# Maximizing the Utility in Location-Based Mobile Advertising

Peng Cheng [#], Xiang Lian [*], Lei Chen [#], Siyuan Liu [†]

[#] *Hong Kong University of Science and Technology, Hong Kong, China*
{pchengaa, leichen}@cse.ust.hk

[*] *Kent State University, Ohio, USA*
xlian@kent.edu

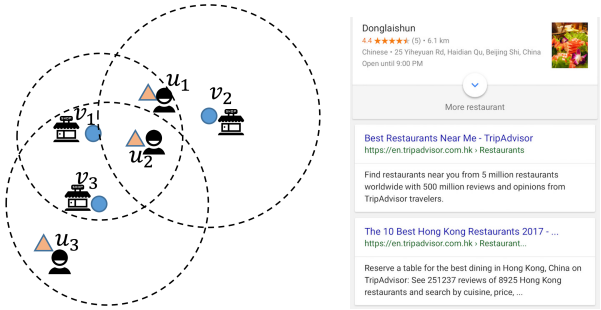
[†] *Pennsylvania State University, Pennsylvania, USA*
siyuan@psu.edu

***Abstract*—With the rapid development of mobile technology, nowadays, people spend a large amount of time on mobile devices. The locations and contexts of users are easily accessed by mobile advertising brokers, and the brokers can send customers related location-based advertisements. In this paper, we consider an important location-based advertising problem, namely maximum utility ad 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 range 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, 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 $u_i$. In addition, we also address the online scenario, in which customers arrive in a streaming fashion, and then propose one novel online algorithm, namely the online adaptive factor-aware approach, which has a competitive ratio (compared to the optimal solution of the offline scenario) of $\frac{\ln(g)+1}{\theta}$, $g > e$, where $e$ is the base of the natural logarithm. Through extensive experiments, we demonstrate the efficiency and effectiveness of our proposed approaches over both real and synthetic datasets.***

## I. Introduction

Nowadays, with the popularity of mobile devices, *location-based mobile advertising* (LBA), as a new form of advertising, can pinpoint the potential customers' locations and provide location-based advertisements on their mobile devices, which integrates mobile advertising with location-based services. LBA has drawn much attention from industry (e.g., MobileAds [4] and xAd [6]). 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 ads 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, which will be modeled in Section II. In the sequel, we illustrate the MUAA problem with a motivation example.




(a) Layout of vendors and customers (b) LBA Ads Example

Fig. 1. An Example of Maximizing Utility of LBA Ads.

**Example 1.** ***(Maximizing the utility in location-based mobile advertising.)*** *Consider a scenario in Figure 1(a), in which there are three vendors, $v_1 \sim v_3$, represented by (blue) circular nodes and three customers, $u_1 \sim u_3$, denoted by (yellow) triangles, at 5:00 pm. Each vendor hopes that its ads are sent to customers in a limited range (e.g., one kilometer) indicated with dash circles. Vendor $v_1$ is a noodle restaurant, vendor $v_2$ is a teahouse, vendor $v_3$ is a pizza restaurant. Vendors provide budgets to an advertising broker system to help them attract more customers for their shops (i.e., to maximize the utility of their ads on the customers). Their budgets are both 3 $.*

*Assume that the broker can send ads through two format: Text Link (TL) and Photo Link (PL), as shown in Figure 1(b). Their prices and effectivenesses are shown in Table I. In this example, we assume that three vendors belong to three distinct tags (e.g., teahouse) separately and the preference value of a customer for a vendor represents the similarity of the customer to the corresponding tag of the vendor. For each customer-*

TABLE I
INFORMATION OF AD TYPES.

| Ad Type | Price | Effectiveness |
|---|---|---|
| Text Link | 1 $ | 0.1 |
| Photo Link | 2 $ | 0.4 |

TABLE II
DISTANCE AND PREFERENCE.

| Vendor | Customer | Distance | Preference |
|---|---|---|---|
| $v_1$ | $u_1$ | 2 | 0.3 |
| $v_1$ | $u_2$ | 1 | 0.2 |
| $v_1$ | $u_3$ | 4.5 | 0.7 |
| $v_2$ | $u_1$ | 2 | 0.2 |
| $v_2$ | $u_2$ | 2.5 | 0.3 |
| $v_2$ | $u_3$ | 7.5 | 0.9 |
| $v_3$ | $u_1$ | 4 | 0.6 |
| $v_3$ | $u_2$ | 2.3 | 0.5 |
| $v_3$ | $u_3$ | 2.3 | 0.1 |

*and-vendor pair, the distance and the preference value at 5:00 pm are given in Table II. For example, a customer $u_1$ like a pizza restaurant $v_3$ more than a noodle restaurant $v_1$, but is located closer to $v_1$. In addition, to avoid the ads disturbing the customers too much, each customer may specify his/her limitation of receiving ads. In this example, each customer prefers to receiving at most 2 ads. As will be introduced in Section II, the utility of one ad can be evaluated based on its distance, preference and effectiveness. For example, sending a PL ad of vendor $v_3$ to customer $u_3$ has the utility value of 0.0174 ($= 0.4 \times \frac{0.1}{2.3}$). One possible solution is $\{\langle u_1, v_1, TL\rangle$, $\langle u_2, v_1, PL\rangle$, $\langle u_1, v_2, TL\rangle$, $\langle u_2, v_2, PL\rangle$, $\langle u_3, v_3, PL\rangle\}$, where each element is a three tuple in format $\langle v, c, \tau\rangle$ indicating that a vendor $v$ sends customer $u$ an ad in $\tau$ type. This solution achieves an overall utility value of 0.1704. However, the optimal solution is $\{\langle u_1, v_1, PL\rangle$, $\langle\ u_1,\ v_2,\ PL\rangle$, $\langle u_2, v_2, TL\rangle$, $\langle u_2, v_3, PL\rangle$, $\langle u_3, v_3, TL\rangle\}$, which results in an overall utility value of 0.2093.*

Motivated by the example above, in this paper, we formalize the *maximum utility ad assignment* (MUAA) problem, which focuses on matching vendors with the most suitable customers and deciding the best ad type for each selected customer-and-vendor pair under the constraints of the limited numbers of receiving ads and the total budget, such that the overall utility of the sent ads is maximized. To optimize the overall utility of ads subject to limited budgets has twofold benefits. First, the higher the overall utility is, the better an ad-broker performed, which can attract more vendors to ask the ad-broker help them manage their location-based ads. From the perspective of vendors, the higher overall utility means better ads results (e.g., more ad clicks or customer visits).

Existing studies focus on solving the privacy issues in LBA [8], [9], investigating the customer attitudes towards LBA [12], [24], analyzing the business models of LBA [14], [20] and proposing approaches to solve the continuous vendor selection problem [30]. Specifically, in [30], the approaches will only fire a recalculation process when the relevant vendors of a given customer have changed such that the response to the query of relevant vendors for each customer is fast, which can be used as a subroutine in our problem to search the relevant vendors for each coming customer. However, no existing works studies finding a good LBA ad strategy to match the vendors and customers having budget-constrained vendors with multiple ad types to select. Thus, we propose novel algorithms to solve MUAA efficiently and effectively.

In this paper, we first prove that our MUAA problem is NP-hard, by reducing it from the 0-1 Knapsack problem [26]. As a result, the MUAA problem is not tractable. Therefore, in order to efficiently handle the MUAA problem, we will propose approximate algorithms from both offline and online perspectives.

Specifically, we make the following contributions:

- We formulate a new problem, called the *maximum utility ads assignment* (MUAA), in Section II-C, and we prove that the MUAA problem is NP-hard, and thus intractable in Section II-E.
- We design one efficient offline algorithm, namely the reconciliation approach, in Sections III.
- We propose a novel online algorithm, namely online adaptive factor-aware approach, in Sections IV.
- We conduct extensive experiments on real and synthetic data sets, to show the efficiency and effectiveness of our MUAA approaches in Section V.

In addition, the rest sections of the paper is arranged as follows, we review and compare the previous works on mobile advertising in Section VI and conclude in Section VII.

## II. PROBLEM DEFINITION

In this section, we first define the location-based mobile ads, which can influence customers (e.g., attracting the customers to visit the corresponding vendors). 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. Each customer/vendor has a vector reflecting his/her interests/similarities on the tags, which can be learned from the historical data of customers (e.g., clicking histories of ads, searching engine records or shopping histories) [14] or the ads contents of vendors (e.g., the text message contents of the ads) [10]. In addition, some location-based social networks, like Foursquare, provide a hierarchy structure (taxonomy) of categories of point-of-interests (e.g., shops, bus stations), as shown in Figure 2. As assumed in [36], we utilize the hierarchy structure of Foursquare in this paper, which can also be built with data mining methods (e.g., Latent dirichlet allocation model [10]) if the taxonomy does not exist or we need to update it.

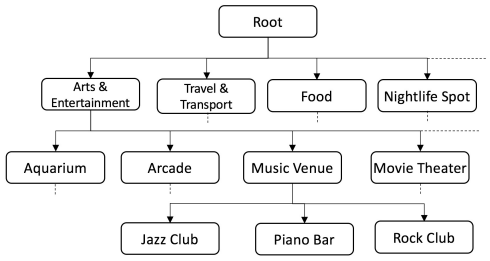



Fig. 2. An Example of categories from Foursquare.

### A. Spatial Customers and Vendors

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

For a customer $u_i$, he or she is located at $l(u_i, \varphi)$. To avoid too many ads sent to her/his mobile device, s/he specifies a maximum number $a_i$ of the received ads. As the customers are moving around and free to come and go, the set of customers may change all the time (i.e., the members of customers are different and the locations of customers are changing). Most importantly, each customer $u_i$ has a vector $\psi_i$ for the tags, which reflects his/her interest levels on the tags. For example, if a customer has visited a set of POIs (e.g., shops and museums), the historical visiting records may reflect the interests of the vendor. Any element $\psi_i^{(k)} \in \psi_i$ is in the range $[0, 1]$ and depicts the interest level of customer $u_i$ for the tag $g_k$. The value of $\psi_i^{(k)}$ can be estimated with the techniques used in recommendation systems (e.g., Collaborative Filtering [7], [15] and Taxonomy Driven Computation [36], [30]). Specifically, we use the taxonomy driven computation method in [36] to calculate the vector $\psi_i$ of customer $u_i$. We briefly introduce the process here. For a customer $u_i$, we first evenly distribute an arbitrary fixed overall score $s$ to the tags (categories) that the customer $u_i$ is interested in (has checked in) as his/her topic scores. Let $h(g_k)$ be the number of checkins of customer $u_i$ performed on tag $g_k$, then the topic score $\text{sc}(g_k)$ of customer $u_i$ for tag $g_k$ is calculated as below:

$$\text{sc}(g_k) = s \cdot \frac{h(g_k)}{\sum_k h(g_k)} \tag{1}$$

Let $E_k = (e_0, e_1, ..., e_q)$ denote the path from the top element $e_0$ to descendant $e_q = g_k$ within the tree-structured taxonomy for the tags in $\Psi$. The explicit checkins (historical records) on the tag $e_q$ may imply the interests on the super-tags $e_m$ ($m < q$). Let $\text{sco}(e_m)$ denote the interest score of customer $u_i$ to the tag $e_m$, then the interest score of tags on path $E_k$ should hold the equation as below:

$$\sum_{m=0}^{q} \text{sco}(e_m) = \text{sc}(g_k) \tag{2}$$

Let $\text{sib}(e_m)$ denote the number of siblings of $e_m$ in the taxonomy. We assume that the sub-tags have *equal shares* of their super-tags within the taxonomy. Then, we have the equation to calculate the interest scores of the ancestors of tag $g_k$ as below:

$$\text{sco}(e_{m-1}) = \kappa \cdot \frac{\text{sco}(e_m)}{\text{sib}(e_m) + 1}, \tag{3}$$

where $\kappa$ is a propagation factor for fine-tuning the profile generation process. Then, the summation of the computed interest score $\text{sco}(e_m)$ for each tag is used to build the interest vector $\Psi_i$ for customer $u_i$.

**Definition 2.** (Spatial Vendors) Let $V_\varphi$ = $\{v_1, v_2, ..., v_n\}$ be a set of $n$ spatial vendors at timestamp $\varphi$. 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 $\psi_j$ is specified for depicting the characteristics of the vendor.

In Definition 2, a spatial vendor $v_j$ is located at location $l(v_j)$. As a result, the customers in the effective area (e.g., the distances from the customers to $v_j$ are smaller than $r_j$ at timestamp $\varphi$) may receive the ads of the vendor. To pay the fee of sending ads, the vendor provides a limited budget $B_j$ for the ad broker system to cover its ads cost. In addition, a vector $\psi_j$ 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 $g_k$. For example, on Foursquare, users are allowed to label the categories of POIs, we can estimate the value of $\psi_j^{(k)}$ for the similarity of the vendor $v_j$ towards the tag $g_k$ (e.g., use the similar method in estimating the vector $\psi_i$ for customer $u_i$). If we cannot know the detailed labeling information, we can simply set $\psi_j^{(k)} = 1$ if the vendor $v_j$ has been classified into the category $g_k$, which can be easily accessed from the API of Foursquare [2].

### B. Ad Types and Ad Assignment Instance Set

We first introduce the definitions of ad types and the ad assignment instance set.

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

As the location-based mobile advertising (LBA) broker may have different methods to push ads to the customers (e.g., text-link, photo-link and in-app video), sending one ad of type $\tau_k$ may need $c_k$ price. Then, the ad may have an utility effectiveness $\beta_k$ on the customer who has viewed it, which measures the probability of a customer taking action/transaction (e.g., using the coupons or visiting the shop) of the ad that he/she just viewed. For example, compared with a text-link ad, a multimedia ad may cost more, but has a better effect (i.e., a high action rate).

**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 $\lambda_{ijk}$ with a similar method in [29] to derive the equation as below:

$$\lambda_{ijk} = \beta_k \cdot \frac{s(u_i, v_j, \varphi)}{d(u_i, v_i, \varphi)}, \tag{4}$$

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

As 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 $\varphi$. For example, a tag of "coffee" may be active in the mornings and the afternoons but inactive at nights. A tag of "Chinese food" may be active in lunch

or dinner time. We use the *weighted Pearson correlation coefficient* [19] to define $s(u_i, v_j, \varphi)$ as below:

$$\begin{aligned}
\mathrm{m}(\psi_i, \varphi) &= \frac{\sum_x \alpha_x(\varphi) \cdot \psi_i^{(x)}}{\sum_x \alpha_x(\varphi)}, \\
\mathrm{cov}(\psi_i, \psi_j, \varphi) &= \frac{\sum_x \alpha_x(\varphi)(\psi_i^{(x)} - \mathrm{m}(\psi_i, \varphi))(\psi_j^{(x)} - \mathrm{m}(\psi_j, \varphi))}{\sum_x \alpha_x(\varphi)}, \\
\mathrm{s}(u_i, v_j, \varphi) &= \frac{\mathrm{cov}(u_i, v_j, \varphi)}{\sqrt{\mathrm{cov}(u_i, u_i, \varphi)\mathrm{cov}(v_j, v_j, \varphi)}},
\end{aligned} \tag{5}$$

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

Intuitively, if a customer and a shop are both interested in or similar to an active tag, the customer may have a higher probability to visit the shop or use the coupon of the ad. For example, a customer who likes coffee may visit a nearby coffee shop sending coupons in the morning. However, the distance of the customer and the vendor may affect the customer's probability of taking actions/transactions: the closer they are, the more likely the customer visits the shop of the vendor.

Note that, to estimate of the utility of an ad instance, we can also use other existing methods in POI (Point-of-Interest) recommendation system [32], [33], [34], [27]. The major contribution of this paper is the ad assignment strategy to maximize the overall utility of the ad instances, which is independent to the definition of the ad utility. In addition, our ad assignment algorithms can work without modification when we use other methods to estimate the utility of ad instances.

### C. The Maximum Utility Ad Assignment Problem

In this section, we will formalize the problem of *maximum utility ad assignment* (MUAA), which matches vendors with the most suitable customers and determines the best ad type for each selected customer-and-vendor pair with a goal of maximizing the overall utility subject to the constraints of the specified ranges and budgets of vendors, and the number of receiving ads of customers.

**Definition 5.** (Maximum Utility Ad Assignment Problem, MUAA) Given a set of spatial customers $U$ and a set of spatial vendors $V$, the problem of *maximum utility ad assignment* (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.

Definition 5 can be rewritten in the form of the linear programming problem below:

TABLE III
SYMBOLS AND DESCRIPTIONS

| Symbol | Description |
|---|---|
| $U_\varphi$ | a set of $m$ spatial customers at timestamp $\varphi$ |
| $V_\varphi$ | a set of $n$ spatial vendors at timestamp $\varphi$ |
| $T$ | a set of $q$ types of ads |
| $u_i$ | a spatial customer |
| $v_j$ | a spatial vendor |
| $\tau_k$ | an ad type |
| $l(u_i)$ (or $l(v_j)$) | the location of customer $u_i$ (or vendor $v_j$) |
| $a_i$ | the capacity of receiving ads for customer $u_i$ |
| $r_j$ | the radius of the range of vendor $v_j$ |
| $B_j$ | the ad budget of vendor $v_j$ |
| $c_k$ | the cost of an ad in type $\tau_k$ |
| $\beta_k$ | the utility effectiveness of an ad in type $\tau_k$ |
| $\lambda_{ijk}$ | the utility value of an ad instance $\langle u_i, v_j, \tau_k \rangle$ |

$$\begin{aligned}
\max \quad & \sum \lambda_{ijk} \cdot x_{ijk} \\
\text{s.t.} \quad & d(u_i, v_j) \cdot x_{ijk} \leq r_j, && i = 1, \ldots, m; j = 1, \ldots, n; k = 1, \ldots, q, \\
& \sum_{i=1}^{m} \sum_{k=1}^{q} c_k \cdot x_{ijk} \leq B_j, && j = 1, \ldots, n, \\
& \sum_{j=1}^{n} \sum_{k=1}^{q} x_{ijk} \leq a_i, && i = 1, \ldots, m, \\
& \sum_{k=1}^{x} x_{ijk} \leq 1, && i = 1, \ldots, m; j = 1, \ldots, n,
\end{aligned}$$

where, $x_{ijk}$ is an indicator, if an ad in type $\tau_k$ of vendor $v_j$ is sent to customer $u_i$, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$.

Table III summarizes the commonly used symbols.

### D. Evaluation Model

In this subsection, we describe the methods to evaluate the approximate offline algorithms (i.e., approximation ratio) and the deterministic online algorithms (i.e., competitive ratio), which will be used to evaluate our proposed the offline reconciliation approach in Section III and online adaptive factor-aware approach in Section IV.

**Approximation Ratio.** The approximation ratio (or approximation factor) of an approximation algorithm is the ratio between the result obtained by the algorithm and the optimal results [26].

**Definition 6.** (Approximation Ratio): The approximation ratio of an approximation algorithm for the MUAA problem is $\varepsilon$ if and only if the inequality below holds for every possible MUAA problem instance $\mathbb{M}$, that is,

$$\begin{cases} \varepsilon \cdot \lambda(\mathbb{I}) \geq \lambda(\mathbb{I}_{opt}), & \varepsilon > 1 \\ \lambda(\mathbb{I}) \geq \varepsilon \cdot \lambda(\mathbb{I}_{opt}), & \varepsilon < 1 \end{cases} \tag{6}$$

where $\lambda(\mathbb{I})$ is the overall utility value of the ad instances selected by the approximation algorithm and $\lambda(\mathbb{I}_{opt})$ is the overall utility value of the ad instances of the optimal result.

**Competitive Ratio.** Competitive analysis [22], [23], [17] is a method to analyze online algorithms, in which the performance of an online algorithm is compared to that of an optimal offline algorithm having the global information in advance.

**Definition 7.** (Competitive Ratio): The competitive ratio of an online algorithm for the MUAA problem is $\sigma$ if and only if the below inequality can hold for every possible MUAA problem instance $\mathbb{M}$,

$$\begin{cases} \sigma \cdot \lambda(\mathbb{I}^*) \geq \lambda(\mathbb{I}_{opt}), & \sigma > 1 \\ \lambda(\mathbb{I}^*) \geq \sigma \cdot \lambda(\mathbb{I}_{opt}), & \sigma < 1 \end{cases} \tag{7}$$

where $\lambda(\mathbb{I}^*)$ is the overall utility value of the ad instances selected by the online algorithm and $\lambda(\mathbb{I}_{opt})$ is the overall utility value of the ad instances of the optimal result.

The approximation and competitive ratios are two measures widely used to evaluate approximation algorithms and online algorithms, respectively. Besides them, we also evaluate the approaches through extensive experiments on both real and synthetic data sets in Section V.

### *E. The Hardness of the MUAA Problem*

We prove that the MUAA problem is NP-hard by reducing it from *0-1 Knapsack problem* [26], which is NP-hard.

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

*Proof:* Please refer to Appendix A of the technical report [11]. ■

From Theorem II.1, we can see that the MUAA problem is NP-hard, and thus intractable. Alternatively, we will later design efficient approximation algorithms to tackle the MUAA problem and achieve good assignment strategies. What is more, in practical system, the customers are highly dynamic, which means customers switch from the available status of receiving ads to the inactive status (i.e., the status of rejecting to receive any ads) very quickly (e.g., several seconds). Therefore, the processing speed is crucial for *location-based mobile advertising* (LBA) to successfully achieve the goal of maximizing the overall utility of the ads. In this paper, we will propose an online algorithm that assigns ads to customer in an online mode where the customers come to the system one-by-one and can reach a high overall utility value (close to the result by offline algorithms).

## 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 [16], [21], 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*

In this subsection, we discuss the single-vendor problem, where only one vendor exists. In Definition 5, an MUAA problem instance $\mathbb{M}$ may contain multiple vendors, and they may send their ads to the same customers to maximize the total utility. Then, some popular customers may receive many ads exceeding their limited numbers of receiving ads. However, if there is only one vendor existing, this kind of scenario will disappear.

Specifically, we assume that if there is only one vendor $v_o$ existing, we can obtain a single-vendor problem $\mathbb{M}_o$. Then, we can present $\mathbb{M}_o$ as the linear programming problem below:

$$\begin{aligned} \max \quad & \sum \lambda_{iok} \cdot x_{iok} \\ \text{s.t.} \quad & d(u_i, v_o) \cdot x_{iok} \leq r_o, \qquad i = 1, \ldots, m; k = 1, \ldots, q, \\ & \sum_{i=1}^{m} \sum_{k=1}^{q} c_k \cdot x_{iok} \leq B_o, \\ & \sum_{k=1}^{q} x_{iok} \leq 1, \qquad i = 1, \ldots, m, \end{aligned} \tag{8}$$

This single-vendor problem $\mathbb{M}_o$ is a variant of the existing problem, multi-choice knapsack problem [16], [21], which can be solved with $\varepsilon$-approximate LP-relaxation algorithm in $O(m)$ time complexity. That is, the utility value of the solution obtained with the $\varepsilon$-approximate LP-relaxation algorithm is at least (1-$\varepsilon$) of that of the optimal solution. In this paper, we use the LP-relaxation algorithm in [16] to solve the single-vendor problems.

### *B. The Reconciliation Algorithm*

The existing algorithm for multi-choice knapsack problem [16], [21] can only solve the single-vendor problems. However, our MUAA problem generally has multiple vendors who compete for suitable customers, and thus the existing single-vendor algorithms [16], [21] cannot be used to solve our MUAA problem directly. We propose a *reconciliation algorithm* to solve the general MUAA problem.

Specifically, 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, and then resolves the violations by replacing the low-utility customer-and-vendor pairs with other available pairs, until all the ad limitation constraints are satisfied.

**Algorithm 1:** Reconcile Algorithm

**Input:** A set $U_\varphi$ of $m$ customers and a set $V_\varphi$ of $n$ vendors at timestamp $\varphi$
**Output:** An ad assignment instance set $\mathbb{I}$

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

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 [3] 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 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 4) 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* $u_i$*.*

The approximation ratio of our ViolationReconcile algorithm (shown in Algorithm 1) consists of two parts: the approximation ratio of the single-vendor problems, which is $1-\varepsilon$ as proved in existing studies [16], and the bounded ratio of $\theta$, which happens when we reconcile reconciles the violations of the customers. For the details of the proof, please refer to Appendix B of the technical report [11].

**The Time Complexity.** We discuss the time complexity of the reconciliation algorithm in Algorithm 1 below. Specifically, the time complexity of retrieving a set of valid customers $U_j$ for vendor $v_j$ is $O(m)$. To solve each single-vendor problem, the time complexity is $O(m\log(m)^{O(1)}/\varepsilon^{O(1)})$ [13]. Thus, lines 2-5 need $O(mn\log(m)^{O(1)}/\varepsilon^{O(1)})$ time. To check the number of assigned ads for each customer needs $O(n)$, line 6 needs $O(mn)$ time. Since each customer can be assigned to at most $m$ vendors, the time complexity of sorting the instances in line 8 needs $O(m\log m)$. Line 10 needs $O(1)$ and line 11 needs $O(m)$. As the loop of lines 9-11 runs at most $n$ times, the time complexity of lines 9-11 is $O(mn)$. In addition, as at most $m$ customers have violations, lines 8-11 run at most $m$ times. Thus, lines 7-11 need time complexity of $O(\max\{m^2\log m, m^2 n\})$. The time complexity of Algorithm 1 is $O(\max\{m^2\log m, m^2 n, mn\log(m)^{O(1)}/\varepsilon^{O(1)}\})$.

## IV. Online Adaptive Factor-Aware Approach

Although the offline algorithm proposed in Section III can achieve good assignment strategies w.r.t the total utility values, in real-world applications, customers are appearing in a streaming fashion and may become inactive within several seconds, thus we cannot know all the information of the customers and vendors at the beginning and apply the offline algorithms. To assign the ads to customers in an online mode, in this section, we propose a deterministic online adaptive factor-aware approach (O-AFA), which has a competitive ratio of $\frac{\ln(g)+1}{\theta}$, $g > e$, where $e$ is the base of the natural logarithm, $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$, and $n_i^c$ is the larger value between the number of valid vendors and the capacity $a_i$ of customer $u_i$. We assume that, for any possible ad assignment instance $\langle u_i, v_j, \tau_k \rangle$, its budget efficiency $\gamma_{ijk} \geq \gamma_{min}$. We will introduce how to set $\gamma_{min}$ and $g$ in Section IV-C. Specifically, our O-AFA approach only selects possible ad instances with adaptively high budget efficiencies and guarantees the constraints of specified ranges and budgets of vendors, and the number of receiving ads of customers.

### A. The Online Adaptive Algorithm

The key issue to design a good online algorithm is to determine whether or not to push an ad to a customer when he/she comes to the system without the information of future customers. If we push ads to every customers at the beginning, then the budgets of vendors will be used up very quickly and no ads can be pushed to customers later. In a word, we should properly distribute the budgets and only push the ads to customers with high budget efficiencies. A simple and direct way is to set a threshold $\phi$ of the budget efficiency and only push ads to customers with budget efficiencies higher than the threshold $\phi$. As discussed in [22], in the online matching and allocation problems, an adaptive threshold will perform better than a static threshold. Different from their approaches using a set of discrete thresholds, we will propose an algorithm with a threshold function of the remaining budget ratio for each vendor to achieve a competitive ratio under the assumption that every ad instance has the budget efficiency $\gamma_{ijk} \geq \gamma_{min}$.

Briefly, when a new customer $u_i$ appears, the O-AFA approach first filters out a set of vendors that the customer is in the range of them. Next, for each vendor $v_j$, if the "best" ad type for it has a higher budget efficiency than a threshold $\phi(\delta_j)$ calculated with a function of the ratio $\delta_j(=\frac{b(\mathbb{I}_j)}{B_j})$ between its used budget $b(\mathbb{I}_j)$ and its total budget, we denote the vendor-customer-ad triple as a potential ad instance (the details of $\phi(\delta_j)$ will be introduced in Section IV-B). Among the potential ad instances of the customer, O-AFA finally picks the top $a_i$ ad instances. The intuition of O-AFA is that when the budget

of a vendor is sufficient at the beginning, we can push its ads to the customers without worrying about using up its budget; when there is just a little portion of the budget left, we need to reserve the remaining budget and only select the ad instances with high budget efficiencies.

Specifically, the pseudo code of the online adaptive factor-aware approach is shown in Algorithm 2. The selected ad instance set $\mathbb{I}$ is initialized as empty at the beginning as no instances are picked (line 1). Then, we select a set of valid vendors for customer $u_i$ according to the spatial constraint of the vendors (i.e., the customer $u_i$ is located in the circular areas of the vendors) (line 2). Next, for each vendor $v_j$, we try to add one "best" ad instance of customer $u_i$ and vendor $v_j$ in to the set $\mathbb{I}$ of the potential ad instances, if and only if the budget efficiency of the ad instance is higher than a threshold function $\phi(\delta_j^{(i)})$, where $\delta_j^{(i)}$ is the ratio between the used budget and the total budget of vendor $v_j$ when customer $u_i$ appears (lines 3-6). Finally, we only return the ad instances with the top-$a_i$ highest budget efficiency values from the set $\mathbb{I}$ (lines 7-9).

**Algorithm 2:** Online Adaptive Factor-Aware Algorithm

**Input:** A customer $u_i$ appears at timestamp $\varphi$ and a set $V_\varphi$ of $n$ vendors at timestamp $\varphi$
**Output:** An ad assignment instance set $\mathbb{I}$ to the customer $u_i$

1 $\mathbb{I} \leftarrow \emptyset$
2 select a set $V'$ of valid vendors for customer $u_i$
3 **foreach** $v_j \in V'$ **do**
4  select one "best" ad tpye $\tau_k$ for customer $u_i$ and vendor $v_j$
5  **if** $\gamma_{ijk} \geq \phi(\delta_j^{(i)})$ **then**
6  add $\langle u_i, v_j, \tau_k \rangle$ to $\mathbb{I}$
7 **if** $|\mathbb{I}| > a_i$ **then**
8  keep the ad instances with the top-$a_i$ highest budget efficiencies in $\mathbb{I}$
9 **return** $\mathbb{I}$

### B. Performance Analysis

In this subsection, we analyze the performance of the O-AFA algorithm through comparing the worst total utility value of the result achieved by the O-AFA algorithm to that of the optimal result for any MUAA problem instance.

In the existing work [35], the authors studied an online knapsack problem with only one vendor. However, in MUAA problem, there are multiple vendors. Inspired by the situation of the online one vendor problem in [35], we design a different threshold function $\phi(\delta_j)$ and prove the competitive ratio of our O-AFA algorithm. In the analysis below, we have two assumptions: 1) we know the lower bound $\gamma_{min}$ of the budget efficiency of any ad instance; 2) the cost of any ad instance is much smaller than the budget of any vendor; 3) the threshold function $\phi(\delta_j)$ is a monotone increasing function of the used budget ratio $\delta_j$ of vendor $v_j$. Note that, the assumption 3) follows the intuition that when the remaining budget of a vendor is low, we should only select the ad instances with high budget efficiencies. Here we denote: a) the primitive integral of $\phi$ as $\Phi$ (i.e., $\Phi(x) = \int \phi(x)dx$); b) the maximum used budget ratio of all the vendors as $\delta_{max}$ when the algorithm finishes; c) a constant $h$ for the value of the threshold function $\phi$ to reach the lower bound budget efficiency $\gamma_{min}$, that is $\phi(h) = \gamma_{min}$. Then, we have a theorem below.

**Theorem IV.1.** *The online algorithm, O-AFA, has a competitive ratio of* $\frac{1}{\theta} \cdot \left(\frac{\phi(\delta_{max})}{h \cdot \gamma_{min} + \Phi(\delta_{max}) - \Phi(h)} + 1\right)$.

The proof of Theorem IV.1 is based on one key observation that any ad instance $\langle u_i, v_j, \tau_k \rangle$ selected by O-AFA has a lower budget efficiency than $\phi(\delta_{max})$. Then we analyze the difference between the selected ad instances and the maximum possible optimal results through the observation. In addition, as the capacity constraint of the customers, the competitive ratio still has a similar factor of $\frac{1}{\theta}$, which also exists in the proof of Theorem III.1. For the details of the proof of Theorem IV.1, please refer to Appendix C of the technical report [11].

**Corollary IV.1.** *The online algorithm, O-AFA, has a competitive ratio of* $\frac{\ln(g)+1}{\theta}$ *when* $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^\delta$ *and* $g > e$.

Corollary IV.1 can be deducted by plugging the definition of $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^\delta$ in Inequality (15). For the details of the proof, please refer to Appendix D of the technical report [11].

**Discussions of the competitive ratio.** When we want to simplify the competitive ratio in Theorem IV.1, we have two concerns: 1) the value $\delta_{max}$ in $\phi(\delta_{max})$ and $\Phi(\delta_{max})$ can be removed; 2) $h \cdot \gamma_{min}$ should equal to $\Phi(h)$. For the first concern, we notice that only the derivative of an exponential function contains the same factor to be removed. However, if we choose some natural exponential functions for constructing $\phi(\delta)$, we cannot meet the second concern. Then we construct $\phi(\delta)$ as a form of $\omega \cdot g^\delta$ to try to simplify the competitive ratio of O-AFA in Theorem IV.1, where $\omega$ and $g$ are two constants.

Next, we need to determine the value of $\omega$ and $g$. When $\phi(\delta) = \omega \cdot g^\delta$, $\Phi(\delta) = \frac{\omega \cdot g^\delta}{\ln(g)}$. Take $h \cdot \gamma_{min} = \Phi(h)$ and $\phi(h) = \gamma_{min}$ into consideration, we have:

$$h \cdot \omega \cdot g^h = \frac{\varphi \cdot g^h}{\ln(g)}.$$

Thus, $h = \frac{1}{\ln(g)}$ and $\omega \cdot g^h = \gamma_{min}$, then we have:

$$\omega = \frac{\gamma_{min}}{e}.$$

Then we have $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^\delta$, and the Inequality (15) can be rewritten as:

$$\frac{\lambda(\mathbb{I}^*)}{\lambda(\mathbb{I})} \leq \frac{\ln(g)+1}{\theta}. \tag{9}$$

As $h = \frac{1}{\ln(g)}$ should be smaller than 1, we obtain $g > e$. Thus, no matter what $g$ (larger than $e$) is chosen, the competitive ratio of O-AFA algorithm is at most $\frac{2}{\theta}$.

**The Time Complexity.** We discuss the time complexity of the online adaptive factor-aware algorithm in Algorithm 2 below. In particular, to retrieve a set of valid vendors for customer $u_i$ needs $O(n)$ (line 2). To keep one "best" ad instance of vendor

$v_j$ to customer $u_i$ needs $O(q)$ where $q$ is the number of ad types (line 4). Thus lines 3-6 need $O(nq)$. Choosing top-$a_i$ ad instances from at most $n$ candidates needs $O(n)$. Thus, in total, the time complexity of Algorithm 2 is $O(nq)$.

### C. Configuration of Parameters

In Section IV-B, we analyze the performance of our O-AFA algorithm based on a known lower bound $\gamma_{min}$ of the budget efficiency of any ad instance. However, in the real system, we cannot know the value of $\gamma_{min}$ in advance and need to estimate its value. In this section, we first introduce the method to estimate the value of $\gamma_{min}$.

*1) Estimation of $\gamma_{min}$:* The main intuition of our O-AFA algorithm (shown in Algorithm 2) is to assign as many ads as possible at the beginning, then it will block ads with low budget efficiencies when the budgets of vendors become less and less. Thus, we can have a simple strategy to update $\gamma_{min}$ when we are running the O-AFA algorithm: we first initialize $\gamma_{min}$ as $+\infty$; when a new customer $u_i$ appears and has a smaller budget efficiency $\gamma_{ijk}$ than the current minimum budget efficiency $\gamma_{min}$, we update $\gamma_{min}$ as $\gamma_{ijk}$. This updating process should be conducted before running O-AFA algorithm for each appearing customer. Similarly, we can use the same strategy to track the current maximum budget efficiency $\gamma_{max}$ to guide the selection of parameter $g$.

*2) Selection of parameter g:* According to definition of competitive ration in Equation (7), to have a higher total utility we should send as many ads as possible after all the customers appear, which is same to utilize as much budget as possible. However, if we simply use a first-come-first-server strategy, we may use up all the budget very fast. Then, we may have no budget to push ads when other good customers appear, which may leads to the low competitive ratio.

Our O-AFA can adaptively select ad instances with high budget efficiency and block low budget-efficiency ads. The parameter $g$ can determine the the blocking effectiveness of O-AFA. Specifically, the larger $g$, the higher O-AFA threshold $\phi(\delta_j^{(i)})$ for each customer-and-vendor pair $\langle u_i, v_j \rangle$, which means only the ad instances with higher budget efficiency will be selected. However, on the other hand, the larger $g$ is, the lower the ratio of used budget will be at the end of the algorithm. In conclusion, from the perspective of picking high-efficiency ad instances, we should have a larger $g$; from the perspective of using as much budget as possible at the end, we should have a smaller $g$.

An observation here is that the ad instance with the maximum budget efficiency $\gamma_{max}$ should always be selected if the remaining budget is sufficient, which means $\phi(1) \leq \gamma_{max}$ then $g \leq \frac{\gamma_{max}}{\gamma_{min}} \cdot e$. In addition, since $e < g$ in Corollary IV.1, we have $e < g \leq \frac{\gamma_{max}}{\gamma_{min}} \cdot e$. The value of g is depended on the real situation of the problem, which can be estimated through the historical records, and we can gradually achieve a proper value of $g$ for the real systems after a period of tuning.

## V. Experimental Study

In this section, we evaluate the effectiveness and efficiency of our MUAA processing approaches based on the experiments on both real and synthetic data sets.

TABLE IV
Experimental Settings.

| Parameters | Values |
|---|---|
| range $[B^-, B^+]$ of vendor budget | [1,5], [5,10], **[10, 20]**, [20, 30], [30, 40], [40, 50] |
| range $[r^-, r^+]$ of areas of vendors (%) | [1, 2], **[2, 3]**, [3, 4], [4, 5] |
| range $[a^-, a^+]$ of customer capacities | **[1, 4]**, [1, 6], [1, 8], [1, 10] |
| number, $m$, of customers | 5K, **10K**, 20K, 50K, 100K |
| number, $n$, of vendors | 300, **500**, 800, 1K, 2K |

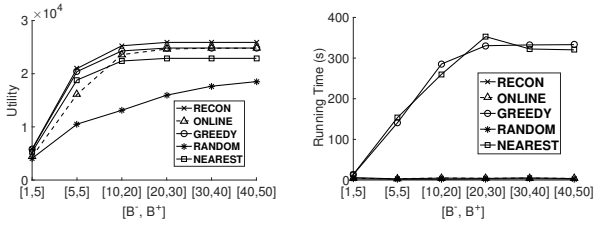
### A. Experimental Methodology

**Real/Synthetic Data Sets.** For real data sets, we used one check-in data set of Foursquare [31], 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. According to the discussion in Section II-A, we use the category to indicate the tags and estimate the interests/similarities of customers and vendors for the tags with the methods in Section II-A. 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 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 checkins). According to a report [5] 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 effectivenesses of ad types, respectively.

For synthetic data sets, we generate customers and vendors as follows. We randomly produce locations of customers and vendors in a 2D data space $[0, 1]^2$, following Gaussian $\mathcal{N}(0.5, 1^2)$ and Uniform distributions, respectively. As the specific timestamps of the customers will not affect the results of our online algorithm (only the orders of the customers affect the online algorithm), we use the orders of the customers to indicate their timestamps.

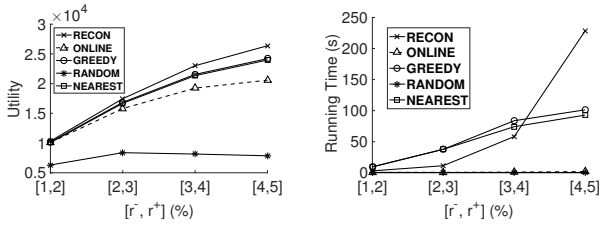
For both real and synthetic data sets, we simulate the budget $B_j$ of each vendor $v_i$ with Gaussian distribution $\mathcal{N}(\frac{B^- + B^+}{2}, (B^+ - B^-)^2)$ within range $[B^-, B^+]$, for $0 < B^- \leq B^+ < 1$. Similarly, we can generate the radius values of the range where vendor $v_j$ wants to post ads and the capacities of customers with Gaussian distributions within the range of the values shown in Table IV.

**Measures and Competitors.** We evaluate the effectiveness and efficiency of our MUAA processing approaches (both offline and online algorithms), in terms of the overall utility




(a) Utility Value (b) Running Time

Fig. 3. Effect of the Range $[B^-, B^+]$ of Budgets.




(a) Utility Value (b) Running Time

Fig. 4. Effect of the range $[r^-, r^+]$ of areas of vendors.

score and the CPU time. Specifically, the utility of an ad instance is defined in Equation (4), which can measure the quality of the assignment strategy, and the CPU time is given by the average time cost of performing MUAA assignment for a single customer.

In our MUAA problem, w.r.t. the effectiveness, we compare our MUAA approaches, including the *reconciliation* (RECON) and the *online adaptive factor-aware* (ONLINE) algorithms, 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 "currently best ad instance" that has the currently highest budget efficiency), and a nearest-neighbor (NEAREST) method (which greedily assigns the ads of the nearest vendors to a customer when he/she appears in the system). Specifically, The RECON algorithm first uses existing Linear Programming solver to solve the single-vendor problems, then reconciles the violations of the capacity constraints of the customers. To support the real world situation, where customers come in a stream fashion, the ONLINE algorithm decides to push or not to push ads to a newly incoming customers with no knowledge of the future customer. In particular, ONLINE algorithm only pushes ads to customers who have higher budget efficiencies than the values calculated by the adaptive threshold function shown in Section IV. Since RANDOM and NEAREST do not take into account the utility of ads, they are expected to achieve worse overall utility than our MUAA approaches (although they are expected to be faster than MUAA approaches). We prove the MUAA problem is NP-hard in Section II-E, and prove that the approximation ratio of RECON is $(1-\epsilon)\cdot\theta$ and the competitive ratio of ONLINE is $\frac{\ln(g)+1}{\theta}$ for $g > e$, where $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$, $n_i^c$ is the larger value between the number of valid vendors and the capacity $a_i$ of customer $u_i$, and $e$ is the base of the natural logarithm.

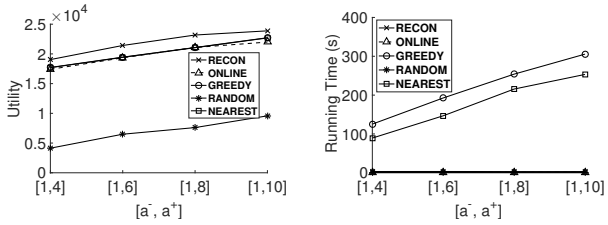
**Experimental Settings.** Table IV shows our experimental settings, where the default values of parameters are in bold font. In subsequent experiments, each time we vary one parameter, while setting others to their default values. All our experiments were run on an Intel Xeon X5675 CPU @3.07 GHZ with 32 GB RAM in Java.

### *B. Experiments on Real Data*

In this section, we show the effect of the range $[B^-, B^+]$ of the vendor budgets, the range $[r^-, r^+]$ of the radius of the vendors' valid areas, and the range $[a^-, a^+]$ of the customer capacities.

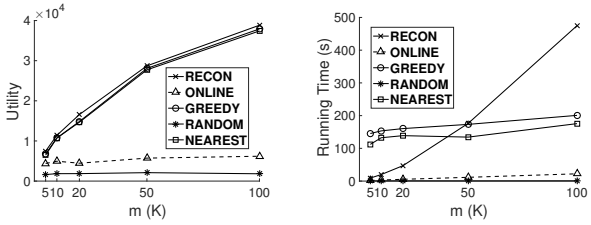
**Effect of the range $[B^-, B^+]$ of vendor budgets .** Figure 3 illustrates the experimental results on different ranges, $[B^-, B^+]$, of the vendor budgets $B_j$ from [1,5] to [40, 50]. In Figure 3(a), the total utility values of all the tested approaches improve when the value range of the vendor 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. Our two approaches can always achieve higher overall utility values compared to the RANDOM approach. The overall utility values of GREEDY and NEAREST are always lower than that of RECON. GREEDY and NEAREST can achieve higher utility values than ONLINE when the budgets are low, while their utility values are close to each other when range the budgets are higher than [20, 30]. As shown in Figure 3(b), the running times of ONLINE and RANDOM are always lower than 10 seconds, but that of GREEDY, NEAREST and RECON grow with the increase of the average budgets. For GREEDY and NEAREST, the reason is that, the number of iterations grows to use up the improved budgets, which requires more total running times. 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. We can see that GREEDY and NEAREST use more time than other tested algorithms. The vendors provide the budgets to promote their businesses. Higher budgets can result in higher utility values, which confirms our intuitive expectation.

**Effect of the range $[r^-, r^+]$ of areas of vendors.** Figure 4 reports the effect of the range $[r^-, r^+]$ of the radius of the vendors' valid areas over the real data set, which grows from [0.01, 0.02] to [0.04, 0.05]. As shown in Figure 4(a), when the radii of the valid areas increase, the overall utility values of our two proposed approaches improve while that of RANDOM will first increases, and then decreases. For our proposed two approaches (RECON and ONLINE), the improvement of the utility values is due to more customer-and-vendor pairs will become valid for larger radii. When the radii increase at the beginning more short-distance customer-and-vendor pairs can be used, which has a higher probability to have high utility ad instances. Later, when the radii become larger, the newly




(a) Utility Value (b) Running Time

Fig. 5. Effect of range $[a^-, a^+]$ of customer capacities.




(a) Utility Value (b) Running Time

Fig. 6. Effect of the number $m$ of customers.

valid customer-and-vendor pairs have a higher probability to have low utility ad instances. However, as RANDOM just randomly selects one ad instance at each time, its utility value increases at the beginning and then drops. We can still find that our ONLINE can achieve much higher utility values than RANDOM, and not surprisingly three offline algorithms, GREEDY, NEAREST and RECON, can beat ONLINE. In Figure 4(b), the range of $[r^-, r^+]$ affects ONLINE and RANDOM very little, and they always run fast. The running times of GREEDY and NEAREST increase linearly w.r.t. $r$. However, the running times of RECON increase exponentiallyas the Linear Programming solver [3] is sensitive to the problem size, which becomes larger when the range $[r^-, r^+]$ grows.

**Effect of the range $[a^-, a^+]$ of customer capacities.** Figure 5 presents the experimental results with different upper bounds of the customer capacities from [1,4] to [1,10]. As the capacities of customers only affects the results when there are more vendors, we select 5,000 vendors and 500 customers to test the effect of the upper bounds of the customer capacities while other parameters are in their default value. In Figure 5(a), we can see that the more ads one customer allows to receive, the higher overall utility values the tested approaches can achieve. This is because when the capacities of customers improve, we can push more ad instances to customers such that the overall utility values increase. RECON is still the best w.r.t. the overall utility value. ONLINE is not as good as GREEDY and NEAREST but much better than RANDOM. In Figure 5(b), the running times of RECON, ONLINE and RANDOM are always low while that of GREEDY and NEAREST increases with the improvement of the upper bound of the capacities of customers. For RECON, larger capacities will not affect the problem size of each single-vendor problem, furthermore, fewer violations of the capacity constraints of customers reduce the reconciling cost. Thus, the running time of RECON will not increase. However, for GREEDY and NEAREST, larger capacities cause the valid ad instances to increase, which requires GREEDY and NEAREST to run more iterations, and thus spend more time.

### C. Experiments on Synthetic Data

In this subsection, we first show the effectiveness of our approaches compared with the optimal result on a small scale dataset with 30 vendors and 1,000 customers, then we test the scalability and effectiveness of our two MUAA approaches, RECON and ONLINE, compared with GREEDY, NEAREST and RANDOM on synthetic data sets by varying the number $m$ of customers and the number $n$ of vendors.

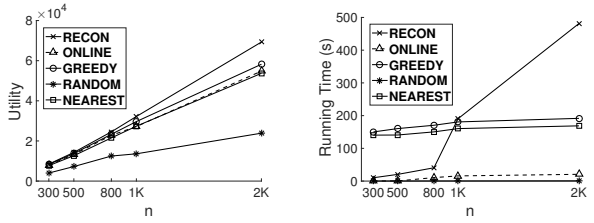
**Results on a Small MUAA Instance.** To show the effectiveness, we test all the approaches on a small scale MUAA problem instance with only 30 vendors and 1,000 customers. In Table V, we use LP-Solver [3] to solve the problem and get the optimal (OPT) result. The OPT achieves a total utility score of 248.653, but uses almost 2 hours to solve this small MUAA instance, which cannot be used to solve the real MUAA instances with thousands of vendors and customers. Our MUAA approaches RECON and ONLINE can achieve close utility scores but only use less than 1/4 seconds compared to the optimal result.

TABLE V
RESULTS OF SMALL MUAA INSTANCE.

| Algorithms | Utility Score | Running Time (s) |
|---|---|---|
| RECON | 226.456 | 0.235 |
| ONLINE | 217.905 | 0.129 |
| GREEDY | 224.128 | 0.330 |
| RANDOM | 147.424 | 0.028 |
| NEAREST | 223.456 | 0.338 |
| OPT | 248.653 | 6564.939 |

**Effect of the number $m$ of customers.** Figure 6 shows the effect of the number $m$ of the customers, by varying $m$ from 4,000 to 100,000. As shown in Figure 6(a), when the number of customers increases, our three MUAA approaches can obtain higher overall utilities, as more customers are available to be pushed with more ads. However, RANDOM just achieves similar utility values no matter what $m$ is, as RANDOM does not choose "good" ad instances from all the possible ad instances. Thus, when the budgets and radii of vendors do not change, the results of RANDOM will also be unchanged. In Figure 6(b), the running times of GREEDY, NEAREST, ONLINE and RANDOM grow linearly w.r.t. the number of customers while that of RECON grows exponentially, as the running time of the existing Linear Programming (LP) solver is very sensitive to the problem size [3]. When the problem size gets larger, the LP solver needs more iterations to obtain a result that is very close to the optimal result (i.e., 1-$\varepsilon$ approximation ratio), which makes the time to solve each single-vendor problem increase dramatically. As the running times of RECON grow faster than that of GREEDY and NEAREST when the number of customers increases, RECON is faster than GREEDY and NEAREST when $m$ is small and it is the slowest when $m$ is larger than 50K.

**Effect of the number $n$ of vendors.** Figure 7 illustrates the effect of the number $n$ of the vendors, by varying $n$ from 300 to 2,000. As shown in Figure 7(a), when $n$ increases, all the tested algorithms can achieve higher utility values as more vendors need higher total budgets to support more ad instances. RECON still achieves the highest utility values. GREEDY performs close to RECON. The results of ONLINE




(a) Utility Value (b) Running Time

Fig. 7. Effect of the number $n$ of vendors.

are higher than RANDOM but lower than that of RECON and GREEDY and NEAREST. In Figure 7(b), the running time of RECON grows dramatically w.r.t. $n$, as more vendors mean that more single-vendor problems need to be solved by RECON. The running times of GREEDY and NEAREST grow slightly. The reason is that the number of customers does not change, and then the iterations of GREEDY and NEAREST do not change. The increasing costs of GREEDY and NEAREST are from the increasing cost of picking out one best instance in each iteration from more candidate customer-and-vendor pairs when $n$ gets larger. ONLINE needs slightly more time when $n$ increases, as for each customer there are more vendors to consider (only the vendors that the customer is in their effective areas affect the process of the customer). ONLINE is slightly slower than RANDOM but much faster than GREEDY, NEAREST and RECON.

In summary, over both real and synthetic data sets, our ONLINE approach can achieve results with close total utility values compared with those of offline approaches. Our two proposed approaches, RECON and ONLINE, achieve much higher utility values compared with RANDOM. However, the total running time of ONLINE are very low, which is close to that of RANDOM. More importantly, ONLINE is designed to work in online scenarios where customers keep on coming and we have no knowledge of the future customers. ONLINE can respond to each incoming customer very quickly in less than 1 second even when there are 20K vendors in the system, which can offer good user experiences to customers.

## VI. Related Work

Recently, the popularity of GPS-equipped smart devices brings convenience to the daily life of people. The users can easily access the information of the POIs (e.g., shops, restaurants and cinemas) around them, which can help the users plan their trips, order dinners and book tickets online. The historical records reveal the interests of the users, which enable the advertising systems to accurately push ads to each individual to improve the effects of the ads. As a result, nowadays, the location-based advertising attracts much attention from both academia and industry. In POI recommendation system, existing studies [32], [33], [34], [27] focus on estimating the customers' preference towards the POIs to recommend attracting and interesting places to customers. Another similar problem is the spatial matching problem [28], which assigns a set of customer to their closest service providers such that the total distance is minimized subject to the constraint of the capacity of each service provider. Then, [18] studies on fast maintaining the optimal matching (the total distance is minimized) in dynamic setting where customers are moving around. However, in our MUAA problem, we need to maximize the overall utility of spatial ad instances subject to the constraints of the budget and active ranges of vendors and capacities of customers. The main difference is that in our MUAA problem utility of a given ad instance depends on the preference and the distance of the related customer-and-vendor pair, and the utility effectiveness of the ad type while the spatial matching problem only considers the distance. In addition, in our MUAA problem, ads with different types may have different costs, which is not considered in spatial matching problem. Thus, we cannot use the existing methods to solve our MUAA problem directly.

Some prior studies investigate the customer attitudes towards location-based advertising (LBA) [12], [24]. Specifically, the researchers investigate the attitudes of customers towards LBA and the relationship between the attitude and the behavior through surveys. Their results show that: (1) customers generally have negative attitudes toward LBA unless they have agreed to receive it explicitly; (2) customers' attitude can directly affect their behavior (e.g., visiting the shop of the received ad). These studies reveal some location-based advertising types (e.g., Text-Link) have low budget efficiency. To successfully apply LBA and improve the revenues of vendors, well developed *business models* are necessary. Existing studies [14], [20] analyze the factors related to designing business models for LBA. In [25], the authors report that the pulling-based LBA is more effective, which indicates the online property of LBA (i.e., in general the servers should return related LBA ads to customers very quickly when they are querying interested nearby POIs).

In [30], the authors propose one novel location-based advertising framework on temporary social networks to select vendors as advertising sources for mobile customers under the constraint of the capacity of the customers. To query the relevant vendors quickly, they propose an algorithm to track the conservative safe region for moving customers to address the continuous vendor selection problem, which only fires a recalculation process when the relevant vendors have changed for a given customer. In our work, we consider the customer will come to the location-based advertising system randomly. When a customer comes, we need to report a set of "best" ad instances to the customer under his/her capacity constraint. Most importantly, we consider each vendor has a limited budget, and thus propose a novel online adaptive factor-aware (O-AFA) algorithm, which will only select the ad instances having high budget efficiency when the remaining budget of the vendors becomes less. We prove that the O-AFA algorithm has a competitive ratio of $\frac{\ln(g)+1}{\theta}$, $g > e$, where $e$ is the base of the natural logarithm, $\theta = \min(\frac{a_1}{n_1^c}, \frac{a_2}{n_2^c}, ..., \frac{a_m}{n_m^c})$, and $n_i^c$ is the larger value between the number of valid vendors and the capacity $a_i$ of customer $u_i$. Besides that, we also propose one approximation offline algorithm, namely the reconciliation, to fast estimate the upper bound of the maximum utility for a given MUAA problem instance, which has an approximation ratio of $(1 - \epsilon) \cdot \theta$.

Privacy is another important concern for location-based advertising, as customers need to report their locations to the server, which may release some sensitive location data. Some existing studies focus on protecting the users' privacy [8], [9].

## VII. 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 by reducing it from the 0-1 Knapsack problem. We design one offline algorithm, namely the reconcile algorithm, 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 $u_i$. In addition, we also propose one online algorithm, namely online adaptive factor-aware algorithm (O-AFA), to return a set of "best" ad instances to a customer when he/she enters the location-based advertising system. We prove that O-AFA has a competitive ratio of $\frac{\ln(g)+1}{\theta}$, $g > e$, where $e$ is the base of the natural logarithm. Extensive experiments have been conducted to confirm the efficiency and effectiveness of our offline and online algorithms on both real and synthetic data sets.

## References


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

[2] [online] Foursquare API. https://developer.foursquare.com/docs/venues/categories.

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

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

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

[6] [online] xAd. http://www.xad.com.

[7] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[8] P. Barwise and C. Strong. Permission-based mobile advertising. *Journal of interactive Marketing*, 16(1):14–24, 2002.

[9] E. Beatrix Cleff. Privacy issues in mobile advertising. *International Review of Law Computers and Technology*, 21(3):225–236, 2007.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[11] P. Cheng, X. Lian, L. Chen, and S. Liu. Maximizing the utility in location-based mobile advertising (technical report). http://haidaoxiaofei.github.io/d/spatialAD.pdf, 2017.

[12] H. K. Chowdhury, N. Parvin, C. Weitenberner, and M. Becker. Consumer attitude toward mobile advertising in an emerging market: An empirical study. *International Journal of Mobile Marketing*, 1(2), 2006.

[13] G. Dantzig. *Linear programming and extensions*. Princeton university press, 2016.

[14] S. Dhar and U. Varshney. Challenges and business models for mobile location-based services and advertising. *Communications of the ACM*, 54(5):121–128, 2011.

[15] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

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

[17] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[18] K. Mouratidis, N. Mamoulis, et al. Continuous spatial assignment of moving users. *The VLDB JournalThe International Journal on Very Large Data Bases*, 19(2):141–160, 2010.

[19] F. Pozzi, T. Di Matteo, and T. Aste. Exponential smoothing weighted correlations. *The European Physical Journal B-Condensed Matter and Complex Systems*, 85(6):1–21, 2012.

[20] J. Salo. The success factors of mobile advertising value chain. *Business Review*, 4(1):93–97, 2004.

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

[22] T. Song, T. Yongxin, L. Wang, J. She, and et al. Trichromatic online matching in real-time spatial crowdsourcing. *ICDE*, 2017.

[23] Y. Tong, J. She, B. Ding, L. Chen, and et al. Online minimum matching in real-time spatial data: experiments and analysis. *VLDB*, 2016.

[24] M. M. Tsang, S.-C. Ho, and T.-P. Liang. Consumer attitudes toward mobile advertising: An empirical study. *International journal of electronic commerce*, 8(3):65–78, 2004.

[25] R. Unni and R. Harmon. Perceived effectiveness of push vs. pull mobile location based advertising. *Journal of Interactive advertising*, 7(2):28–40, 2007.

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

[27] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou. St-sage: A spatial-temporal sparse additive generative model for spatial item recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3):48, 2017.

[28] R. C.-W. Wong, Y. Tao, A. W.-C. Fu, and X. Xiao. On efficient spatial matching. In *Proceedings of the 33rd international conference on Very large data bases*, pages 579–590. VLDB Endowment, 2007.

[29] D. Wu, M. L. Yiu, and C. S. Jensen. Moving spatial keyword queries: Formulation, methods, and analysis. *ACM Transactions on Database Systems (TODS)*, 38(1):7, 2013.

[30] W. Xu, C.-Y. Chow, and J.-D. Zhang. Calba: capacity-aware location-based advertising in temporary social networks. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 364–373. ACM, 2013.

[31] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 45(1):129–142, 2015.

[32] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems (TOIS)*, 35(2):11, 2016.

[33] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen. Adapting to user interest drift for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2566–2581, 2016.

[34] H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq. Joint modeling of user check-in behaviors for point-of-interest recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1631–1640. ACM, 2015.

[35] Y. Zhou, D. Chakrabarty, and R. Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *International Workshop on Internet and Network Economics*, pages 566–576. Springer, 2008.

[36] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 406–415. ACM, 2004.

## APPENDIX

### A. Proof of Theorem II.1

*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 $W$, the 0-1 knapsack problem is to 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 $\sum_{\forall\langle u_0,v_0,\tau_i\rangle\in\mathbb{I}} c_i \leq B_0$.

Given this mapping, we can show that the 0-1 knapsack problem instance can be solved, if and only if the transformed MUAA problem can be solved.

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 [26], MUAA is also NP-hard. ■

### B. Proof of Theorem III.1

**Theorem A.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:

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

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:

$$\lambda(\mathbb{I}) \geq \sum_{u_i\in U'}\sum_{\langle u_i,v_j,\tau_k\rangle\in\mathbb{I}(u_i)} \lambda_{ijk}. \quad (10)$$

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 (10) 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:

$$\begin{aligned}\lambda(\mathbb{I}) &\geq \min_{u_i\in U'}(\frac{a_i}{n_i^c}) \cdot \sum_{u_i\in U'}\sum_{j=1}^n \sum_{\langle u_i,v_j,\tau_k\rangle\in\mathbb{I}_j(u_i)} \lambda_{ijk} \\ &= \min_{u_i\in U}(\frac{a_i}{n_i^c}) \cdot \sum_{j=1}^n \lambda(\mathbb{I}_j) \quad (11)\end{aligned}$$

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 [16] as $\mathbb{I}_j$. As the LP-relaxation algorithm for the multi-choice knapsack problem is a $\varepsilon$-approximate algorithm, we have $\lambda(\mathbb{I}_j) \geq (1-\varepsilon)\cdot\lambda(\mathbb{I}_j^*)$. Then, based on the Inequality (11), we have:

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

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. ■

### C. Proof of Theorem IV.1

**Theorem A.2.** *The online algorithm, O-AFA, has a competitive ratio of* $\frac{1}{\theta}\cdot\big(\frac{\phi(\delta_{max})}{h\cdot\gamma_{min}+\Phi(\delta_{max})-\Phi(h)}+1\big)$.

*Proof:* For a given MUAA problem instance $\mathbb{M}$, let $\mathbb{I}$ and $\mathbb{I}^*$ ($\mathbb{I}_j$ and $\mathbb{I}_j^*$) be the instance sets selected by the O-AFA algorithm and the optimum (of vendor $v_j$), respectively. For each vendor $v_j$, we denote its used budget ratio as $\delta_j^{(i)}$ when customer $u_i$ appears and as $\delta_j$ when the O-AFA terminates. Then, $\delta_{max} = \max_{v_j\in V}\delta_j$.

From the view of a single vendor $v_j$, if any instance $\langle u_i, v_j, \tau_x\rangle$ is not picked by the O-AFA algorithm, the reason can be 1) it has the budget efficiency $\lambda_{ijx} \leq \phi(\delta_j^{(i)})$; 2) it has the budget efficiency larger than $\lambda_{ijx} > \phi(\delta_j^{(i)})$ but $\lambda_{ijx} \leq \lambda_{ijk}$ where $\lambda_{ijk}$ is the efficiency of the selected instance for customer $u_i$. Let $\Delta_{ij}$ be the minimum value such that for the given customer $u_i$, vendor $v_j$ and the selected advertising instance $\langle u_i, v_j, \tau_k\rangle$, any other instance $\langle u_i, v_j, \tau_x\rangle$ has $\lambda_{ijx} \leq \phi(\delta_j^{(i)}) + \Delta_{ij}$ and $\lambda_{ijk} > \phi(\delta_j^{(i)}) + \Delta_{ij}$.

Since $\phi(\delta)$ is a monotone increasing function of $\delta$, we have

$$\lambda(\mathbb{I}_j^*) - \sum_i \Delta_{ij} \leq \lambda(\mathbb{I}_j \cap \mathbb{I}_j^*) + \phi(\delta_j)(B_j - C_j),$$

where $C_j = \sum_{\langle u_i,v_j,\tau_k\rangle\in\mathbb{I}_j\cap\mathbb{I}_j^*} c_k$.

Next, as $\sum_i \Delta_{ij} < \lambda(\mathbb{I}_j)$ we can have another inequality below:

$$\frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)} \leq \frac{\lambda(\mathbb{I}_j \cap \mathbb{I}_j^*) + \phi(\delta_j)(B_j - C_j)}{\lambda(\mathbb{I}_j \cap \mathbb{I}_j^*) + \lambda(\mathbb{I}_j/\mathbb{I}_j^*)} + 1. \quad (12)$$

Here, we denote the used budget ratio of a vendor $v_j$ is $\delta_j^{(i)}$ when it selects the ad instance $\langle u_i, v_j, \tau_k \rangle$. Thus, we have $\gamma_{ijk} \geq \phi(\delta_j^{(i)})$. Then, we have:

$$\lambda(\mathbb{I}_j \cap \mathbb{I}_j^*) \geq \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k, \quad and$$

$$\lambda(\mathbb{I}_j / \mathbb{I}_j^*) \geq \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j / \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k.$$

As $\lambda(\mathbb{I}_j^*) \geq \lambda(\mathbb{I}_j)$, inequality 12 can be rewritten as below:

$$\begin{aligned} \frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)} &\leq \frac{\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k + \phi(\delta_j)(B_j - C_j)}{\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k + \lambda(\mathbb{I}_j / \mathbb{I}_j^*)} + 1 \\ &\leq \frac{\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k + \phi(\delta_j)(B_j - C_j)}{\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j} \phi(\delta_j^{(i)}) \cdot c_k} + 1. \end{aligned}$$

As $\phi(\delta)$ is monotone increasing for $\delta$, $\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j^{(i)}) \cdot c_k \leq \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j \cap \mathbb{I}_j^*} \phi(\delta_j) \cdot c_k = \phi(\delta_j) \cdot C_j$. The above inequality implies that

$$\frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)} \leq \frac{\phi(\delta_j)}{\sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j} \phi(\delta_j^{(i)}) \cdot \frac{c_k}{B_j}} + 1. \tag{13}$$

As the cost of any ad instance is much smaller than the budget of vendor $v_j$, we can have an integration below to approximate the summation:

$$\begin{aligned} \sum_{\langle u_i, v_j, \tau_k \rangle \in \mathbb{I}_j} \phi(\delta_j^{(i)}) \cdot \frac{c_k}{B_j} &\approx \int_0^{\delta_j} \phi(\delta) d\delta \\ &= \int_0^{h} \phi(\delta) d\delta + \int_h^{\delta_j} \phi(\delta) d\delta \\ &\leq h\gamma_{min} + \Phi(\delta_j) - \Phi(h). \end{aligned} \tag{14}$$

Then, we combine inequalities 13 and 14, we have

$$\frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)} \leq \frac{\phi(\delta_j)}{h \cdot \gamma_{min} + \Phi(\delta_j) - \Phi(h)} + 1.$$

Next, we combine the analysis of all the vendors together. We notice that for each customer $u_i$ we only select the ad instances with top-$a_i$ highest budget efficiency values. On each customer $u_i$ we at most reduce the his/her influence value to $\frac{a_i}{n_i^c}$ of the summation of all the potential ad instances, where $n_i^c$ is the larger value between the number of valid vendors and the capacity $a_i$ of customer $u_i$. In addition, as $\phi(\delta)$ is larger than 0 and monotone increasing on $\delta$, thus $\Phi(\delta)$ is also increasing on $\delta$. Since $\frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)} \geq 1$ , when $\delta_j$ increase, $\frac{\lambda(\mathbb{I}_j^*)}{\lambda(\mathbb{I}_j)}$ also increases. Then we have,

$$\begin{aligned} \frac{\lambda(\mathbb{I}^*)}{\lambda(\mathbb{I})} &\leq \frac{1}{\theta} \cdot \frac{\sum \lambda(\mathbb{I}_j^*)}{\sum \lambda(\mathbb{I}_j)} \\ &\leq \frac{1}{\theta} \cdot \Big( \frac{\phi(\delta_{max})}{h \cdot \gamma_{min} + \Phi(\delta_{max}) - \Phi(h)} + 1 \Big). \end{aligned} \tag{15}$$

Thus, this completes the proof. ■

### D. Proof of Corollary IV.1

**Corollary A.1.** *The online algorithm, O-AFA, has a competitive ratio of $\frac{\ln(g)+1}{\theta}$ when $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^{\delta}$ and $g > e$.*

*Proof:* The corollary can be proved by plugging the definition of $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^{\delta}$ in Inequality (15).

Specifically, when $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^{\delta}$, $\Phi(\delta) = \frac{\gamma_{min}}{e} \cdot \frac{g^{\delta}}{\ln(g)}$, and $h = \frac{1}{\ln(g)}$. Then Inequality 15 can be rewritten as below:

$$\begin{aligned} \frac{\lambda(\mathbb{I}^*)}{\lambda(\mathbb{I})} &\leq \frac{1}{\theta} \cdot \Big( \frac{\phi(\delta_{max})}{h \cdot \gamma_{min} + \Phi(\delta_{max}) - \Phi(h)} + 1 \Big) \\ &= \frac{1}{\theta} \cdot \Big( \frac{\frac{\gamma_{min}}{e} \cdot g^{\delta_{max}}}{\frac{\gamma_{min}}{\ln(g)} + \frac{\gamma_{min}}{e} \cdot \frac{g^{\delta_{max}}}{\ln(g)} - \frac{\gamma_{min}}{\ln(g)}} + 1 \Big) \\ &= \frac{\ln(g) + 1}{\theta} \end{aligned} \tag{16}$$

According to the definition of competitive ratio in Section II-D, we have the conclusion that the competitive ratio of our online algorithm O-AFA is $\frac{\ln(g)+1}{\theta}$ when $\phi(\delta) = \frac{\gamma_{min}}{e} \cdot g^{\delta}$ and $g > e$. ■