has the current shortest distance between the vendor and the worker). The experiments were run on an Intel Xeon X5675 CPU @3.07 GHZ with 32 GB RAM in Java.

Effect of the range $[B^-, B^+]$ of vendor budgets. Figure 1 illustrates the experimental results on different ranges, $[B^-, B^+]$, of the vendor budgets B_j from [1,5] to [40, 50]. In Figure 1(a), the total utility values of all the tested approaches improve when the value range of the vendor budgets becomes larger firstly, and then remain with high values when the range of the vendor budgets reaches [20, 30]. The reason is that at the beginning, the budgets of vendors are low and insufficient thus only a small portion of the possible ad instances can be selected and the overall utility values of all the approaches are low. When the average budgets increase, the algorithms can select more ad instances, which leads to higher overall utility values. However, when the budgets are enough to select all the good ad instances, the overall utility values of our three approaches stay at a high level. RECON can always achieve higher overall utility values compared to the competitors. As shown in Figure 1(b), the running times of RANDOM are always lower than 10 seconds, but that of GREEDY, NEAREST and RECON grow with the increase of the average budgets. For RECON, higher vendor budgets cause the number of the selected ad instances for each single-vendor problem to increase, which in turn increases the number of violations of the capacity constraints of customers, and then the time for reconciling them as well. GREEDY and NEAREST use more time than RECON.

V. Conclusion

In this paper, we propose the problem of maximum utility ad assignment (MUAA) problem, which assigns a set of "best" ad instances of vendors to a given customer with the goal of maximizing the utility of the ads under the constraints of the budgets of vendors and the capacities of the customers. We prove that MUAA is NP-hard. We design the reconciliation approach, which has an approximation ratio of $(1-\epsilon)\cdot\theta$, where $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$, and n_i^c is the larger value between the number of valid vendors and the capacity a_i of customer u_i. Experiments on real data sets confirm the efficiency and effectiveness of our offline algorithm.

VI. Acknowledgment

Peng Cheng and Lei Chen are partially supported by the Hong Kong RGC GRF Project 16207617, the National Science Foundation of China (NSFC) under Grant No. 61729201, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, Hong Kong ITC ITF grants ITS/391/15FX and ITS/212/16FP, Didi-HKUST joint research lab project, Microsoft Research Asia Collaborative Research Grant and Wechat Research Grant. Xiang Lian is partially supported by NSF OAC No. 1739491 and Lian Start Up No. 220981, Kent State University. Siyuan Liu is partially supported by Natural Science Foundation of China: 61572488 and 61673241. Peng Cheng is the corresponding author.

References

[1] [online] AdWords. https://adwords.google.com/.

[2] [online] LP Solve. http://lpsolve.sourceforge.net/5.5.

[3] [online] MobileAds. http://www.mobileads.com.

[4] [online] the cost of pay-per-click (ppc) advertisingtrends and analysis. https://www.hochmanconsultants.com/cost-of-ppc-advertising/.

[5] H. K. Chowdhury, N. Parvin, et al. Consumer attitude toward mobile advertising in an emerging market: An empirical study. *IJMM*, 2006.

[6] S. Dhar and U. Varshney. Challenges and business models for mobile location-based services and advertising. *CACM*, 2011.

[7] T. Ibaraki, T. Hasegawa, K. Teranaka, and J. Iwase. The multiple choice knapsack problem. *J. Oper. Res. Soc. Japan*, 21:59–94, 1978.

[8] F. Pozzi, T. Di Matteo, and T. Aste. Exponential smoothing weighted correlations. *EPJ B*, 85(6):1–21, 2012.

[9] P. Sinha and A. A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

[10] V. V. Vazirani. Approximation algorithms. Springer Science & Business Media, 2013.

[11] D. Wu, M. L. Yiu, and C. S. Jensen. Moving spatial keyword queries: Formulation, methods, and analysis. *ACM TODS*, 2013.

[12] W. Xu, C.-Y. Chow, et al. Calba: capacity-aware location-based advertising in temporary social networks. In *ACM SIGSPATIAL*, 2013.

[13] D. Yang et al. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE SMC*, 2015.