Crowd Foraging: A QoS-oriented Self-organized Mobile Crowdsourcing Framework over Opportunistic Networks

Lingjun Pu, Xu Chen, Member, IEEE, Jingdong Xu, and Xiaoming Fu, Senior Member, IEEE.

Abstract—Recent years have witnessed the proliferation of mobile crowdsourcing that brings a new opportunity to leverage human intelligence and movement behaviors to wider application areas. In parallel with the development of online centralized platforms, we look into the realization of self-organized mobile crowdsourcing drawing on opportunistic networks, and propose the Crowd Foraging framework, in which a mobile task requester can proactively recruit a massive crowd of opportunistic encountered mobile workers in real-time for quick and high-quality results. We present a comprehensive framework model that fully integrates human behavior factors for modeling task profile, worker arrival and work ability, and then introduce a service quality concept to indicate the expected service gain that a requester can enjoy when she recruits an arrival worker by jointly taking into account work ability of workers as well as timeliness and reward of tasks. Further, we formulate a sequential worker recruitment problem as an online multiple stopping problem to maximize the expected sum of service quality, and accordingly derive an optimal worker recruitment policy through the dynamic programming principle, which exhibits a nice threshold based structure. We provide data-driven case studies to validate the assumptions used in the policy design, and conduct extensive trace-driven numerical evaluations which demonstrate that our policy can achieve superior performance (e.g., improve more than 30% performance over classic policies). Besides, our Android prototype shows that the Crowd Foraging framework is cost-efficient, such as requiring less than 7 seconds and 6 Joule in terms of time and energy consumption for the optimal threshold calculation in our policy in most cases.

Index Terms—Mobile Crowdsourcing, Service Quality, Worker Recruitment, Opportunistic Networks.

I. INTRODUCTION

Recent advances in computing, communicating and sensing capabilities of mobile devices along with their widespread use in daily life lead to mobile crowdsourcing, an emerging task-solving paradigm where a crowd of human workers are recruited to solve complex tasks such as data-collection, solution-finding and opinion-seeking by using their mobile devices. Realizing the great potential, many web-based platforms such as Amazon mTurk and Crowdflower as well as popular mobile applications including Stereopublic and Placemeter have been developed, in which requesters define their tasks, and human workers with diverse expertise levels, interest preferences and movement behaviors execute them in exchange of monetary rewards.

However, two common issues restrict the development of those online mobile crowdsourcing systems. First, they neither proactively recruit workers nor provide an advertising mechanism to inform workers for newly created tasks but instead post the tasks online and passively wait for the workers to participate [1]. As such, it is difficult for newly created tasks to attract enough workers in a short time. For example, less than 15% tasks can be completed within one hour in Amazon mTurk [2]. Second, they are centralized, which is not flexible and scalable especially when mobile crowdsourcing reaches maturity [3], [4]. For example, if many mobile users launch image/audio transcription tasks (a very popular task category in existing online systems [5]), the online systems have to consume a large volume of storage resources to maintain the tasks, and the amount of data traffic introduced by the tasks (e.g., the uploaded images to the online systems) is not negligible for cellular networks [6].

To complement online mobile crowdsourcing services, in this paper we advocate Crowd Foraging, a QoS-oriented self-organized mobile crowdsourcing framework, where a mobile user can self-organize her task crowdsourcing in a proactive manner by leveraging a massive crowd of opportunistically encountered mobile human workers in realtime. Its main rationality is four-fold. First, opportunistic user encounters are prevalent and sufficient in daily life [7], which offers plenty of opportunities to exploit nearby human intelligence for task solving [1], [8], [9]. Second, many mobile crowdsourcing tasks require local knowledge and information (e.g., location-based content sensing [3], [4], [6] and content transmission [10], [11], [12]), and hence nearby human workers are more adept at executing them than the online workers. Third, D2D communications such as Bluetooth, WiFi-direct and LTE-D2D are promising to replenish traditional cellular communications in terms of user throughput increase, cellular traffic reduction and network coverage extension [13]. At last, this framework shares the similar spirit with the emerging paradigm “cyber foraging” over opportunistic networks, such that mobile users opportunistically exploit nearby device resources to facilitate their computational task processing [14], [15], [16].

Lingjun Pu and Jingdong Xu are with the College of Computer and Control Engineering, Nankai University, Tianjin 300071, China. (e-mail: pulingjun@nankai.edu.cn; xujd@nankai.edu.cn).

Xu Chen was with the Institute of Computer Science, University of Göttingen, Göttingen 37073, Germany. He is now with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China (e-mail: chenxu35@mail.sysu.edu.cn).

Xiaoming Fu is with the Institute of Computer Science, University of Göttingen, Göttingen 37073, Germany. (e-mail: fu@cs.uni-goettingen.de).

A preliminary version of this work appeared in the Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 10–15 April 2016, San Francisco, CA, USA.
In this paper, we tackle these challenges by formulating the worker recruitment problem as an online multiple stopping problem and derive an optimal online worker recruitment policy accordingly. Our main contributions are summarized as follows:

- **We propose Crowd Foraging**, a novel mobile crowdsourcing framework where a mobile user can self-organize her task crowdsourcing in a proactive manner by leveraging a massive crowd of one-hop human workers in realtime. We present a comprehensive framework model that fully integrates human behavior factors for modeling task profile, worker arrival and work ability. In addition, we introduce a service quality concept to indicate the expected service gain that a requester can obtain when she recruits an arrived worker, by jointly taking into account work ability, task timeliness and task reward.

- **We study the worker recruitment problem for the framework and propose an online multiple stopping problem formulation accordingly.** Then, we derive an optimal online worker recruitment policy to achieve the maximum expected sum of service quality (i.e., total QoS) by leveraging the principle of backward induction, and find that it has a nice threshold based structure (i.e., the non-increasing property in the time domain and in the required worker amount domain, respectively). Next, we show that the thresholds can be computed by solving a set of differential equations, and illustrate the threshold calculation for two typical cases. At last, we provide data-driven case studies to not only validate the assumptions used in the policy design but also illustrate how to apply the policy for worker recruitment in practice.

- **We evaluate our policy against several well-known policies in terms of trace-driven simulations.** Numerical results demonstrate that the total QoS achieved by our policy is 30–40% higher than that achieved by greedy, random and classic multiple secretary policies, and our policy can well adapt to mobile crowdsourcing tasks with realtime requirement as well as diverse network scenarios with different user amount. In addition, we implement a system prototype of the framework in Android platform to evaluate the practical overheads. Test results reveal that the prototype is lightweight.

### II. Framework Model and Problem Statement

In this section, we first introduce the specific model of the Crowd Foraging framework, and on that basis we discuss the online sequential worker recruitment problem.

#### A. Framework Model

**Task model**: A mobile user can open the Requester Mode in the framework (see Fig. 1 for illustration) to launch a mobile crowdsourcing task in the Task Generator. In this paper, we consider a general model widely used in online systems that a task \( i \) is described by a set of attributes, <Category, Description, Keywords, Amount, AllottedTime, Reward>:

- Category, represents the category of the task \( i \) such as location-based content sensing, image/audio transcription and event surveys [5];

...
(2) Description, indicates the specific demands of the task i. For instance, a description could be “please help me make geo-tag for 20 sight-seeing photos in this city and specify the tour route information”;

(3) Keywords, is a set of keywords that can reflect the characteristics of the task i. For instance, the keywords can be the specific locations in location-based content sensing tasks;

(4) Amount, represents the number of workers that the task requester wants to recruit for the task i in order to gain robust and high-quality service. In general, Amount, depends on the requester’s QoS requirement and the amount of her budget for the task [2], [17];

(5) AllottedTime, represents the maximum allowed execution time within which a recruited worker is required to return the results. During the task execution, the task requester and each recruited worker do not need to stay connected all the time. The worker can send the results to the requester through either cellular link directly or through D2D link when they encounter with each other again with the AllottedTime.

(6) Reward, is the monetary reward to a recruited worker when his results are returned within the AllottedTime. Similar to the online systems, we adopt a common “posted price model” where the workers are given a take-it-or-leave-it price offer1 [18].

Worker arrival model: A mobile user can open the Worker Mode in the framework to become a worker who wants to execute tasks in exchange of monetary rewards. We denote N as the set of workers in the framework. As far as user mobility is concerned, we assume that the inter-encounter2 time between any pair of a requester r and a worker j follows an Exponential distribution with parameter \( \lambda_{rj} \). Note that, this assumption has been validated by realistic dataset analyses [19] and widely used in many existing researches (e.g., [1], [4], [11], [19] and references therein). As a result, the inter-encounter time of a requester r encountering two consecutive workers also follows an Exponential distribution with parameter \( \lambda_r = \sum_{j \in N} \lambda_{rj} \) [20]. In other words, the arrival of workers to a requester r follows a Poisson process with arrival rate \( \lambda_r \). We should emphasize that, as proved in [20], this Poisson process also holds in a more general scenario where the inter-encounter time between a pair of users follows a mixture between Power law and Exponential distribution [21], [22]. Note that, the task requester r in our framework does not need to know the parameter \( \lambda_{rj} \) for each individual user j, but the information of the aggregate arrival rate \( \lambda_r \) is enough. In practice, the framework can predict the parameter by calculating the average number of encountered users per time unit in terms of the historical user encounter records in a given backoff time window3.

The Poisson process models the arrival of workers to a task requester, while it does not capture whether an arrived worker is valid. Here, we say a “valid” worker meaning that the worker accepts the task reward and is confident to return the results within the allotted time (otherwise the requester will not pay the reward). For the former factor, we consider that each worker will accept the task reward with a probability \( p_{ac} \), which is a function of the posted reward value (i.e., Reward) \([23],[24]\). For the latter factor, we denote the probability4 that the arrived worker is capable to finish the task within the allotted time as \( p_{re} \). In practice, these probabilities can be learned from historical statistical data or requester’s past experiences by following the approaches in \([2],[23],[24]\). Based on these two independent probabilities, a Bernoulli process can be used to model whether an arrived worker is valid. Suppose that there are n arrived workers in a time interval and the number of valid workers follows a Binomial distribution \( B(n, p_{ac}p_{re}) \).

In the end, the valid worker arrival model to a task requester \( r \) is a composition of Poisson process and Binomial distribution, which is also a Poisson process with a composite parameter \( \beta_r = \lambda_r \cdot p_{ac}p_{re} \) [25]. Therefore, the number of valid workers arrived to a requester \( r \) within a time interval \([t, t + \Delta t]\) will follow a Poisson distribution with parameter \( \beta_r \Delta t \). That is to say, when \( \Delta t \to 0 \), the parameter \( \beta_r \Delta t \) can be viewed as the probability of exact one worker arrives at a time point \( t \) according to the property of Poisson distribution [25].

Work Ability model: For a mobile crowdsourcing task, we introduce the work ability to indicate how well the task is executed by a worker, which is modeled by the Worker Profiler associated with the task attributes. The common rationale is that human workers familiar with specific domains or locations (indicated by the task categories and keywords) are adept at solving their related tasks \([2]\). Suppose that there are C task categories and each category \( c \in C \) includes a fixed keyword space \( I^c \) with a maximum size \( K \). We define the following concepts.

Definition 1: The Worker Profiler of a worker \( j \) is described by a \( K \times C \) matrix. Each column \( c \) represents a rate vector \( P^c_j = [p^c_{j1}, p^c_{j2}, \ldots, p^c_{jK}]^T \), where \(^T\) indicates matrix transpose operation and \( p^c_{jk} \) denotes the familiarity degree of the worker \( j \) with respect to the k-th keyword in category \( c \). We define the following concepts.

Definition 2: Each task \( i \) is described by a set of keywords (i.e., Keywords\(_i\)) \( H_i \subseteq \bigcup \limits_{c \in C} I^c \) in a category \( c \) (i.e., Category\(_i\)) and their corresponding weights. A weight \( \omega_{ik} \) indicates the importance of a keyword \( k \) in describing the task \( i \). Therefore, the task \( i \) is formulated as a \( 1 \times K \) vector \( V_i = [v_{i1}, v_{i2}, \ldots, v_{iK}] \), where \( v_{ik} = \omega_{ik} \) if a keyword \( k \in H_i \) and \( v_{ik} = 0 \) otherwise. According to the above two definitions, the work ability of a worker \( j \) with respect to a task \( i \) belonging to a category \( c \)

1As an initial study, we assume the task reward is determined a priori by the task requester, and leave the optimal task reward selection to maximize the task requester’s total QoS in a future work.

2Here, we say that two mobile users “encounter” meaning that they move close and can conduct D2D communication.

3The valid period of a task can be viewed as the backoff time window since the framework only cares about the future arrived workers within the task valid period. In this sense, the aggregate worker arrival rate is actually task-specific.

4Although a valid worker is confident at executing the task in time, occasionally there may exist some uncontrolled risk such that the worker can not return the results in time (e.g., running out of battery power or network coverage). To hedge against such risk, the requester can increase the number of recruited workers (i.e., Amount\(_i\)), to achieve a robust service.
is given as follows:

$$q_{j,i} = V_i^c P_j^c = \sum_{k=1}^{K} v_{ik}^c p_{jk}^c. \quad (1)$$

We should emphasize that, this general model fits a variety of mobile crowdsourcing tasks. For example, for location-based services (e.g., location-based content sensing and content transmission), the keyword space will be a set of PoI locations, the rate vector in the Worker Profiler can be viewed as the visiting frequency of workers with respect to those locations, and the work ability indicates the probability of workers visiting the required locations; For content (e.g., image) transcription services, the keyword space will be a tag cloud (e.g., culture and landscape), the rate vector in the Worker Profiler can be viewed as the interest preference degree or expertise level of workers with respect to those tags, and the work ability describes the familiarity of workers with the specific domains. Without loss of generality, we assume that for a given task the work ability of workers follows a general distribution with the probability density function (PDF) $f(x)$ and the cumulative distribution function (CDF) $F(x) = \int_{-\infty}^{x} f(u)du$. In addition, we consider that the rate vector of each worker in the Worker Profiler is unknown to the task requester in advance. Hence, the task requester has to probe each arrived worker on the spot.

**Interaction between task requester and worker:** As illustrated in Fig. 1, the interaction contains a worker probe stage (steps (1), (2), (3)) and a task execution stage (steps (4), (5)). In the former stage, after a task is launched, the Worker Selector is then invoked to recruit workers in real-time. Specifically, the selector first obtains the task attributes from the Task Generator and send them to an arrived worker via D2D link; The worker receiving the task attributes decides whether to execute the task according to the Reward and the AllottedTime. If yes, the worker (actually the Worker Profiler) will calculate the work ability according to the equation (1), and sends it to the task requester via D2D link. Otherwise, a negative value will be sent to the task requester; With the value of work ability, the requester carries out a worker recruitment policy to decide whether to recruit that worker. If yes, the requester sends the specific contents of the task (e.g., images, videos) to the recruited worker via D2D link, and attaches the worker ID associated with his recruited task to the worker in the Result Collector;

In the latter stage, when a recruited worker finishes the task, he will send the results to the corresponding task requester via D2D link if he encounters with the requester again or via cellular link directly within the AllottedTime; When the Result Collector of the task requester receives the results from a worker in time, it will mark the worker as "completion" and grant the task reward to him. It is obvious that the worker recruitment policy is the core of the Crowd Foraging framework, and on the basis of the framework model we will discuss it in the following.

**B. Sequential Worker Recruitment Problem**

Due to users’ mobility in opportunistic networks, a task requester needs to decide whether to recruit an arrived worker once they encounter with each other. In general, a task requester expects high-quality and quick results with low monetary expenditure. Therefore, to facilitate the recruitment decision making, we introduce a service quality concept by jointly considering the work ability, task timeliness (i.e., the recruited time of workers$^5$) and task reward as follows.

$$R_i(t_n) = q_{n,i} \phi_i(t_n) - b_i, \quad (2)$$

where $b_i$ indicates the posted reward of the task $i$, $n$ indicates the $n$-th arrived worker, $t_n$ is the worker arriving time and $q_{n,i}$ is the corresponding work ability. Moreover, $\phi_i(t)$ is the task utility function, which indicates the surplus task value that a requester can achieve when she recruits a worker at a time $t$ to execute the task. In general, the function $\phi_i(t)$ should capture the real-time requirement and the waning interest of task requesters to their created tasks. Therefore, we specify $\phi_i(t)$ satisfying the following conditions:

- $\phi_i(0) = B_i$, the initial task utility.
- If $t_1 \leq t_2$, then $\phi_i(t_1) \geq \phi_i(t_2)$.
- If $t_1 < T_i \leq t_2$, then $\phi_i(T_i) > \phi_i(t_2) = 0$.

The first two conditions ensure that the function captures the non-increasing feature between task utility and time. The third condition supports the task utility function with deadline $T_i$ after which the task will be discarded by the task requester. In addition, we assume $\phi_i(t)$ is piecewise continuous and differentiable by time $t$ for ease of presentation. In this paper, we consider two typical task utility functions: the 1-0 step function for modeling the hard-deadline driven scenarios, and the exponential declining function for modeling the time discounting scenarios.

$$\phi_i(t) = \begin{cases} 
B_i & t < 0 \\
1 & 0 \leq t \leq T_i \\
0 & t > T_i 
\end{cases}$$

Fig. 2: Illustration of the sequential worker recruitment procedure.

Since a task requester requires to recruit Amount ($m_i$ for short) distinct workers for her created task $i$ to enhance the total QoS and the work ability as well as arriving time of future workers are unknown to the task requester in advance, we next consider a sequential worker recruitment problem to maximize the expected sum of service quality. Let us first consider the procedure of the sequential worker recruitment as illustrated in Fig. 2. When a task requester opportunistically encounters a “valid” worker at a time point (e.g., $t_1$), she will conduct the worker probe stage as stated in the above section. If she satisfies the service quality of that worker, she will recruit him, conduct the task execution stage and decrease the required worker amount by 1 (e.g., $q_1$). Otherwise, she will reject the worker and wait for a better worker in the later recruitment (e.g., $q_2$). From the above statements we can see

$^5$Since the recruited worker will return the result within the AllottedTime and the precise return time is difficult to predict across different workers, we leverage the time when a requester successfully recruits a worker to reflect the task timeliness.
that a task requester faces a clear tradeoff between exploitation and exploration. If we regard a successful worker recruitment as a stopping point and the achieved service quality as the point utility, the sequential worker recruitment problem can be interpreted as selecting \( n_t \) stopping points to maximize the expected sum of point utility in an online manner. This motivates us to formulate the sequential worker recruitment problem as an online multiple stopping problem, which boils down to deriving the optimal multiple stopping policy.

Definition 3: A multiple stopping policy for a task \( i \) is a set of stopping points \( S_i = \{n_1; s_1, n_2; s_2, \ldots, n_m; s_m\} \), where a stopping point \( n_1; s_1 \) indicates that the \( n_1 \)-th arrived worker is recruited at the time \( s_1 \). The objective is to find the optimal multiple stopping policy \( S^*_i \) to maximize the expected sum of service quality, i.e.,

\[
S^*_i \triangleq \arg \max_{S_i} \mathbb{E} \left[ \sum_{t=1}^{m_i} R_t(s_t) \right]. \tag{3}
\]

We should emphasize that, our framework operates in a “memoryless” manner. That is, if a worker is rejected at a time, he will also be considered in the future. However, if a worker is recruited at a time, he will not be considered again since nobody wishes duplicated results. In practice, the Result Collector with the worker “completion” records can facilitate this operation.

III. ONLINE WORKER RECRUITMENT POLICY

Essentially, the online multiple stopping problem in the equation (3) is a dynamic programming problem, and hence we can leverage the principle of backward induction to solve it. Since we consider the problem for an arbitrary task and requester, we will omit the requester’s index \( r \) and the task’s index \( i \) in all variables for simplicity.

A. Optimal Policy

According to the principle of backward induction, an online multiple stopping problem can be characterized by the Bellman equation. To proceed, we denote \( X^k_t \) as the maximum expected sum of service quality achieved by recruiting \( k \) workers beginning from a time point \( t \). Note that \( X^0_t \) is always 0. Suppose that a requester is probing an arrived worker at a time point \( t \). If the requester decides to reject the worker, the maximum expected sum of service quality is \( X^k_{t+\Delta t} \). (Here, “\( t+\Delta t \)” denotes the next time point of worker recruitment). If the requester decides to recruit the worker, the maximum expected sum of service quality is \( X^k_{t+\Delta t} + R(t) \). Accordingly, we can build up the Bellman equation as follows.

\[
X^k_t = \max \left\{ R(t) + X^k_{t+\Delta t}, X^k_{t+\Delta t} \right\}.
\]

According to the principle of the Bellman equation, if \( R(t) + X^k_{t+\Delta t} \geq X^k_{t+\Delta t} \), that is to say, \( R(t) \geq X^k_{t+\Delta t} - X^k_{t+\Delta t} \), it is optimal for the task requester to recruit the current arrived worker. Therefore, we can see that \( Q(k,t) = X^k_{t+\Delta t} - X^k_{t+\Delta t} \) actually serves as the optimal threshold of service quality. Moreover,

\[
X^k_t = X^k_{t+\Delta t} = Q(k,t) + X^k_{t+\Delta t} \tag{4}
\]

We can hence obtain \( X^k_t = \sum_{i=1}^{k} Q(i,t) \) by iterating the equation (4) (Note that, \( X^0_t \equiv 0 \)), which is used for the next optimal threshold calculation. Based on the above discussion, we can get the optimal multiple stopping policy as follows.

**Theorem 1:** For the sequential worker recruitment problem, it is optimal for a requester to recruit \( m \) workers at the stopping points \( S^*_r = \{n_1^r; s_1^r, n_2^r; s_2^r, \ldots, n_m^r; s_m^r\} \) which satisfy that \( R(s_l^r) \geq Q(m+1-l, s_l^r), l \in [1, m] \).

To better comprehend this theorem, let us consider a general case in which a task requester requires to recruit \( m \) workers for her tasks and she has already recruited \( h \) workers at time points \( s_1, s_2, \ldots, s_h \), before a time point \( t \). In other words, she still requires \( m-h \) workers from the beginning of the time point \( t \). To ease of presentation, we assume that the task requester encounters a worker at the time point \( t \). Conditioned on them, the maximum expected sum of service quality the task requester can obtain in the sequential worker recruitment problem (i.e., recruiting \( m \) workers) denoted by \( Y(t|s_1, s_2, \ldots, s_h) \) is represented as

\[
Y(t|s_1, s_2, \ldots, s_h) = \max \left\{ \sum_{j=1}^{h} R(s_j) + R(t) + X^{m-h-1}_{t+\Delta t}, \right.
\]

\[
\left. \sum_{j=1}^{h} R(s_j) + X^{m-h}_{t+\Delta t} \right\}.
\]

It is not hard to see that, when \( R(t) \) is not less than \( X^{m-h}_{t+\Delta t} = X^{m-h-1}_{t+\Delta t} = Q(m-h, t) \), the arrived worker should be recruited rather than rejected, in order to increase the expected sum of service quality during the sequential worker recruitment. Without loss of generality, if let \( t = s_l^r \) and hence \( l = h+1 \), then the above criterion is in consistence with Theorem 1.

B. Threshold Calculation

In order to calculate the optimal threshold \( Q(k,t) \), we require to study the expression of \( X^k_t \). In terms of the worker arrival model mentioned in Section II-A, the inter-arrival time between two consecutive workers follows the time-continuous Exponential distribution, and hence there is at most one arrived worker at a time point \( t \). This has also been validated by realistic user mobility trace analyse in [19], [20], [22]. In order to obtain the expression of \( X^k_t \), we first consider \( P_q(k,t) \), the probability that a requester successfully recruits a “qualified” worker at a time point \( t \) when she still requires to recruit \( k \) workers. Here, we say a “qualified” worker means that his service quality is greater than the threshold \( Q(k,t) \). Since the probability that a worker arrives at a time point \( t \) is \( 0 ) \), we can therefore obtain \( P_q(k,t) = \beta \Delta t P_q(k,t) \). Further, let \( P_q(k,t) \) be the probability that a requester fails to recruit a
qualified workers (i.e., $1 - P_s(k,t)$). Since $P_s(k,t)$ is small enough at a time point (i.e., $\Delta t \to 0$), according to the Taylor series for exponential function, we can approximately obtain

$$P_f(t,k) = 1 - P_s(k,t) \approx e^{-\beta \Delta t P_s(k,t)}.$$  \hspace{1cm} (6)

In addition, let $E_s(k,t)$ be the conditional expectation of service quality if a requester successfully recruits a qualified workers at a time point $t$, which is

$$E_s(k,t) = E[R(t)]R(t) \geq Q(k,t) = \frac{E[R(t), R(t) \geq Q(k,t)]}{P[R(t) \geq Q(k,t)]} = \frac{\int_{Q(k,t)}^{\infty} \phi(t) x - b \int_{Q(k,t)}^{\infty} f(x) dx}{\int_{Q(k,t)}^{\infty} f(x) dx} \triangleq \frac{E_p(k,t)}{P_f(k,t)}.$$  \hspace{1cm} (7)

According to the definitions of the conditional expectation and the law of total probability, the expression of $X_t^k$ can be viewed as follows.

$$X_t^k = \sum_{v=1}^{\infty} \prod_{t=1}^{k} P_f(k,u) \cdot P_s(k,v) \cdot [E_s(k,v) + X_v^{k-1}],$$  \hspace{1cm} (8)

where $\prod_{t=1}^{k} P_f(k,u)$ represents the probability that a requester recruits no workers during time point $t$ to $v$, $P_s(k,v)$ represents the probability that the first recruited worker beginning from a time point $t$ occurs at a time point $v$, and $E_s(k,v) + X_v^{k-1}$ represents the maximum expected sum of service quality when a qualified worker is recruited at a time point $v$. If we put the equations (5)–(7) into (8), we have

$$X_t^k = \int_{0}^{\infty} e^{-\beta P_s(k,u)du} \cdot \beta P_s(k,u) \cdot [E_s(k,v) + X_v^{k-1}] du = \beta \int_{0}^{\infty} e^{-\beta P_s(k,u)du} \cdot [E_s(k,v) + P_s(k,v) X_v^{k-1}] du.$$  \hspace{1cm} (9)

As $X_t^k$ equals to $\sum_{i=1}^{k} Q(i,t)$, we can get the following theorem.

**Theorem 2**: (Existence) For each $k$ and $t$, the optimal threshold $Q(k,t)$ exists and is the unique solution of the following differential equations with the initial conditions $Q(k,t) = 0$:

$$Q(1,t)' = -\beta E_Q(1,t) + \beta P_Q(1,t)Q(1,t),$$

$$Q(k,t)' = -\beta E_Q(k,t) - E_Q(k-1,t) + \beta P_Q(k,t)Q(k,t) - P_Q(k-1,t)Q(k-1,t),$$

where $Q(k,t)'$ is the derivative of $Q(k,t)$ with respect to time $t$. (Monotonicity) The optimal threshold $Q(k,t)$ is non-increasing in the time domain (i.e., $Q(k,v) \geq Q(k,u)$ if $u \geq v$) and in the required worker amount domain (i.e., $Q(y,t) \geq Q(z,t)$ if $z \geq y$), respectively.

Proof: To begin with, we prove the “Existence” property. As we mention in Section II that the task utility function $\phi(t)$ and the cumulative distribution function of the work ability of workers $F(x)$ are differentiable, $E_0(k,t)$ and $P_0(k,t)$ are also differentiable in terms of the chain rule in calculus. On this basis, the differential equations can be obtained via differentiating both sides of the equation (9) by $t$. For a more solvable mathematical form, we put the expression $E_0(k,t)$ and $P_0(k,t)$ in the equation (7) into the differential equations, and can obtain the following forms:

$$Q(1,t) + b = H(Q(1,t) + b,t),$$

$$Q(k,t) + b = H(Q(k,t) + b,t) - H(Q(k-1,t) + b,t),$$

where $H(x,t) = -\beta \phi(t)g(x/\phi(t))$ and $g(x) = \int_{x}^{\infty} u f(u) du - x \int_{x}^{\infty} f(u) du$. As far as the existence and uniqueness of $Q(k,t)$ in concerned, we can leverage the classic Picard–Lindelöf theorem [25] to prove them. First, since $g(x) \geq 0, g'(x) = -\int_{x}^{\infty} f(u) du \leq 0$ and $g''(x) = f(x) \geq 0$, we can say that $g(x)$ is non-increasing and convex, which indicates that the function satisfies the Lipschitz condition [25]. That is, for any $x_1$ and $x_2$ we have $g(x_1) - g(x_2) \leq C|x_1 - x_2|$, where $C$ is a constant value. In addition, we can have $H(x_1,t) - H(x_2,t) \leq \beta \phi(t) \cdot \frac{\phi(t)}{\phi^2(t)} |x_1 - x_2| = \beta C|x_1 - x_2|$, which indicates that $H(x,t)$ satisfies the Lipschitz condition. Therefore, the equation $Q(1,t) + b = H(Q(1,t) + b,t)$ with initial condition $Q(1,t) + b = 0$ single out exactly one solution according to the Picard–Lindelöf theorem. Further, we can determine $Q(k,t)$ for $k \geq 2$ by an easy induction.

Second, we prove the former part of the “Monotonicity” property (i.e., $Q(k,v) \geq Q(k,u)$ if $u \geq v$). The proof is by mathematical induction. Initially, for $k = 1$, we have $Q(1,t)' = Q(1,t) + \beta = H(Q(1,t) + b,t) = -\beta \phi(t)g(Q(1,t) + b)$. Since $\beta \phi(t)$ and $g(x)$ are all non-negative, $Q(1,t)' \leq 0$, and thus $Q(1,t)$ is non-increasing. Next, we assume that the property holds for $k = n$, and check whether $Q(n+1,t)$ also holds. According to the equation (8), we have

$$X_t^{n+1} = \sum_{v=1}^{\infty} \prod_{t=1}^{n+1} P_f(n+1,u) \cdot P_s(n+1,v) \cdot [E_s(n+1,v) + X_v^{n-1}],$$

Then, we bring the equation (4) in it, and further obtain

$$X_t^{n+1} = \sum_{v=1}^{\infty} \prod_{t=1}^{n+1} P_f(n+1,u) \cdot P_s(n+1,v) \cdot [Q(n,v) + X_v^{n-1}].$$

Since the property is assumed to hold for $k = n$, we have $Q(n,v) \leq Q(n,t)$. Therefore, the first integral on the right-hand-side is no more than $Q(n,t) \sum_{v=1}^{\infty} \prod_{t=1}^{n+1} P_f(n+1,u) \cdot P_s(n+1,v)$, where the expression $\sum_{v=1}^{\infty} \prod_{t=1}^{n+1} P_f(n+1,u) \cdot P_s(n+1,v)$ can be interpreted as the total probability that the first stopping time at a time point $v$ when the task requester requires to recruit $n+1$ workers beginning from a time point $t$. Therefore, we can say that the first integral on the right-hand-side is no more than $Q(n,t)$. In addition, the second integral on the right-hand-side can be interpreted as the expected sum of service quality achieved by selecting $n$ workers\footnote{Specifically, it includes recruiting a worker at a time point $v$ when the task requester requires to recruit $n+1$ workers beginning from a time point $t$ plus recruiting $n-1$ workers beginning from a time point $v + \Delta t$.} beginning from a time point $t$, which is obviously no more than $X_t^n$, the maximum expected values. As such, we can obtain:

$$X_t^{n+1} \leq Q(n,t) + X_t^n \Rightarrow Q(n+1,t) \leq Q(n,t).$$  \hspace{1cm} (12)
As \( Q(n+1,t) = [Q(n+1,t) + b] H(Q(n+1,t) + b) - H(Q(n,t) + b,t) \) and \( H(x,t) \) is non-decreasing based on the non-increasing feature of \( g(x) \), we have that \( Q(n+1,t) \leq 0 \) due to the equation (12). Therefore, \( Q(n+1,t) \) is non-increasing, which completes the proof. At last, since the former part of the “Monotonicity” property holds, the proof of the latter part (i.e., \( Q(y,t) \geq Q(z,t) \) if \( z \geq y \)) is intuitively achieved by the equation (12).

**Further discussions:** First, as many real trace analyses show that opportunistic user encounters are prevalent and sufficient in daily life [7], [15], to enable tractable analysis as stated above, in this paper we consider a mobile crowdsourcing scenario in which the encountering worker set of each task requester is large enough such that the valid worker arrival rate (i.e., \( \beta \)) is relatively constant. Note that, this can also be a good approximation for the case that the number of the required worker amount of a mobile crowdsourcing task is small. In addition, we should emphasize that when the worker set is small, the valid worker arrival rate in the later recruitment actually decreases once a worker is recruited. In other words, we should substitute \( \beta \) with \( \beta(t) \) in the derived differential equations as stated in Theorem 2. Since the value of \( \beta(t) \) is correlated with the recruited worker amount before the time point \( t \), which is impossible to know and predict in advance, ideally we should update the worker arrival rate (i.e., excluding the recruited workers) and recalculate the optimal thresholds after every recruiting one worker, which however leads to much computation overhead. Nevertheless, as the optimal thresholds in our policy are used to filter out the qualified workers (i.e., Theorem 1), and a slight increase of them normally has a limited effect on the worker recruitment, in practice we can update the worker arrival rate and recalculate the thresholds after every recruiting a moderate number of workers in terms of task required worker amount and user encounter frequency. We believe this moderate number can be estimated in offline simulations and analyses by the service providers using our framework.

Second, since workers’ stochastic arrivals occur over a continue-time horizon in reality, in this paper we consider a continue-time dynamic programming solution (i.e., solving the differential equations as stated in Theorem 2). Alternatively, we can design a discrete-time solution as an approximation, which can derive the close-form thresholds in a lightweight way. Specifically, we can discretize the time horizon such that the state space (i.e., \( t \) and \( k \)) is finite, and reformulate the worker recruitment problem to be a finite-horizon dynamic programming problem with finite states which can be solved by standard backward induction. However, in practice it is not easy to determine a proper slot length which is related with task required worker amount and user encounter frequency for partitioning the horizon into the discrete time setting. If the slot length is too small, then the state space for dynamic programming would grow greatly and lead to high complexity. If the slot length is large, the approximation error can be large. Thus, we consider using the continuous-time policy for worker recruitment. In other words, as per Theorem 2, once the task utility function \( \phi(t) \) is specified, we can compute the optimal thresholds by solving the differential equations therein. In practice, this can be done by leveraging the ODE solver in the Apache math library, which is computed efficiently as discussed in the prototype evaluation in Section V-B.

**C. Special Cases: Two Typical Task Utility Functions**

We next illustrate how to calculate the optimal thresholds in terms of two typical task utility functions: 1-0 step function and exponential decaying function.

**1-0 step function:** In this case, we have \( \phi(t) = B \) if \( t < T \), and \( \phi(t) = 0 \) otherwise. This is useful for modeling the hard-deadline driven scenarios. For this case, we can get the following differential equations based on (11) and (12):

\[
y(1,t) = -\beta g(y(1),t),
\]
\[
y(k,t) = -\beta \left[ g(y(k),t) - g(y(k-1),t) \right],
\]

where \( y(k,t) = (Q(k,t) + b)/B \), with the initial condition \( y(k,T) = 0, \forall k \geq 1 \). Generally speaking, it is difficult to obtain the close-form threshold expression from the differential equations. In practice, due to the simple iterated form, we can use the Apache math library to efficiently obtain the numerical results. Since the number of workers that the task requester wants to recruit for the task and the task deadline normally are moderate in mobile crowdsourcing scenarios, the time and device energy consumption for threshold calculation will not be too much. For example, if the required worker amount is 20 and the task deadline is 1 hour, then the time and device energy consumption is only 1.5s and 0.5J as shown in the prototype evaluation in Section V-B.

**Exponential decaying function:** In this case, we have \( \phi(t) = Be^{-\alpha t}, \lim_{t \to \infty} \phi(t) = 0 \). This is useful for modeling the time discounting scenarios. For this case, besides resorting to solving the differential equations (10) and (11) directly, we find a simple way to calculate the thresholds in terms of the connection between \( Q(k,t) \) and \( Q(k,0) \) as follows.

**Lemma 1:** \( Q(k,t) + b = e^{-\alpha t}(Q(k,0) + b) \).

**Proof:** According to Definition 3, the maximum expected sum of service quality achieved by recruiting \( k \) workers beginning from a time point \( t \) can be represented as

\[
X_k^t = \max \mathbb{E} \left[ \sum_{i=1}^{k} q_i R_i(s_i) | s_i \geq t \right]
\]

Taking the equation (2) into above equation and rearranging it, we further have

\[
X_k^t + kb = \max \mathbb{E} \left[ \sum_{i=1}^{k} q_i B e^{-\alpha s_i} | s_i \geq t \right]
\]

\[
e^{-\alpha t} \max \mathbb{E} \left[ \sum_{i=1}^{k} q_i B e^{-\alpha (s_i - t)} | s_i \geq t \right].
\]

If we let \( s_i' = s_i - t \), according to the memoryless of Poisson process (i.e., worker arrival model) and the independence between the work ability and arrival time of workers, we have

\[
X_k^t + kb = e^{-\alpha t} \max \mathbb{E} \left[ \sum_{i=1}^{k} q_i B e^{-\alpha s_i'} | s_i \geq 0 \right]
\]

\[
e^{-\alpha t}(X_0^0 + kb).
\]

Since \( Q(k,t) = X_k^t - X_{k-1}^t \), we have \( Q(k,t) + b = e^{-\alpha t}(Q(k,0) + b) \) based on the above equation, which completes the proof.
According to Lemma 1, we have

\[
\begin{align*}
(Q(1,t) + b') &= \left[e^{-\alpha t}(Q(1,0) + b)\right] = -\alpha e^{-\alpha t}(Q(1,0) + b), \\
H(Q(1,t) + b, t) &= -\beta Be^{-\alpha t}g((Q(1,t) + b)/Be^{-\alpha t}) \\
&= -\beta Be^{-\alpha t}g((Q(1,0) + b)/B).
\end{align*}
\]

Putting the above expressions into (10) and (11), we have

\[
\begin{align*}
\alpha y(1,0) &= \beta g(y(1,0)), \quad (15) \\
\alpha y(k, 0) &= \beta [g(y(k,0)) - g(y(k-1,0))], \quad (16)
\end{align*}
\]

where \(y(k,0) = (Q(k,0) + b)/B\). Therefore, we can obtain the numerical results of \(Q(k,0)\) by solving the numerical equation (15) and (16) with the Apache math library, and then can get any \(Q(k,t)\) based on Lemma 1. Note that, the complexity of threshold calculation is only to solve \(m\) numerical equations, which is lightweight.

IV. DATA-DRIVEN CASE STUDY

In this section, we conduct data-driven case studies to validate our framework model, then illustrate how to apply the optimal worker recruitment policy in practice, and discuss the feasibility of our framework in the end. As user encounter model in opportunistic networks has been used and validated in many existing researches as mentioned in Section II-A, we next do not study it any more but concentrate on the assumption about the work ability. That is, we will consider whether the work ability of workers follows a probability distribution given a specific task. As our general work ability model fits a variety of mobile crowdsourcing services, we mainly discuss two popular ones, content (e.g., image and audio) transcription services and location-based services (e.g., location-based content sensing and transmission).

A. Modeling Work Ability Distribution for Content Transcription Services

As we mention in section II-A, for a given category of this kind of services the keyword space will be a series of representative tags, and the rate vector in the Worker Profiler can be viewed as the interest preference degree or expertise level in terms of those tags.

Since there is no open-access crowdsourcing datasets available, we alternatively analyze the interest preference degree of users by using the data traces from tag-based folksonomy systems. The key rationale is that folksonomy systems generally have diverse categories and users in the systems prefer to mark their interested objects with personal tags or generally have diverse categories in the systems. The key rationale is that folksonomy systems able, we alternatively analyze the interest preference degree of users by using the data traces from tag-based folksonomy systems. The key rationale is that folksonomy systems generally have diverse categories and users in the systems prefer to mark their interested objects with personal tags or public tag cloud, which reflects their interest preference. In practice, multiple users frequently use tags to indicate the same or similar objects, which also helps to reduce the size of the keyword space. Technically, for a folksonomy dataset (i.e., a category) we first construct a weighted tag graph where each node is a tag, the edge between two tags indicates they are used to mark the same object, and the edge weight describes the times they are used together. Then, we apply community detection algorithms\(^8\) to cluster the tags in the graph into communities which indicate the representative keywords in that category.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Dataset} & \text{Modularity} & \text{Clusters} \\
\hline
\text{Delicious} & 0.3024 & 10 \\
\text{CiteULike} & 0.3570 & 7 \\
\text{Vi.sualize.us} & 0.2896 & 9 \\
\text{MovieLen} & 0.2715 & 12 \\
\hline
\end{array}
\]

TABLE II: Community detection results.

The maximum modularity [27] that measures the strength of division of a network into clusters, and the number of representative keywords\(^9\) are listed in Table II. The positive modularity results indicate that the tags in the datasets are of cluster feature, and therefore these clusters can well reflect different keywords for each category.

For the second question, we measure the interest preference degree of a user with respect to a keyword as the ratio of the number of tags he uses belonging to that keyword to the number of tags he uses belonging to any other keyword.

\(^{8}\)For simplicity, we consider a tag community whose size is greater than 1% of the total tags as a representative keyword.

\(^{9}\)In this paper, we adopt the multilevel algorithm [26], and we should emphasize that selecting other kinds of algorithms has little impact on our conclusion about the work ability distribution.
Fig. 3: The distribution of interest preference degree (Top 4 keywords in terms of cluster size in four datasets, red line is Gamma distribution).

B. Modeling Work Ability Distribution for Location-based Services

As we mention in section II-A, for a given category of this kind of services the keyword space will be a series of PoI locations, and the rate vector in the Worker Profiler can be viewed as the visiting frequency of workers in terms of those locations. Taking into account both user amount and time span, we deliberately consider two location-based user mobility datasets:\(^{(1)}\): USCD reflecting a short-term, small-scale user behaviors and Dartmouth reflecting a long-term, large-scale user behaviors. Table III lists the characteristics of the two datasets\(^{(2)}\). In order to explore the work ability distribution we next consider three important issues:

1) Does a mobile user have a regular location-based movement behavior?
2) If yes, how can we measure the visiting frequency of users in terms of a location?
3) Given a location, does the visiting frequency of users follow a probability distribution?

\(^{(1)}\)Dartmouth Community Resource: http://www.crawdad.org/.
\(^{(2)}\)Note that, we take all the APs with the same building name as one location in the Dartmouth dataset. In addition, we only consider the users whose number of daily records are greater than half month.
TABLE III: Characteristics of location-based user mobility datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Locations</th>
<th>Users</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>USCD</td>
<td>140</td>
<td>190</td>
<td>2 months</td>
</tr>
<tr>
<td>Dartmouth</td>
<td>175</td>
<td>2075</td>
<td>3 years</td>
</tr>
</tbody>
</table>

TABLE IV: The average and standard deviation of user movement regularity in location-based user mobility datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>USCD (k=2)</td>
<td>0.814</td>
<td>0.295</td>
</tr>
<tr>
<td>USCD (k=3)</td>
<td>0.870</td>
<td>0.242</td>
</tr>
<tr>
<td>Dartmouth (k=2)</td>
<td>0.749</td>
<td>0.186</td>
</tr>
<tr>
<td>Dartmouth (k=3)</td>
<td>0.770</td>
<td>0.170</td>
</tr>
</tbody>
</table>

For the first question, we consider that mobile users (e.g., students and company staffs) should have regular location-based movement behaviors such as Working Day Movement Model [28]. That is, they will visit their interested locations (or areas) routinely. In order to validate this argument by using data-driven analysis, we first classify the visited locations of each user into $M$, $O$ and $E$ category indicating Mostly, Occasionally, and Exceptionally visited locations, respectively. Specifically, for all the records of a user we count the number of each location he visits, and apply the k-means algorithm (i.e., $k=3$) to get those three categories. Then, we calculate the daily ratio as the number of visited locations belonging to $M$ category\(^{13}\) a day divided by the total location amount in that category for each user, and define the movement regularity of a user as his daily ratio on average. Table IV lists the average and standard deviation of the movement regularity in terms of all the users in each dataset. Note that, we also consider the case that the visited locations of each user are classified into two categories (i.e., Frequently and Infrequently), in order to enhance the reliability of the analysis conclusion. Overall, we can see from Table IV that the average movement regularity of users in each dataset is greater than 70% with small deviation, which indicates that most of mobile users (i.e, students) indeed have regular location-based movement behaviors.

\(^{13}\)We consider that the mostly visited locations of each user can well reveal his movement behavior, and hence we neglect the effect of the other two categories in this paper.

days he visits that location to the total number of days he is active, and we explore the distribution of the visiting frequency of users to answer the third question. Leveraging the same approach in the above section, we can find that the distribution of visiting frequency as partly shown in Fig. 5 can also be well fitted by Gamma distribution since it achieves both the smallest KS value on average and the lowest deviation as shown in Fig. 4(b). Therefore, our assumption that the work ability of workers for location-based services follows a probability distribution (i.e., Gamma distribution) is reasonable. Note that, since the sum of $n$ independent Gamma distributions approximately follows Gamma distribution as mentioned above, our framework can integrate the keyword space of location-based services with that of content transcription services to support mixed services.

C. Applying Optimal Worker Recruitment Policy

We next illustrate how to apply the optimal worker recruitment policy in practice. Suppose a dummy case in which the task utility function is a 1-0 step function with $B=1$ and $T=5$, the work ability of workers follows Gamma(3, 0.02) in terms of task category and keywords, the other task attributes $b=0$ and $m=3$, and the worker arrival rate $\beta$ is predicted to 1. In this case, the requester needs to recruit 3 workers within 5 time points in realtime. After the task is created, our framework will solve the differential equation (12) and (13) to get the thresholds of the optimal worker recruitment policy for the requester as shown in Table V.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$t$</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k=3$</td>
<td>0.056</td>
<td>0.056</td>
<td>0.040</td>
<td>0.023</td>
<td>0.007</td>
<td></td>
</tr>
<tr>
<td>$k=2$</td>
<td>0.102</td>
<td>0.088</td>
<td>0.070</td>
<td>0.049</td>
<td>0.023</td>
<td></td>
</tr>
<tr>
<td>$k=1$</td>
<td>0.163</td>
<td>0.148</td>
<td>0.128</td>
<td>0.103</td>
<td>0.067</td>
<td></td>
</tr>
</tbody>
</table>

TABLE V: The thresholds for the dummy case ($k$ is residual required number of workers and $t$ is time point).

We assume that each worker arrives at a time point and our framework will observe a sequence of workers with work ability $[0.186, 0.057, 0.097, 0.071, 0.085]$. When the first worker arrives, our framework will calculate the corresponding service quality $R(1)$, then $s_1^*=1$ since $R(1) > Q(3, 1)$ (i.e., $0.186>0.068$). Similarly, $s_2^*=3$ and $s_3^*=5$. Therefore, the sum of online service quality gain is $0.186+0.097+0.085=0.368$, which is just the same as the offline optimal one.
D. Applicability Discussion

As we mentioned before, the **Crowd Foraging** framework does not aim to replace online systems but to complement them with local human workers. In this spirit, we consider that the collaboration among service providers of online systems, mobile requesters and mobile workers will make the proposed framework become true in reality. This is because this framework enables service providers of online systems to extend the scope of their services, enables mobile requesters to obtain the task results quickly by proactively recruiting nearby workers, and enables mobile workers to share information in exchange of monetary rewards without scanning online systems periodically.

Specifically, they can create a mobile crowdsourcing application as the framework, in which service providers of online systems can adopt our data analysis approaches above to obtain the distribution of the representative keywords in their supported task categories offline, build the Worker Profiler in terms of the historical tasks (i.e., category and keywords) a user has executed in online systems and this application, mobile users (i.e., requesters and workers) can leverage many approaches in existing researches (e.g., [19], [20], [22]) to obtain their individual movement behaviors (e.g., relationship between users as well as between users and locations), and the QoS of this application can be guaranteed by the proposed optimal worker recruitment policy. In addition, in the presence of service providers the **Crowd Foraging** framework can borrow some reputation management strategies from online systems (e.g., multiplexing online servers for reputation management) to reduce the number of malicious and untruthful workers.

V. PERFORMANCE EVALUATION

In this section, we first conduct extensive trace-driven simulations with the widely adopted Opportunistic Network Environment simulator (ONE) to evaluate the performance of the **Crowd Foraging** framework, and further implement a system prototype in Android platform to evaluate the framework overheads in practice.

A. Trace-driven Evaluation

**Simulation scenario:** Since there is no available realistic datasets involving both user D2D contact records and user interest preference records, we attempt to simulate the scenarios for content transcription services by integrating realistic user contact traces with user interest preference in the folksonomy datasets. Specifically, we consider two typical user contact traces: Infocom06 and MIT Reality, where mobile users with Bluetooth-enabled devices periodically detect their peers nearby, and record contacts over several days. The reasons for selecting them are two-fold. First, the inter-encounter time of majority of users in them follows Exponential distribution, as evidenced by the previous researches [19], [22]. Second, Infocom06 and MIT Reality reflect diverse network scenarios (i.e., a dense conference and a sparse campus, respectively). Then, according to the user amount in user contact traces, we randomly select the corresponding number of user interest preference from the CiteULike dataset for simulation.

In addition, since the existing location-based user mobility datasets such as UCSD and Dartmouth do not involve user D2D contact records, we create a synthetic scenario for location-based services. That is, we consider the virtual city scenario built in ONE simulator where users move on the roads to visit our specified locations in terms of the Working Day Movement Model which captures the exponential property of user inter-encounter time [28]. In detail, we uniformly pick several locations in the city map, and randomly select the corresponding number of locations from Dartmouth dataset to match them. Then, we randomly select the visiting frequencies of a user with respect to those selected locations from the dataset for a mobile user in the scenario. In this context, each mobile user in the simulation will decide the next visiting location according to the (normalized) visiting frequencies, and will get there using the shortest path on the map.

**Simulation Setup:** We compare our worker recruitment policy, namely **Crowd** for short with three alternative policies:

- **Greedy:** a gradient ascend policy widely used in oppor-
tunistic content routing researches [1], [6], [10], [11]. That is, a task requester recruits the worker whose service quality is greater than the average service quality among the workers the requester has recruited.

- mSecretary: a classic multiple secretary recruitment policy proposed in [29]. That is, a task requester skips the first \( BT/e \) workers but records their service qualities as the benchmark, and then she recruits the worker whose service quality is greater than the \( k \) (i.e., the residual required worker amount) biggest one in the benchmark.
- Random: a task requester recruits a worker in a random manner (e.g., with 50% probability).

For the performance metric, we leverage the offline optimal policy (i.e., a task requester knows all the future information including the arrival time and the corresponding work ability of workers beforehand, and recruits the best ones in terms of their service qualities and the required worker amount of the requester) to get the offline optimal performance. Then, we define QoS Ratio as the result of the performance provided by a policy to the offline optimum.

In the simulation for content transcription services, we evaluate the performance of different policies in terms of the required worker amount under two typical task utility functions. For the other parameters, we set the task initial value \( B = 8000 \), task reward \( b = 120 \) (i.e., \( p_{\text{acc}} = 0.69 \) using the model in [24]), and consider all the workers can return the results in time (i.e., \( p_{\text{re}} = 1 \)). We choose two consecutive days with the highest number of user contact records in two traces for simulation, and calculate the \( \lambda_r \) for each user \( r \) by using the method in [19]. In the simulation for location-based services, we evaluate the performance of different policies in terms of the number of users and locations in the scenario under the exponential decaying function. The other settings expect \( p_{\text{re}} \) are similar to those above (e.g., we also choose two days for simulation and user arrival rate calculation). Here, we consider each task requester will recruit 5 workers to sense data at her specified location (i.e., \( m = 5 \)), and a recruited worker can return the results if and only if he visits that location. The simulation results below are the average value of 20 chose requesters in each dataset running 50 simulations with different scenarios.

**Simulation Results**: The simulation results for content transcription services are shown in Fig. 6 and Fig. 7. As for the step function (i.e., sub-figure (a)), the performance of Crowd in both figures slightly increases and is more closer to the offline one (i.e., less than 10% in the end) as the required worker amount increases. This is because our threshold-based policy is adept at screening out qualified workers, and thus the positive stacking effect is much clearer with the required worker amount increasing. When the required worker amount is small, the samples in mSecretary make positive effect to filter out qualified ones, while when the required work amount is large, the samples will deteriorate the performance since the residual number of workers is small before deadline (i.e., the samples include many qualified workers). From the overall perspective, our policy achieves superior performance, which increases 30%~40% QoS compared with the others.

As for the exponential function (i.e., sub-figure (b)), we observe that the overall trend is similar to that of the step function, and ours achieves more than 30% QoS gain. The only difference is that our performance slightly decreases when the required worker amount is large. This is because the threshold \( Q(k, t) \) is decided by \( Q(k, 0) \) (i.e., Lemma 1), and the value of \( Q(k, 0) \) will decrease with \( k \) increasing (i.e., monotonicity property). Therefore, the threshold may be not tight enough to filter out the qualified workers especially when the decay rate is small as shown in Fig. 6(d) and Fig. 7(d). With the
decay rate increasing, our performance is getting better. Since the case of the exponential decaying function with small rate is similar to that of the step function with infinity deadline, as shown in Fig. 6(c) and Fig. 7(c), our performance is getting slightly worse with the task deadline $T$ increasing. Despite of this, we can see that our policy still has the better performance than the others, and that it is more adaptive to the mobile crowdsourcing scenario with realtime requirement.

![Fig. 8: The simulation results of the synthetic trace.](image)

The simulation results for location-based services are shown in Fig. 8. As depicted in Fig. 8(a), the performance of our policy slightly decreases as the total number of users increases, while that of Greedy and Random policy decreases clearly. We consider the reason is that, the increasing number of users not only increases the worker arrival rate but also injects many diverse workers in the network, which may enable Greedy and Random policy to select more non-qualified workers. On the contrary, both Crowd and mSecretary policy are aware of the worker arrival rate, and therefore their performance is better. In general, our policy still achieves 25% QoS increase than mSecretary policy on average. As depicted in Fig. 8(b), the performance of all the policies keep stable as the total number of locations increases. This is because the visiting frequency of users in terms of the locations are derived from the realistic Dartmouth dataset in which both the hot locations and the regular visited locations of each user are limited. That is, each user in the simulation will move around several locations with very high probability, and hence increasing the number of locations has little impact on the performance of the policies.

B. Prototype Implementation

We implement a system prototype in Android platform as shown in Fig. 9 according to the framework model in Section II-A, and evaluate it on several off-the-shelf smartphones including XiaoMi 4, Samsung Note 5, and Meizu Pro 5. Specifically, we consider the system overhead in terms of threshold calculation of optimal policy, automatic peer discovery, and Bluetooth based data transmission (i.e., task contents transmitted from task requesters to workers), and show the average evaluation results in Fig. 10.

For the overhead of threshold calculation, we first test different combination of required worker amount and task deadline (hour) in the case of 1-0 step function, and leverage Monsoon power monitor to obtain the execution time and energy consumption. As shown in Fig. 10, we observe that the required worker amount dominates the calculation overhead. The reason is that according to the iterative form in equation (12) and (13) for threshold calculation, if we want to get the threshold for a large required worker amount $m$, we have to obtain all the thresholds from 1 to $m$, and therefore the calculation overhead is superposed. Since the situation that task requesters expect a large number of workers with a long deadline is not common in mobile crowdsourcing scenario, we can say that the overhead of threshold calculation for the 1-0 step function case is acceptable. As the complexity of threshold calculation is to solve numerical equation in the case of exponential decaying function, the calculation overhead is much lightweight. For example, when $m$ is 40, the energy and time consumption is only 76mJ and 20ms, respectively.

In accordance with the Infocom06 and MIT Reality dataset, we adopt Bluetooth as the technology of D2D communication in the prototype. In this context, for automatic peer discovery in opportunistic networks, mobile devices (i.e., workers) can continuously be discoverable without wasting much energy (i.e., the evaluated discoverable power is 6.8mW on average). Although the process of actively scanning for peers (i.e., requesters) naturally consumes much more energy such as the evaluated scanning power is 187.4mW on average when the scanning period is 1 minute, the hourly energy consumption for device discovery is less than 12J, which is still acceptable in practice.

At last, we conduct a simple test to approximately measure...
the energy consumption of Bluetooth based data transmission. Specifically, two smartphones that have closed all the background services, cellular link and screen are connected with each other through Bluetooth at a moderate distances (roughly 5 meters in an outdoor environment). Then, one smartphone with full battery will repeatedly transfer a 500Mb video file to the other one until its battery is depleted, and we observe that the number of the iterated file transfer is 76, and the average TCP throughput is 1.83Mpbs. In other words, transferring 500Mb data in this setting roughly consumes $\frac{1}{76}=1.3\%$ of total device energy. If we consider that the content size of a task is 20Mb (e.g., a dozen of images), the energy consumption of task content transmitted to a worker accounts for $0.052\%$ (i.e., $\frac{20\times0.052}{7600}$) of device battery, and the delivery time is 10.92s (i.e., $\frac{20}{7600}$). To sum up, the test results of the system prototype demonstrate that the proposed Crowd Foraging framework is cost-efficient and feasible in practice.

VI. RELATED WORK

In order to cope with the first issue in current online mobile crowdsourcing systems that it is difficult for newly created tasks to attract enough workers in a short time, many researches study to proactively recruit available workers for newly created tasks in online systems, where a central service entity collects the information of both task requesters (e.g., budget constraint) and workers (e.g., truthful ability and bidding price), and makes the task-worker assignment to achieve an optimal global utility (e.g., [30], [31], [32] and the references therein). However, due to the self-organized nature, task requesters in the Crowd Foraging framework do not have global worker information in advance, and thus have to probe the work ability and make the sequential recruitment decision, which motivates us to develop an online multiple stopping policy for worker recruitment accordingly. In addition, these online centralized worker recruitment policy has no help to improve system flexibility and scalability when mobile crowdsourcing reaches maturity.

Mobile crowdsourcing over opportunistic networks and its closely-related paradigms such as cooperative opportunistic computing leveraging local worker knowledge and device resources have motivated many researchers in recent years. For example, the researches [6], [10] consider the location visiting frequency as the work ability of workers, and propose gradient ascend policies that forward the task contents to a user with higher visiting frequency in terms of the destination area. The authors in [3] regards user deterministic movement trajectory as work ability, and formulate the recruitment of workers as a minimum cost set cover problem which is solved by a heuristic greedy algorithm. However, the objective of the above researches is all to design optimal multi-hop mechanisms to minimize the number of recruited workers conditioned on the QoS requirements, and meanwhile their worker ability models are different from us, which makes their solutions not to be applied to our framework. In addition, some researches [1], [14] take user encounter frequency and device computation resources (i.e., work ability) into consideration, and explore how to partition the whole computational task into suitable subtasks in terms of the work ability of arrival workers so as to minimize the task makespan, which is still different from the objective in this paper.

VII. CONCLUSION

In this paper, we propose a novel Crowd Foraging framework for QoS-oriented, self-organized mobile crowdsourcing. We present a comprehensive framework model that fully integrates human behavior factors for modeling task profile, worker arrival and work ability. Then, we formulate the sequential worker recruitment problem as an online multiple stopping problem to maximize the expected sum of service quality (i.e., total QoS), derive an optimal online worker recruitment policy, and show that it exhibits a nice threshold based structure. Extensive trace-driven evaluations and Android based system prototype demonstrate that our framework achieves superior performance and is cost-efficient.

VIII. ACKNOWLEDGMENT

We thank the anonymous reviewers and editors for their valuable and insightful comments. This work is supported in part by the National Key Research and Development Program of China (No. 2016YFB0201900), Start-Up Fund from Sun Yat-Sen University, EU FP7 IRSES MobileCloud project (No. 612212), EU-Japan Horizon2020 ICN2020 project (EU No. 723014 and NICT No. 184), Alexander Humboldt Foundation and the Natural Science Foundation of Tianjin, China (No. 16JCQNJC00700).

REFERENCES


