Noname manuscript No.
(will be inserted by the editor)

The Design and Evaluation of Task Assignment Algorithms for GWAP-based Geospatial Tagging Systems

Ling-Jyh Chen · Yu-Song Syu · Hung-Chia Chen · Wang-Chien Lee

Received: 2010/XX/XX / Accepted: date

Abstract Geospatial tagging (geotagging) is an emerging and very promising application that can help users find a wide variety of location-specific information, and thereby facilitate the development of advanced location-based services. Conventional geotagging systems share some limitations, such as the use of a two-phase operating model and the tendency to tag popular objects with simple contexts. To address these problems, a number of geotagging systems based on the concept of 'Games with a Purpose' (GWAP) have been developed recently. In this study, we use analysis to investigate these new systems. Based on our analysis results, we design three metrics to evaluate the system performance, and develop five task assignment algorithms for GWAP-based systems. Using a comprehensive set of simulations under both synthetic and realistic mobility scenarios, we find that the Least-Throughput-First Assignment algorithm (LTFA) is the most effective approach because it can achieve competitive system utility, while its computational complexity remains moderate. We also find that, to improve the system utility, it is better to assign as many tasks as possible in each round. However, because players may feel annoyed if too many tasks are assigned at the same time, it is recommended that multiple tasks be assigned one by one in each round in order to achieve higher system utility.

A preliminary version of this study was published in the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'09), Washington D.C., USA, 2009 [12]. In this extended version paper, we have refined our analysis in modeling GWAP-based geospatial tagging systems, re-evaluated the five task assignment algorithms based on the new analytical model, and included a more comprehensive set of evaluations with different numbers of tasks per assignment and different buffer sizes per LOI. Moreover, we have updated the literature review of this study, and incorporated all the comments/suggestions of the conference attendees. Hence, this manuscript is a much more thorough and authoritative presentation of our study on GWAP-based geospatial tagging systems.

L.-J. Chen, Y.-S. Syu, and H.-C. Chen
Institute of Information Science, Academia Sinica
128, Sec. 2, Academia Road, Taipei 11529, Taiwan
Tel.: +886-2-27883799; Fax: +886-2-26518661
E-mail: {ccljj, yssyu, hcchen}@iis.sinica.edu.tw

W.-C. Lee
Department of Computer Science and Engineering, The Pennsylvania State University
111 Information Sciences and Tech. Building, University Park, PA 16802, USA
E-mail: wlee@cse.psu.edu

Keywords Geospatial Tagging · Games with a Purpose · Human Computation · Tagging

1 Introduction

With the advances in computer and communication technology, mobile and wireless computing technologies have affected every aspect of our working and living environments. Mobile handheld devices, which exploit the capabilities of wireless communications and people's mobility in their daily lives, have shown promise in a variety of advanced pervasive applications. Integrating such devices with other widely used technologies (e.g., environmental sensors and GPS receivers) could facilitate the development of future cyber-physical systems (CPS) [21], such as mobile urban sensing [17, 25], urban monitoring [15, 19], and geospatial tagging (geotagging) systems [3, 4].

Among these applications, geotagging is an emerging and very promising application because it can help users find a wide variety of location-specific information, and thereby facilitate the development of advanced location-based services (LBS). For instance, in a geotagging system, a user might 'tag' an Italian restaurant with its latitude and longitude coordinates, a picture of the restaurant, and its menu; and another user might ask the LBS provider: "*Are there any Italian restaurants nearby?*" The provider will then search the geospatial database provided by the geotagging system.

Conventional geotagging systems operate in two phases. First, users must prepare the GPS coordinates and other descriptive materials (e.g., text, images, or a video clip) of the target object; then they have to upload the information to the system's website. The process is inconvenient and therefore frustrates end users. Moreover, the geotagged items tend to be clustered in hot spots, as they are provided voluntarily by the system users. Since most people are interested in popular items, conventional systems tend to favor general objects (e.g., museums and restaurants), rather than specialized ones (e.g., restaurants that allow pets). As a result, the scalability and extensibility of the systems are limited, so a solution that can properly address the above three issues is highly desirable.

Inspired by the concept of '*Games with a Purpose*' (GWAP) [32, 34], several researchers have proposed GWAP-based geotagging systems in recent years [8, 11, 14, 18, 26, 27, 29, 30]. Unlike conventional approaches, GWAP-based geotagging systems are asynchronously interactive. Moreover, they transform the geotagging process into a game that is entertaining and appealing to players. By taking advantage of people's desire to socialize and be entertained, the systems can successfully outsource the *geo-parsing* and *geo-coding* processes, which are highly sophisticated and labor-intensive operations [20, 31]. Since a mission can be initiated by any person, at any location, and in any context, GWAP-based geotagging systems are better able to provide information about less popular spots and specialized topics. In addition, because each mission can be initiated and processed on a handheld device, the systems are not affected by the legacy problem of the two-phase operating model.

Existing GWAP-based geotagging approaches focus on the design, implementation, and measurement of real-world applications/systems; however, some recent studies have shown that the performance of GWAP systems can be improved significantly if they are played with strategies [13]. Specifically, there are two goals in GWAP-based geotagging systems. First, it must solve as many tasks as possible, and this tends to favor popular LOIs at the expense of unpopular ones. Second, the system needs to strike a balance between the outcomes of LOIs in proportion to LOIs' popularity, and this may result in a low acceptance rate of some task assignments. Thus, a clever task assignment strategy that can accommodate the two goals is highly desirable. To this end, we formulate the problem as a variant of the classic

scheduling problem [9, 10, 24]. Moreover, we analyze the intrinsic properties of GWAP-based geotagging systems, and evaluate the system performance in terms of throughput and fairness under synthetic and realistic mobility scenarios. The contribution of this work is three-fold.

1. We present a detailed analysis that can be applied easily to existing systems to model a generic GWAP-based geotagging system.
2. We develop three metrics to evaluate the performance of GWAP-based geotagging systems.
3. We design five task assignment algorithms, namely, the *Random Assignment* algorithm (RA), the *Simple Assignment* algorithm (SA), the *Acceptance-Rate-First Assignment* algorithm (ARFA), the *Least-Throughput-First Assignment* algorithm (LTFA), and the *Hybrid Assignment* algorithm (HA).

Using a comprehensive set of simulations, we evaluate the algorithms' performances in various scenarios. Based on the results, we draw the following conclusions.

1. Although the HA scheme achieves the best performance in terms of system utility, it is not appropriate for real-world deployment because it is computation-hungry. In contrast, the LTFA is suitable, since its performance is comparable to that of the HA scheme, while its computational complexity is moderate and acceptable.
2. It is better to assign as many multiple tasks as possible in each round, and thereby maximize the system utility. However, players may feel annoyed if too many tasks are assigned at the same time in a round.
3. Therefore, to maximize the system utility, it is recommended that multiple tasks be assigned one by one in each round.

The remainder of this paper is organized as follows. Section 2 contains a review of related works on GWAP-based geospatial tagging systems. In Section 3, we present our analysis of such systems, and discuss three evaluation metrics. In Section 4, we compare the five proposed task assignment algorithms, namely the RA, SA, ARFA, LTFA, and HA schemes. In Section 5, we provide a comprehensive set of simulation results, which we analyze and explain in detail. We then summarize our conclusions in Section 6.

2 Background

The *Games With A Purpose* (GWAP) genre [32, 34] is a type of *Human Computation* that outsources certain steps of the computational process to humans [20, 31, 36]. By taking advantage of people's desire to be entertained, GWAP attract people to play voluntarily, and also produce useful metadata as a by-product. The games have shown promise in solving a variety of problems, such as image annotation [33] and commonsense reasoning [23, 35], which computer computation has been unable to resolve completely thus far.

Several GWAP-based geospatial tagging (geotagging) applications have been proposed in recent years [8, 11, 14, 18, 26, 27, 29, 30]. Generally, they can be grouped into two categories, *non mission-based* and *mission-based*, depending on the level of interaction among the players in a game. Non mission-based GWAP geotagging systems perform human computation by providing players with incentives to contribute geospatial tags voluntarily, without implementing the concept of missions. They are generally implemented in the form of location-based social networks (LBSN) (such as *Foursquare* [2], *Gowalla* [5], and *GyPSii* [6]) or by following the rules of conventional games [26, 27, 29, 30].

For instance, the *GeoTicTacToe* [30] game, which is based on the traditional *TicTacToe* game, is used to study the synchronization problem in geospatial gaming systems. In the *CityPoker* [29] game, which is an adaptation of the popular card game, each player must try to improve his/her starting hand by changing cards at 15 locations in the real world. By analyzing the players' tracks, *CityPoker* provides researchers with a way to model the players' behavior and interpret the trajectory of a person moving in a spatial environment. The results can help *spatially grounded intentional systems* provide adequate services automatically according to the users' intentions. Moreover, *CityExplorer*, proposed by Matyas et al [26, 27], is a location-based variant of the popular board game *Carcassonne* [1], which only allows players to place tokens (followers) in pre-defined types of real-world locations. As a result, players are "forced" to explore the unstructured game area, which allows the system to collect specific geospatial data, such as the geographic coordinates and the classifications of real-world objects, as a by-product.

As their name suggests, mission-based GWAP geotagging systems generate 'missions' that players must solve. They provide incentives via a reward scheme that gives points to players when they contribute to the system (e.g., when they initiate or solve a task, or review a solution) [8, 11, 14, 18]. Since *non mission-based* games follow certain rules, in this study, we regard them as special instances of *mission-based* systems with multiple tasks that are assigned all at once in each round. In this context, 'multiple tasks' means all the actions that players are allowed to take.

For instance, *Hitcher* [14] is a location-based mobile game that exploits the cellular infrastructure. Specifically, in a *Hitcher* game, players create digital hitch hikers (with names, destinations and questions to ask other players) and drop them into their current cellular phone cell; or they pick up available hitchers from their current cell, answer their questions, carry them to new locations, and drop them again. As the hitchers pass from player to player, phone to phone and cell to cell, the system collects the meaningful space names of each cell as a by-product. *MobiMission* [18] is a location-based pervasive social game in which missions are created, solved, and reviewed by players. The system does its best to assign "nearby" missions to players (e.g., *Find a good cafe near the Empire State Building*). If there are no nearby missions, it assigns "location-independent" missions instead (e.g., *Take a picture of a tall building*). Similarly, in the *Gopher* game [11], missions are created, solved, and reviewed by players, but the system only assigns missions to "nearby" players. Finally, the *Eyespy* game [8] allows players to tag geographic locations with photos or text; or 'confirm' the locations of places that other players have tagged. The *MarkIt* game [28] encourages players to take pictures of their physical surroundings and upload to the game server. Then, players view all pictures and have one chance with each image to label it or map it to where they feel the picture was taken. As a result, *Eyespy* and *MarkIt* produce a collection of recognizable and locatable geographic details, which could be used for navigation and location-based applications.

Although most GWAP-based geotagging works focus on the design, implementation, and measurement of real-world applications/systems, recent studies have shown that the performance of GWAP systems can be improved significantly if they are designed and played with strategies [13]. This finding motivates us to investigate the design strategies of generic GWAP-based geotagging systems. We present our analysis in the following sections.

3 GWAP-based Geotagging Systems

In this section, we describe GWAP-based geotagging systems and formulate the problem in sub-section3.1. Then, we present our analysis in sub-section 3.2, and propose the three evaluation metrics, namely the *throughput utility*, the *fairness utility*, and the *system utility*, in sub-section 3.3.

3.1 Game Description

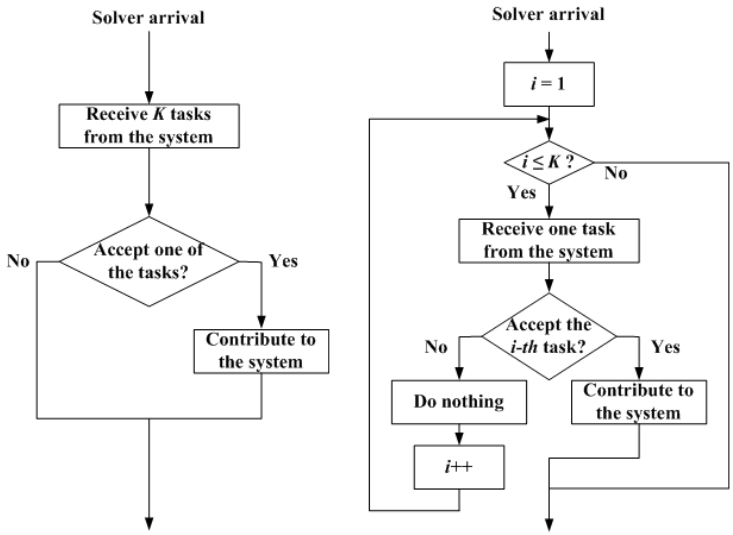
We describe a generic GWAP-based geotagging system that is both *centralized* and *mission-based*. The server is responsible for data storage and task assignment, while the clients are mobile players equipped with GPS-enabled handhelds and wireless connectivity (e.g., 3G/WiFi) to the server via the Internet. There are two game-playing roles: *Requester* and *Solver*; and players are allowed to switch between the roles freely during run time.

We assume that there are N *locations of interest* (LOI) in a game, and that the server maintains N FIFO-based task queues for each LOI in the system. Let Q_i denote the task queue of the i-th LOI, and let players arrive at the i-th LOI at a Poisson rate of λ_i per time unit. We denote the ratio of the number of *Requesters* over the number of *Solvers* as r (i.e., the *Requester* arrival rate is $\frac{r}{r+1}\lambda_i$ and the *Solver* arrival rate is $\frac{1}{r+1}\lambda_i$). The *Requester* always initiates a new task at his current LOI (e.g., *What is the best Chinese restaurant in this area?*). Thus, the task, along with the requester's ID and the timestamp, is stored in his current LOI's task queue on the server. Meanwhile, the *Solver* is assigned the headmost tasks of K distinct task queues at most ($K < N$) and has to decide whether to accept any one of them. The K tasks may be assigned all at once, or in a one-by-one manner, as shown in Figure 1. The task acceptance decision may be based on a combination of factors, such as the distance and the popularity of the task's LOI. Once the solver accepts a task, he must solve it and thus contribute to the system.

To be effective, the system must achieve two task assignment goals. First, it must solve as many tasks as possible (i.e., improve the assignment acceptance rate). This tends to favor popular LOIs at the expense of unpopular ones (i.e., the '*starvation*' problem). Second, the system needs to strike a balance between the outcomes of LOIs (i.e., the number of each LOI's outcomes should be in proportion to the LOI's popularity). This may result in a low acceptance rate of some task assignments (i.e., the '*Equality of Outcome*' problem). Thus, a clever task assignment strategy that can accommodate the two goals is highly desirable. To this end, we formulate the problem as a variant of the classic scheduling problem [9, 10, 24]. Moreover, we analyze the intrinsic properties of GWAP-based geotagging systems, and evaluate several system implementation strategies.

3.2 Game Analysis

First, we model the *willingness* of a solver u to accept the assignment of a task v. In this study, a player's willingness is determined intuitively based on a combination of factors, such as the distance and popularity of the assigned task. Specifically, the shorter distance between u and v, the more willing u will be to accept v; and the more popular the associated LOI of v, the greater the probability that u will accept v. We define $W(u, v)$ as the *willingness* (i.e., the likelihood) that u will accept v as follows:



(a) All-at-once assignment (b) One-by-one assignment

Fig. 1 The flowcharts of task assignment approaches when a solver enters the system.

$$W(u,v) = \min(f_d(u,v) \times f_p(v), 0.95), \tag{1}$$

where $f_d(u,v)$ represents the *distance* factor and $f_p(v)$ represents the *popularity* factor of the task v for the solver u. The upper bound of $W(u,v)$ is set at 0.95 because a player is only allowed to tackle one task in each round, even if there are multiple tasks for which the values of $f_d(u,v) \times f_p(v)$ are equal to or greater than 1. Thus, the *unwillingness* of u to accept v can be defined as $1 - W(u,v)$; that is, a single task is a *Bernoulli* event with a success probability $W(u,v)$.

Note that the definitions of $f_d(u,v)$ and $f_p(v)$ are flexible and depend on the implementation of the system. However, they should satisfy the following criteria: 1) the value of $f_d(u,v)$ decreases monotonically as the distance between u's LOI and v's LOI increases; and 2) $0 \leq f_p(v) \leq 1$, and the value of $f_p(v)$ increases monotonically with the popularity of v's LOI. We define $f_d(u,v)$ and $f_p(v)$ as follows:

– *The distance factor* $f_d(u,v)$*:* We adopt the *Sigmoid Function*[1] to model the distance factor $f_d(u,v)$ as follows:

$$f_d(u,v) = 1.5 - \frac{1}{1 + e^{\tau - d(u,v)}}, \tag{2}$$

where $d(u,v)$ is the Euclidian distance between u and v; and τ is the threshold that determines whether the distance affects the solver's decision in a positive or negative way (i.e., $f_d(u,v) > 1$ when the distance between u and v is less than τ; otherwise, $f_d(u,v) \leq 1$, as shown in Figure 2).

[1] The *Sigmoid Function*, which starts with a small value and accelerates over time to approach a maximum, is widely used for modeling natural processes and complex system learning curves [16].

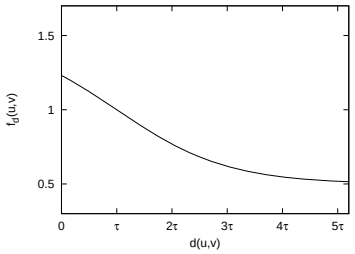


Fig. 2 Illustration of the distance factor $f_d(u, v)$ with different $d(u, v)$ values.

– *The popularity factor $f_p(v)$:* Since the ratio of the number of requesters over the number of solvers is r, the solver arrival rate at the i-th LOI is $\frac{\lambda_i}{r+1}$. Therefore, the probability that there will be n solvers at the i-th LOI can be derived by Equation 3. The popularity factor of the task v (which is located at the i-th LOI) is defined by Equation 4, i.e., the probability that there will be at least three solvers at the i-th LOI.

$$p_i(n) = \frac{(\frac{\lambda_i}{r+1})^n e^{-\frac{\lambda_i}{r+1}}}{n!}. \tag{3}$$

$$f_p(v) = 1 - p_i(0) - p_i(1) - p_i(2). \tag{4}$$

However, since λ_i is unknown in real systems, we use Equation 5 to approximate λ_i based on the exponential moving average (EMA), where α is a constant smoothing factor ($0 < \alpha < 1$), and $N_i(t)$ is the number of player arrivals at the i-th LOI at time t.

$$\lambda_i'(t+1) = \alpha\lambda_i'(t) + (1-\alpha)N_i(t). \tag{5}$$

Let $N_i^{Solver}(t)$ be the number of solver arrivals at the i-th LOI at time t; $u_i(t, j)$ be the j-th solver arriving the i-th LOI at time t; and $v_i^k(t, j)$ be the k-th task assigned to him/her ($1 \le k \le K$). For ease of presentation, we use $\mathcal{W}_k$ to denote $W(u_i(t, j), v_i^k(t, j))$. Thus, the number of tasks completed in each round, denoted by $I_{i,j}(t)$, forms a *Bernoulli* distribution with a success probability $1 - p_0$, i.e.,

$$I_{i,j}(t) = \begin{cases} 0 \text{ , with a probability } p_0; \\ 1 \text{ , with a probability } 1 - p_0; \end{cases} \tag{6}$$

where p_0 represents the probability that no tasks will be completed in a round. Let p_k be the probability that the j-th solver chooses to complete the k-th task. Then, we can obtain the values of p_0 and p_k in the following two cases:

– *Case 1: The headmost K tasks are assigned all at once.* Since the solver can choose to complete one task (i.e., with a willingness $\mathcal{W}_k$ for the k-th task) or reject all the K tasks, we let $\mathcal{W}_0$ denote the solver's unwillingness to accept any k tasks and define the value of $\mathcal{W}_0$ by Equation 7. We can obtain the value of p_k, for $0 \le k \le K$, by Equation 8.

$$\mathcal{W}_0 = 1 - \frac{\sum_{k=1}^{K} \mathcal{W}_k}{K}. \tag{7}$$

$$p_k = \begin{cases} \frac{\mathcal{W}_k}{\sum_{l=0}^{K} \mathcal{W}_l} & \text{, if } 1 \le k \le K; \\ 1 - \sum_{l=1}^{K} p_l & \text{, if } k = 0. \end{cases} \tag{8}$$

– *Case 2: The headmost K tasks are assigned one-by-one.* In this case, the K tasks are consecutive and independent *Bernoulli* events with a success probability $\mathcal{W}_k$ for each $k \in [1 \cdots K]$, and the game round stops when one of the tasks is completed. Thus, p_0 represents consecutive failures and the value of p_k, for $0 \leq k \leq K$, can be obtained by Equation 9.

$$p_k = \begin{cases} \prod_{l=1}^{k-1} (1 - \mathcal{W}_l)\mathcal{W}_k & \text{, if } 1 \leq k \leq K; \\ \prod_{l=1}^{K} (1 - \mathcal{W}_l) & \text{, if } k = 0. \end{cases} \tag{9}$$

Let $T(t, i)$ be the number of tasks completed by the solver arrivals of the i-th LOI at time t; and let $A(t)$ be the overall number of tasks solved in the system up to time t. We derive $T(t, i)$ and $A(t)$ by Equations 10 and 11 respectively.

$$T(t, i) = \sum_{j=1}^{N_i^{Solver}(t)} I_{ij}(t). \tag{10}$$

$$A(t) = \sum_{i=1}^{N} \sum_{j=1}^{t} T(j, i). \tag{11}$$

Let p_i denote the popularity of the i-th LOI. Then, we can obtain $\widetilde{T}(t, i)$, the normalized system throughput at the i-th LOI at time t (w.r.t. p_i) by Equation 12; and derive $\widetilde{\mu}(t)$, the average normalized system throughput at time t by Equation 13.

$$\widetilde{T}(t, i) = \frac{\sum_{j=1}^{t} T(j, i)}{p_i}, \tag{12}$$

$$\widetilde{\mu}(t) = \frac{1}{N} \times \sum_{i=1}^{N} \widetilde{T}(t, i). \tag{13}$$

In addition, the coefficient of variation of the normalized system throughput at time t, $c.v.(t)$, can be derived by:

$$c.v.(t) = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (\widetilde{T}(t, i) - \widetilde{\mu}(t))^2}}{\widetilde{\mu}(t)}. \tag{14}$$

3.3 Evaluation Metrics

As mentioned earlier, GWAP-based geotagging systems have two goals. To evaluate the system performance, we propose the following three evaluation metrics. First, the system prefers players' *best-efforts*, i.e., they should solve as many tasks as possible. Based on this criterion, we evaluate the system performance by using the *throughput utility* metric, $U_{throughput}(t)$, which is identical to $A(t)$, i.e.,

$$U_{throughput}(t) = A(t). \tag{15}$$

Second, the system prefers *fairness*, i.e., the number of solved tasks must be in proportion to the popularity of each LOI. To assess this criterion, we use the *fairness utility* metric, $U_{fairness}(t)$, which is the inverse of $c.v.(t)$; that is,

$$U_{fairness}(t) = \frac{1}{c.v.(t)}. \quad (16)$$

To accommodate the two objectives, we design a third metric called *system utility*, $U_{system}(t)$, by taking the product of $U_{throughput}(t)$ and $U_{fairness}(t)$, as shown in Equation 17.

$$U_{system}(t) = U_{throughput}(t) \times U_{fairness}(t). \quad (17)$$

4 System Strategies

In this section, we compare five task assignment algorithms for generic GWAP-based geotagging systems, namely the *Random Assignment* algorithm (RA), the *Simple Assignment* algorithm (SA), the *Acceptance-Rate-First Assignment* algorithm (ARFA), the *Least-Throughput-First Assignment* algorithm (LTFA), and the *Hybrid Assignment* algorithm (HA). Let $COPY(Q_i)$ denote the function that returns a copy of the headmost task of Q_i; and let $POP(Q_i)$ denote the function that *pops up* and returns the headmost task from the task queue Q_i. We discuss the five algorithms in the following sub-sections.

4.1 Random Assignment Algorithm (RA)

The *Random Assignment* Algorithm (RA) assigns the headmost tasks of K distinct task queues at random. The computational complexity of the RA scheme is $O(N)$, and it provides the baseline performance of the system proposed in this study. The RA scheme has been implemented in several GWAP-based geotagging systems, such as *Eyespy* [8].

4.2 Simple Assignment Algorithm (SA)

The *Simple Assignment* Algorithm (SA) assigns the K headmost tasks from the current LOI's task queue. SA is the simplest scheme because its computational complexity is $O(1)$. Hence, it is adopted by most geospatial blogging systems when the requester and the solver are at the same LOI. The SA scheme has been implemented in many GWAP-based geotagging systems, such as *MobiMission* [18] and the *Gopher* game [11].

4.3 Acceptance-Rate-First Assignment Algorithm (ARFA)

The *Acceptance-Rate-First Assignment* Algorithm (ARFA) assigns the headmost tasks of the K distinct task queues that are most likely to be accepted by the solver, as shown in Algorithm 1. The computational complexity of the ARFA algorithm is approximately $O(NlogN)$, since it has to sort the willingness values of the headmost tasks from the N task queues. The algorithm is a greedy and best-effort approach that only tries to maximize the $U_{throughput}$ metric; therefore, it may cause the *starvation* problem, especially for unpopular LOIs (i.e., the willingness estimate decreases in line with the popularity factor, as shown in Equation 1).

Algorithm 1 The Acceptance-Rate-First Assignment (ARFA)

1: $V \Leftarrow \emptyset$
2: **for** $i = 1...N$ **do**
3: $\quad W[i].value \Leftarrow W(u, COPY(Q_i))$; $W[i].queue \Leftarrow i$
4: **end for**
5: Sort $W[i]$ by $value$ in descending order
6: **for** $i = 1...K$ **do**
7: $\quad V \Leftarrow V \bigcup POP(Q_{W[i].queue})$
8: **end for**
9: Return V

Algorithm 2 The Least-Throughput-First Assignment (LTFA)

1: $V \Leftarrow \emptyset$
2: **for** $i = 1...N$ **do**
3: $\quad T[i].value \Leftarrow \widetilde{T}(t, i)$; $T[i].queue \Leftarrow i$
4: **end for**
5: Sort $T[i]$ by $value$ in ascending order
6: **for** $i = 1...K$ **do**
7: $\quad V \Leftarrow V \bigcup POP(Q_{T[i].queue})$
8: **end for**
9: Return V

4.4 Least-Throughput-First Assignment Algorithm (LTFA)

The *Least-Throughput-First Assignment* Algorithm (LTFA) assigns the headmost tasks of the K distinct task queues that yielded the least throughput, as shown in Algorithm 2. For each LOI (i.e., $O(N)$ steps), the algorithm has to calculate the normalized system throughput in $O(K\lambda)$ steps, and sort the calculation results in $O(NlogN)$ steps; thus, the computational complexity of LTFA is approximately $O(NlogN + NK\lambda)$. However, in practice, N is far larger than K and λ, so the computational complexity is approximately $O(NlogN)$. Similar to ARFA, the LTFA algorithm is a heuristics-based approach that gives priority to the $U_{fairness}$ metric, which may result in the '*equality of outcome*' problem.

4.5 Hybrid Assignment Algorithm (HA)

The *Hybrid Assignment* Algorithm (HA) assigns the headmost task of the K distinct task queues that are most likely to yield the highest system utility, as shown in Algorithm 3. For each headmost task of N queues (i.e., $O(N)$ steps), the algorithm must calculate the system utility, which is a combination of $U_{throughput}$ (i.e., $O(N)$ steps) and $U_{fairness}$ (i.e., $O(N)$ steps). Hence, the overall computational complexity of the HA algorithm is approximately $O(N^2)$, which is the highest among the five compared algorithms. Although this algorithm provides the optimal solution for the system, its computational overhead is a drawback.

5 Evaluation

We use the *Monte Carlo Simulation Method* to evaluate the five task assignment algorithms in three scenarios: the exponential distribution scenario (EXP), the *Self-similar Least Action Walk* scenario (SLAW), and the Taipei city scenario (TPE). It is assumed that there are 400

Algorithm 3 The Hybrid Assignment (HA)

1: $V \Leftarrow \emptyset$
2: **for** $i = 1...N$ **do**
3: $v \Leftarrow COPY(Q_i)$
4: $U[i].value \Leftarrow$ calculate U_{system} assuming v is solved;
5: $U[i].queue \Leftarrow i$
6: **end for**
7: Sort $U[i]$ by $value$ in descending order
8: **for** $i = 1...K$ **do**
9: $V \Leftarrow V \bigcup POP(Q_{U[i].queue})$
10: **end for**
11: Return V

LOIs uniformly distributed on a 20 $\times$ 20 grid, and that players arrive at the i-th LOI at a Poisson rate of λ_i per time unit. In the scenarios, the value of λ_i is determined as follows:

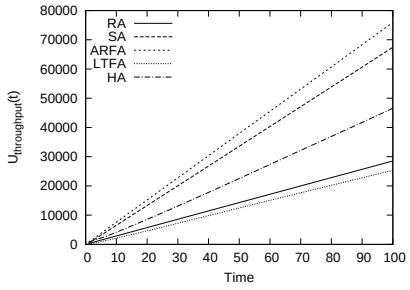
1. The EXP scenario: the value of λ_i is determined based on an exponential distribution with the rate parameter set to 0.2.
2. The SLAW scenario: we apply the SLAW model [22] to a 2,000 $\times$ 2,000 area, and set the SLAW parameters n_wp (i.e., the number of waypoints to be generated) and v_Hurst (i.e., the Hurst parameter used in *fractional Brownian motion*) at 2,000 and 0.75 respectively. After using the SLAW generator [7] to generate the waypoints, we divide the area into a 20 $\times$ 20 equal-sized grid and determine the value of λ_i by counting the number of waypoints in the i-th cell. The mean and the standard deviation of λ_i are 6 and 11.4926 respectively.
3. The TPE scenario: we use the map of central Taipei, which covers approximately 25 sq. km (north to *Taipei SongShan Airport*, east to *Taipei City Hall*, south to *National Taiwan University*, and west to *Taipei Main Station*), and divide it into a 20 $\times$ 20 equal-sized grid. To determine the value of λ_i, we count the number of bus stops in the i-th cell[2]. The mean and the standard deviation of λ_i are 3.4485 and 2.2394 respectively.

For convenience, we set the smoothing factor α (cf. Equation 5) at 0.95. Moreover, unless otherwise specified, we assume that two-thirds of the arriving players are requesters, and the other one-third are solvers (i.e., r=2). The distance threshold (cf. Equation 2) is set to five cell units (i.e., τ=5), and the size of each LOI's task queue is infinite. Each requester initiates a new task at his current LOI, and each solver is assigned K tasks in a round using one of the five task assignment algorithms. All the results presented here are based on the average performance of 100 simulation runs.

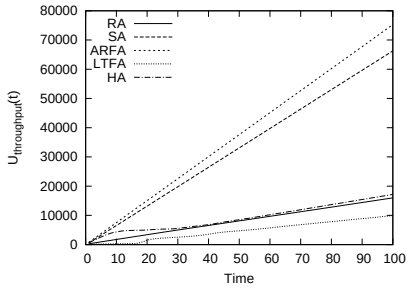
5.1 Evaluation results under the five schemes when $K = 1$

First, we evaluate the five task assignment algorithms in the three scenarios when the number of tasks assigned in each round, K, is equal to one. In Figures 3 and 4, we compare the $U_{throughput}$ and $U_{fairness}$ performance respectively. We observe that the ARFA scheme achieves the highest throughput performance, but it yields the worst fairness performance. This is because the scheme always assigns the tasks with the highest willingness estimates (i.e., Equation 1); thus, it is more productive than the other schemes, but at the expense

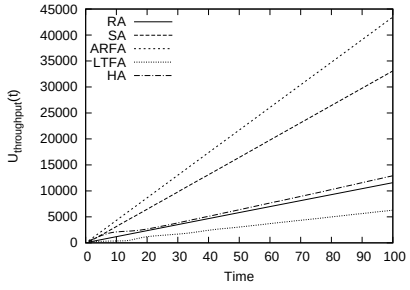
[2] Information about bus routes in Taipei city is available at http://www.e-bus.taipei.gov.tw/english/en_index_6_1.html



(a) EXP scenario

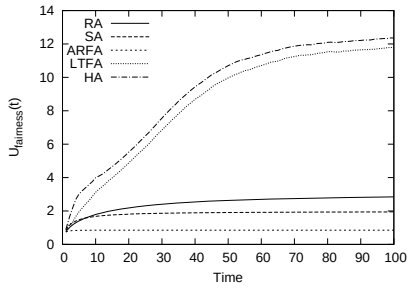


(b) SLAW scenario

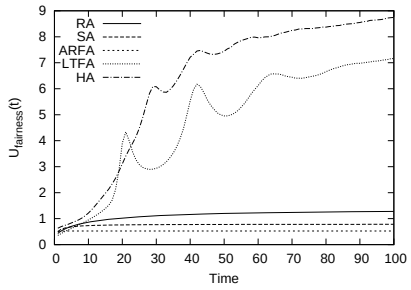


(c) TPE scenario

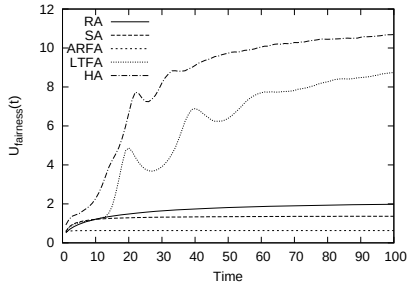
Fig. 3 The simulated $U_{throughput}$ performance in the three scenarios, when $K = 1$.



(a) EXP scenario

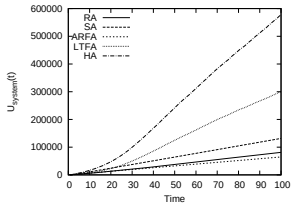


(b) SLAW scenario

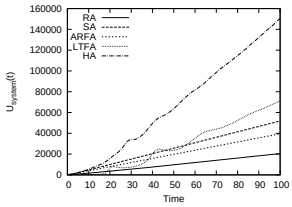


(c) TPE scenario

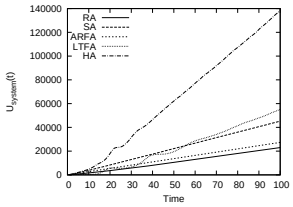
Fig. 4 The simulated $U_{fairness}$ performance in the three scenarios, when $K = 1$.



(a) EXP scenario



(b) SLAW scenario



(c) TPE scenario

Fig. 5 The simulated U_{system} performance in the three scenarios, when $K = 1$.

of increased unfairness. In contrast, because the LTFA scheme gives priority to fairness, it achieves a good coefficient of variation performance, as shown in Figure 4; however, it suffers from the *equality of outcomes* problem (i.e., lower throughput performance), as shown in Figure 3. The results in Figure 4 also show that the LTFA curve oscillates a great deal

Table 1 The time required by each algorithm to complete one simulation round in the three scenarios. (Unit: seconds)

	RA $O(N)$	SA $O(1)$	ARFA $O(NlogN)$	LTFA $O(NlogN)$	HA $O(N^2)$
EXP	14.90	10.93	234.27	424.67	3,280.70
SLAW	16.07	11.95	239.68	460.79	3,230.19
TPE	13.64	8.95	85.76	153.13	1,149.50

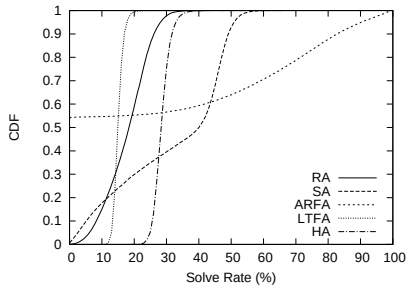
at the beginning of the SLAW and TPE scenarios. This is because its $c.v.(t)$ performance depends to a large extent on the accuracy of the λ estimation (cf. Equation 5), which needs a *warm-up* period to become stable and accurate.

By combining the two metrics, we can compare the system utility performance of the five algorithms, as shown in Figure 5. The results demonstrate that the HA scheme significantly outperforms the other schemes. We also observe that, although the RA and SA schemes represent the "golden mean" in Figures 3 and 4, they are not suitable because their system utility performance is poor. *In terms of computational complexity, LTFA is the most efficient scheme* (see Table 1), because its system utility performance is comparable to that of the HA scheme, but its computational complexity is moderate.

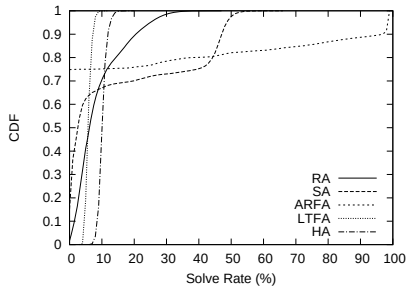
Figure 6 compares the distributions of the solve rates of the five algorithms in Cumulative Distribution Function (CDF) curves. The '*solve rate*' is the percentage of tasks solved at each LOI. The results show that under the ARFA scheme, more than 50% of the LOIs produce zero outcomes (i.e., the solve rate is zero), which confirms our intuition that the scheme tends to suffer from the *starvation* problem. Moreover, we observe that the curves of the HA and LTFA schemes are nearly vertical, while the others are slanted. The results also confirm our intuition that *the two schemes are better able to preserve the fairness of the normalized system throughput*. The HA scheme's solve rate is higher than that of the LTFA scheme because its throughput performance is better, as shown in Figure 3. Finally, we observe that the ARFA curve rises sharply in the SLAW and TPE scenarios, but not in the EXP scenario. This is because the SLAW and TPE scenarios are based on a 2D mobility model, whereas the EXP scenario is a 1D model. Consequently, the SLAW and TPE scenarios are better able to capture the spatial locality of popular LOIs. *Since the ARFA scheme can leverage this characteristic to find a task in nearby areas easily*, it increases the value of the distance factor and the willingness estimate.

The results in Figure 7 show that the system utility of the SA and ARFA schemes is consistent regardless of the value of τ; whereas the system utility of the RA, LTFA and HA schemes increases with τ. This is because the SA scheme always assigns the tasks that are at the same LOI as the solver, while the ARFA scheme always assigns the tasks that have the highest willingness values; therefore, the two schemes are invariant to changes in the value of τ. The other schemes are sensitive to the changes in τ because the willingness estimate that the solver will accept the assignment increases as τ increases (i.e., a higher τ value yields a higher distance factor value, and thus a greater willingness value). For instance, under the LTFA scheme, the solve rate curves shown in Figure 8 shift to the right as the values of τ increase (i.e., a larger number of task assignments are accepted and solved). In each scenario, the HA scheme outperforms the other schemes, and the LTFA scheme is ranked second, regardless of the values of τ.

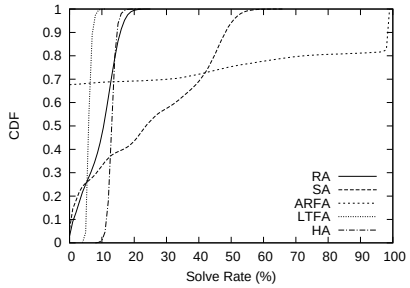
Finally, we observe the impact of r, i.e., the ratio of the number of requesters to the number of solvers, on the system utility performance. The results in Figure 9 show that, as the value of r increases, the system utility performance improves initially, and degrades



(a) EXP scenario

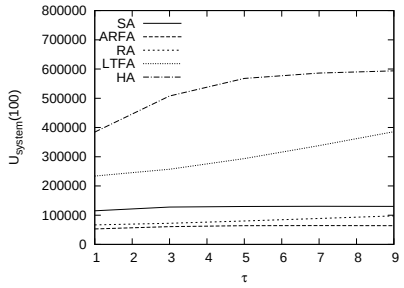


(b) SLAW scenario

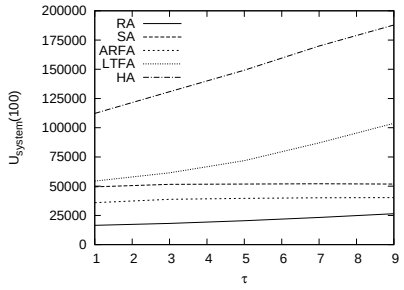


(c) TPE scenario

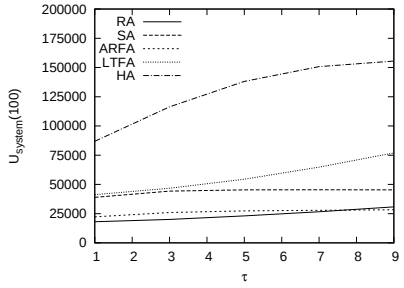
Fig. 6 Comparison of the CDF of the solve rate per LOI after 100 time units in the three scenarios when $K = 1$.



(a) EXP scenario



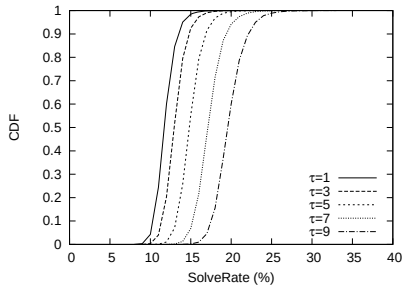
(b) SLAW scenario



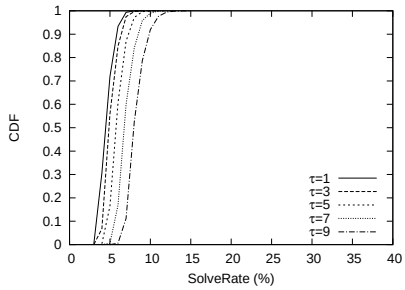
(c) TPE scenario

Fig. 7 Simulation results of the system utility performance with various τ values after 100 time units in the three scenarios when $K = 1$.

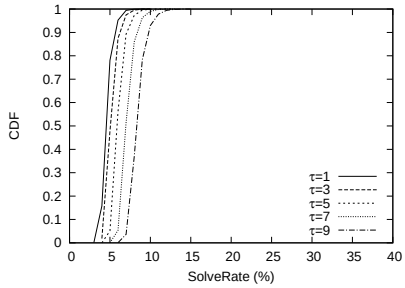
after r becomes larger than a certain number. For instance, under the HA scheme, the optimal value of r is equal to 1 and 2 in the 1D mobility model scenario (i.e., the EXP scenario) and the 2D mobility model scenarios (i.e., the SLAW and TPE scenarios) respectively; whereas, under the LTFA scheme, the optimal value of r is about 2 in the three scenarios. The reason is that, when r is very small (i.e., most players are solvers), the system is short of tasks to assign to solvers; whereas, when r is very large (i.e., most players are requesters), the system is short of solvers to solve tasks. As a result, *there is one 'optimal value' of r that can yield the highest system utility*. Moreover, *to ensure that the system is productive, it is important to provide 'just-enough' incentives to keep the value of r around the optimal value.*



(a) EXP scenario

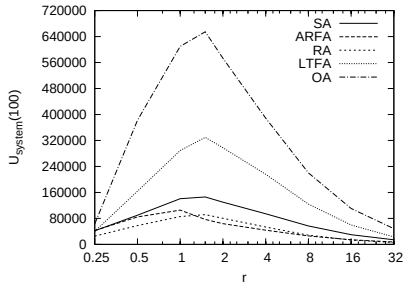


(b) SLAW scenario

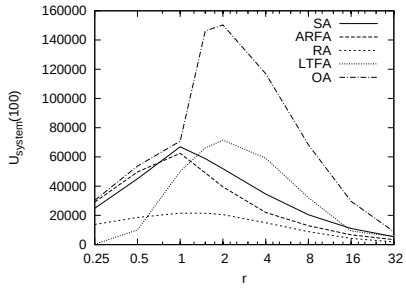


(c) TPE scenario

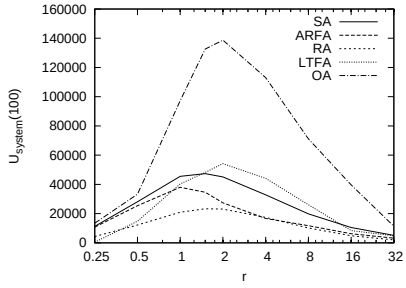
Fig. 8 Comparison of the CDF of the solve rate per LOI with various τ values in the three scenarios when $K = 1$ and the LTFA scheme is used.



(a) EXP scenario



(b) SLAW scenario



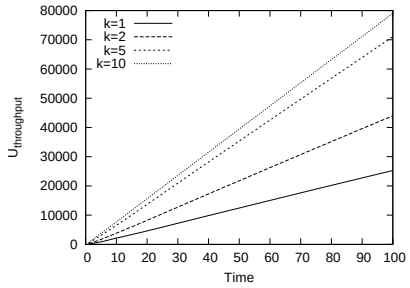
(c) TPE scenario

Fig. 9 Simulation results of the system utility performance after 100 time units and with various r values, when $K = 1$, in the three scenarios.

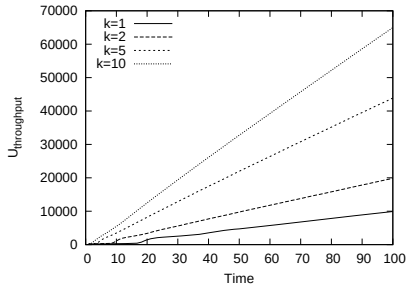
5.2 Evaluation results under the LTFA scheme when $K > 1$

As mentioned earlier, the LTFA scheme is the most suitable for GWAP-based geotagging systems because its system utility performance is competitive and its computational complexity is moderate. Therefore, in this subsection, we only evaluate the LTFA algorithm when the system assigns more than one task (i.e., $K > 1$) to a solver in each game round.

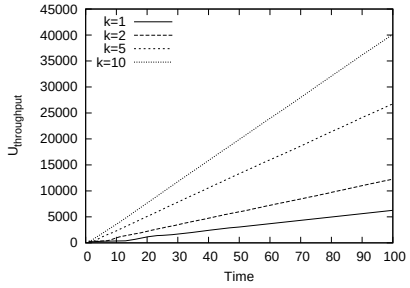
Figures 10, 11, 12, and 13 show the $U_{throughput}$ performance and $U_{fairness}$ performance when K tasks are assigned in a one-by-one manner and an all-at-once manner. Clearly, the system throughput increases with the value of K; hence, the larger the value of K, the faster the $U_{fairness}$ performance will converge. This is because the solver is more likely to contribute a solution to one of the assigned tasks when the value of K increases, so the system throughput will increase. Moreover, since the LTFA algorithm favors the fairness



(a) EXP scenario

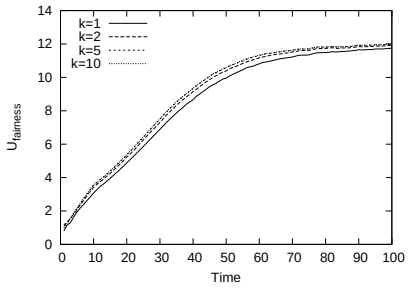


(b) SLAW scenario

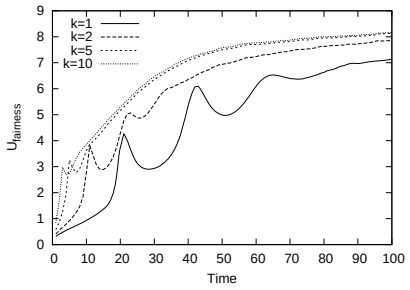


(c) TPE scenario

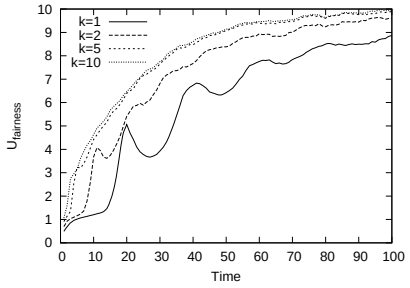
Fig. 10 Simulation results of the $U_{throughput}$ performance in the three scenarios, when $r = 2$ and the K tasks are assigned by the LTFA scheme in a *one-by-one* manner.



(a) EXP scenario



(b) SLAW scenario



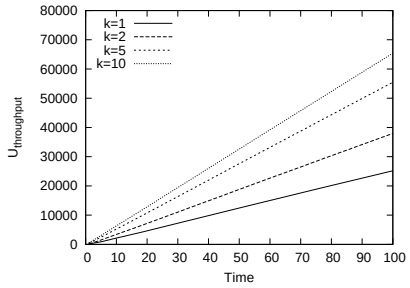
(c) TPE scenario

Fig. 11 Simulation results of the $U_{fairness}$ performance in the three scenarios, when $r = 2$ and the K tasks are assigned by the LTFA scheme in a *one-by-one* manner.

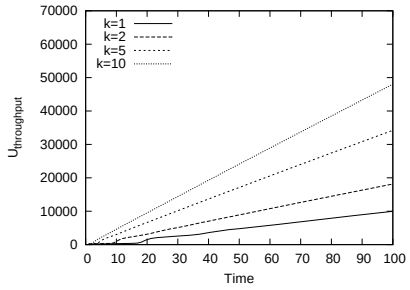
criterion when assigning tasks, the $U_{fairness}$ performance also converges to the equilibrium state more quickly.

Figure 14 shows that the system utility increases with the value of K regardless of the value of r. It is recommended that tasks be assigned in a one-by-one manner because the system utility is consistently higher than when tasks are assigned all at once. In addition, when the value of r is fixed at 2, Figure 15 shows that the system utility has a tendency to converge to an equilibrium state when K is sufficiently large, regardless of whether the K tasks are assigned in a one-by-one manner or an all-at-once manner. However, the value of K must not be too large in practice; otherwise, players will become annoyed and discouraged.

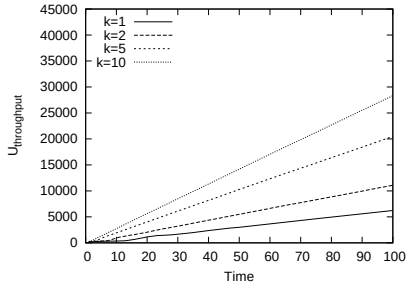
The results in Figure 14 also demonstrate that, as the value of r increases, the system utility performance improves at the beginning, and degrades after r becomes larger than a certain number. Moreover, the 'optimal' r value that yields the highest system utility per-



(a) EXP scenario

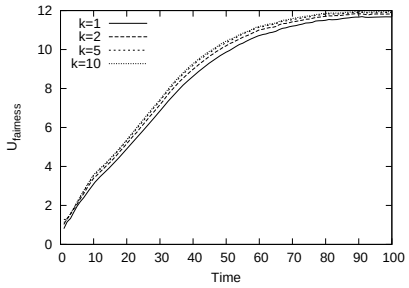


(b) SLAW scenario

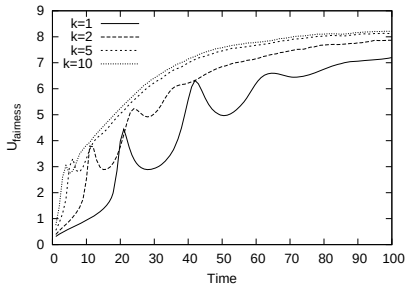


(c) TPE scenario

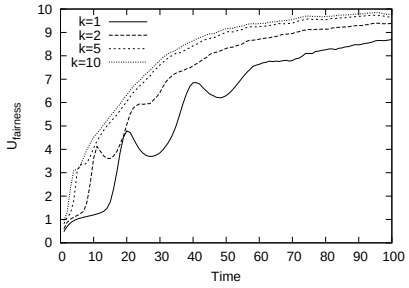
Fig. 12 Simulation results of the $U_{throughput}$ performance in the three scenarios, when $r = 2$ and the K tasks are assigned by the LTFA scheme in an *all-at-once* manner.



(a) EXP scenario



(b) SLAW scenario



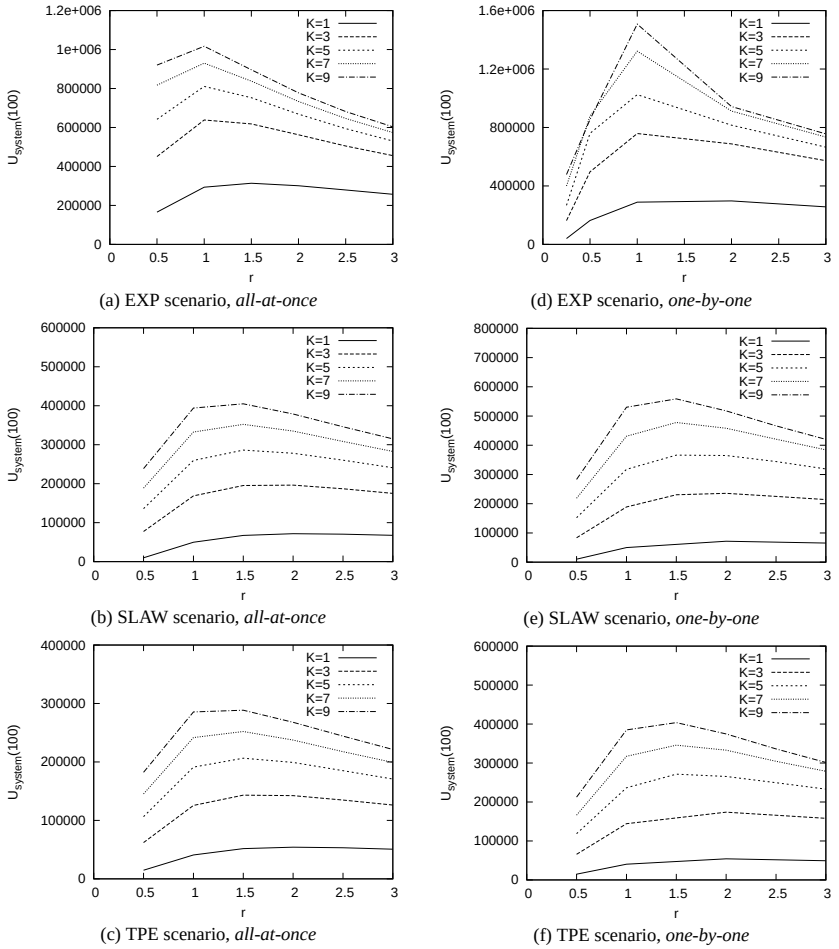
(c) TPE scenario

Fig. 13 Simulation results of the $U_{fairness}$ performance in the three scenarios, when $r = 2$ and the K tasks are assigned by the LTFA scheme in an *all-at-once* manner.

formance is consistent regardless of the value of K and whether the K tasks are assigned all-at-once or one-by-one. Specifically, the optimal value of r is equal to 1 and 1.5 in the 1D mobility model scenario (i.e., the EXP scenario) and the 2D mobility model scenarios (i.e., the SLAW and TPE scenarios) respectively. Again, the results confirm our previous finding that, *to ensure the system is productive, it is important to provide 'just-enough' incentives to keep the value of r around the optimal value*.

5.3 Evaluation results under the LTFA scheme with various buffer sizes

Next, we evaluate the impact of the buffer size per LOI on the performance of GWAP-based geospatial tagging systems in terms of the '*block rate*', i.e. the percentage of tasks that are blocked from entering the system because of buffer overflow. Due to space limitations, it is



(a) EXP scenario, *all-at-once*

(d) EXP scenario, *one-by-one*

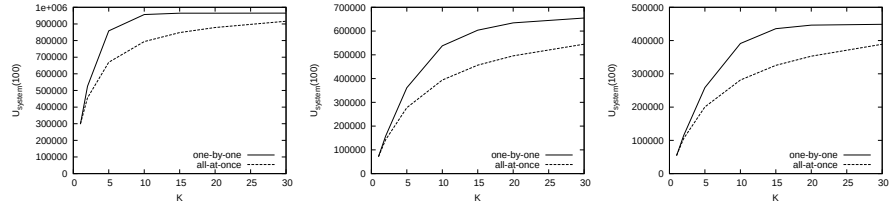
(b) SLAW scenario, *all-at-once*

(e) SLAW scenario, *one-by-one*

(c) TPE scenario, *all-at-once*

(f) TPE scenario, *one-by-one*

Fig. 14 Simulation results of the system utility performance with various values of K and r after 100 time units, when the K tasks are assigned by the LTFA scheme in an *all-at-once* manner and a *one-by-one* manner respectively.



(a) EXP scenario

(b) SLAW scenario

(c) TPE scenario

Fig. 15 Simulation results of the system utility performance with various values of K after 100 time units, when $r = 2$ and tasks are assigned in a one-by-one manner and an all-at-once manner.

not possible to provide a detailed evaluation of each task assignment algorithm with different system configurations. Instead, we only present the evaluation results of the LTFA scheme

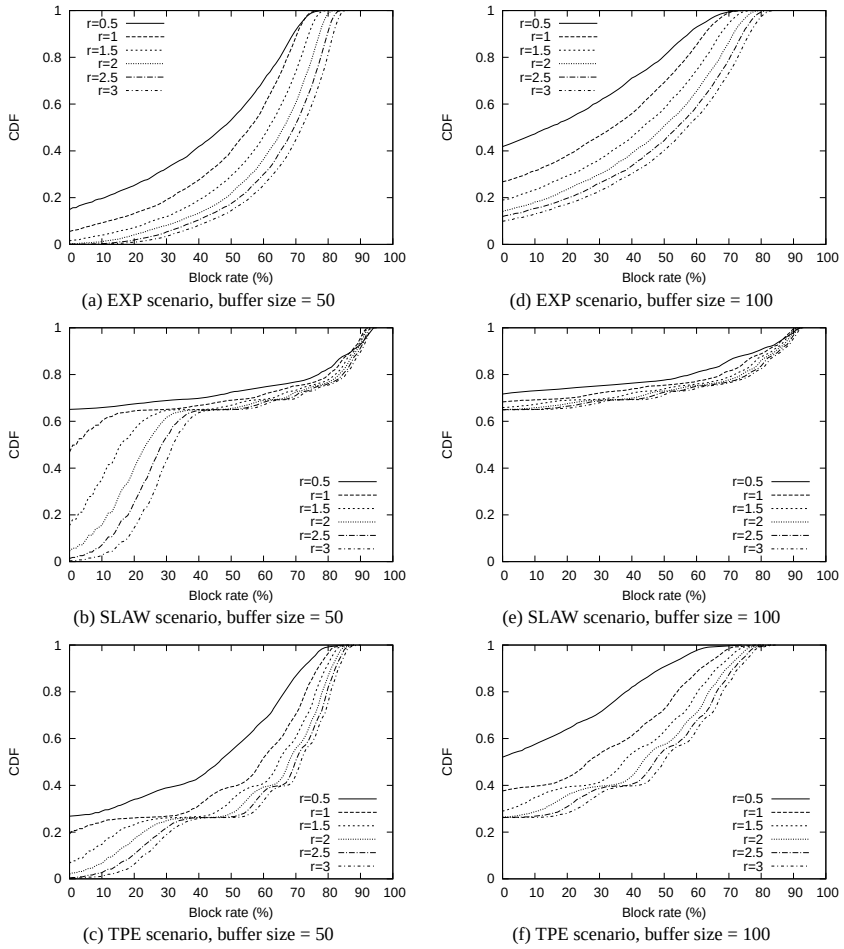
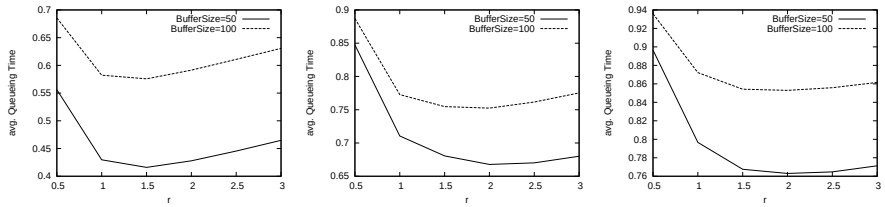


Fig. 16 Comparison of the CDF of the block rate per LOI after 100 time units with various r values and buffer sizes in the three scenarios when $K = 1$ under the LTFA scheme.

when $K = 1$, as it provides the baseline performance of the LTFA scheme. It has been shown that the scheme is the most effective solution for GWAP-based geospatial tagging systems.

Figure 16 compares the block rates in CDF curves with various r values when the buffer size is set at 50 and 100 tasks in the three scenarios. We observe that, in all scenarios, the curves are higher when the buffer size is larger and the value of r is smaller. The reason is simply that the larger the buffer size, the greater the likelihood that tasks will be allowed to enter the system. Moreover, given the same number of players in the system, the number of players that operate as requesters will increase as the value of r increases. Therefore, the 'birth rate' of tasks increases with the value of r, which increases the number of the tasks blocked due to buffer overflow.

However, we note that it is not advisable to have an extremely small value of r, as it may result in a '*short-of-tasks*' problem, i.e., the number of the tasks available is less than the number of solvers in the system. The results in Figure 17 show that the curve of the average queueing time per task (i.e., the time between when a task is input to the system until it is



(a) EXP scenario (b) SLAW scenario (c) TPE scenario

Fig. 17 Simulation results of the average queueing time of each task with various buffer sizes after 100 time units when $K = 1$ and $r = 2$ under the LTFA scheme.

completed by solvers) decreases initially, and then increases as the value of r increases. The reason is that, when the value of r is very small, there are not many tasks available in the system, so the probability that a solver will be assigned a nearby task decreases. As a result, solvers tend to reject the assigned tasks, and the average queueing time per task increases. Meanwhile, if the value of r is very large, the task birth rate becomes larger than the rate at which solvers can complete the tasks. As a consequence, most tasks have to be queued in the system longer before being assigned to solvers. Thus, based on the results shown in Figures 16 and 17, we suggest that the value of r should be maintained at approximately 1.5, which also confirms our previous results (cf. Sub-section 5.2).

The results in Figure 17 also show that, when a larger buffer size is used per LOI, the average queueing time increases. Again, this is because the larger the buffer size, the greater the likelihood that tasks will be able to enter the system, thereby reducing the block rate, as shown in Figure 16. However, on the downside, a long queueing time discourages players from staying with the GWAP system, as requesters may lose patience if each task's turnaround time is too long. Thus, it is essential to configure the buffer size to accommodate both factors (i.e., the block rate and the play experience). We defer a detailed investigation of this issue to a future work.

5.4 Discussion

We now consider some issues that may arise when applying our analysis in real GWAP-based geotagging systems. First, the proposed model only considers the throughput and fairness factors when measuring the performance of GWAP-based geotagging systems. As a result, it may not be sufficiently representative of the "productivity" and "quality" of the system. For instance, a solver may take one month to provide a high-quality solution to a task assigned by the system; or he may just provide a random outcome in five minutes. It may be possible to address this issue by incorporating two more factors into the model, namely, the time required to solve a task and the quality of each solution, which can be measured by using human computation [32, 34]. We defer a detailed evaluation of this issue to a future work.

Second, the proposed model assumes each task is a simple mission that involves only one LOI. It does not consider complex missions comprised of multiple LOIs (e.g., *Find all Indian restaurants on Hollywood Boulevard*) and/or the transitions among multiple LOIs (e.g., *What is the fastest way to get to the airport from downtown in rush hour?*). Obviously, the model will require substantial modification so that it can handle such complex missions. For example, it is necessary to 1) redefine the concept of system utility when a task is comprised of multiple LOIs; and 2) provide both quantitative and qualitative metrics to

evaluate the transitions among multiple LOIs. Again, we defer a detailed discussion of this issue to a future work.

6 Conclusion

In this paper, we have studied emerging GWAP-based geotagging systems, presented an analysis of their intrinsic properties, and proposed three metrics to measure the system performance. Based on our analysis, we designed five task assignment algorithms that incorporate different heuristics and optimization strategies. Using a comprehensive set of simulations of synthetic and realistic mobility scenarios, we found that the *Hybrid Assignment* (HA) algorithm achieves the best system utility performance, but it is too computation-hungry to be deployed. In practice, the *Least-Throughput-First Assignment* (LTFA) algorithm is the most suitable scheme because its computational complexity is moderate, and its overall performance is comparable to that of the HA scheme. Moreover, to improve the system utility, it is better to assign as many tasks as possible in each game round; however, assigning too many tasks at the same time may annoy players. Finally, when multiple tasks are assigned in each round, it is better to assign them in a one-by-one manner, rather than an all-at-once manner, in order to maximize the system utility. The proposed analysis is simple and applicable to emerging GWAP-based geotagging systems. We believe the results of this study could improve the design and implementation of GWAP-based geotagging systems.

Acknowledgements

We wish to thank the editors and anonymous reviewers for their insightful comments and suggestions. This study is based on research supported by the National Science Council of Taiwan under NSC Grants: NSC 98-2221-E-001-014-MY3 and NSC 98-2631-H-001-013.

References

1. Carcassonne. http://www.carcassonne.de/.
2. Foursquare. http://foursquare.com/.
3. GeoTagging Flickr. http://www.flickr.com/groups/geotagging/.
4. Google Maps. http://maps.google.com/.
5. Gowalla. http://gowalla.com/.
6. Gypsii. http://www.gypsii.com/.
7. SLAW Generator. http://netsrv.csc.ncsu.edu/.
8. M. Bell, S. Reeves, B. Brown, S. Sherwood, D. MacMillan, J. Ferguson, and M. Chalmers. Eyespy: Supporting navigation through play. In *ACM SIGCHI*, 2009.
9. S. H. Bokhari. *Assignment Problems in Parallel and Distributed Computing*. Springer, 1987.
10. T. L. Casavant and J. G. Kuhl. A taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Transactions on Software Engineering*, 14(2):141–154, Feburary 1988.
11. S. Casey, B. Kirman, and D. Rowland. The gopher game: a social, mobile, locative game with user generated content and peer review. In *International Conference on Advances in Computer Entertainment Technology*, 2007.

12. L.-J. Chen, Y.-S. Syu, B.-C. Wang, and W.-C. Lee. An Analytical Study of GWAP-based Geospatial Tagging Systems. In *IEEE CollaborateCom*, 2009.
13. L.-J. Chen, B.-C. Wang, and K.-T. Chen. The Design of Puzzle Selection Strategies for GWAP Systems. *Journal of Concurrency and Computation: Practice and Experience, John Wiley & Sons Ltd.*, 22(7):890–908, May 2010.
14. A. Drozd, S. Benford, N. Tandavanitj, M. Wright, and A. Chamberlain. Hitchers: Designing for Cellular Positioning. In *UbiComp*, 2006.
15. J. Froehlich, J. Neumann, and N. Oliver. Measuring the Pulse of the City through Shared Bicycle Programs. In *ACM UrbanSensing*, 2008.
16. N. Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, 1998.
17. J. Goldman, K. Shilton, J. Burke, D. Estrin, M. Hansen, N. Ramanathan, S. Reddy, V. Samanta, M. Srivastava, and R. West. Participatory Sensing: A Citizen-powered Approach to Illuminating the Patterns That Shape our World. *Foresight & Governance Project, White Paper*, 2009.
18. L. Grant, H. Daanen, S. Benford, A. Hampshire, A. Drozd, and C. Greenhalgh. MobiMissions: the game of missions for mobile phones. In *ACM SIGGRAPH*, 2007.
19. T. Horanont and R. Shibasaki. An Implementation of Mobile Sensing For Large-Scale Urban Monitoring. In *ACM UrbanSensing*, 2008.
20. J. Howe. The Rise of Crowdsourcing. *WIRED Magazine*, 14(6), June 2006.
21. E. A. Lee. Cyber physical systems: Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, Jan 2008.
22. K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. SLAW: A Mobility Model for Human Walks. In *IEEE Infocom*, 2009.
23. H. Lieberman, D. Smith, and A. Teeters. Common Consensus: a web-based game for collecting commonsense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.
24. V. M. Lo. Heuristic algorithms for task assignment in distributed systems. *IEEE Transactions on Computers*, 37(11):1384–1397, November 1988.
25. H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. SoundSense: Scalable Sound Sensing for People-Centric Sensing Applications on Mobile Phones. In *ACM/USENIX MobiSys*, 2009.
26. S. Matyas. Playful geospatial data acquisition by location-based gaming communities. *The International Journal of Virtual Reality*, 6(3):1–10, 2007.
27. S. Matyas, C. Matyas, C. Schlieder, P. Kiefer, H. Mitarai, and M. Kamata. Designing Location-based Mobile Games With A Purpose: Collecting Geospatial Data with CityExplorer. In *ACM ACE*, 2008.
28. K. Patel, M. Ismail, S. Motahari, D. J. Rosenbaum, S. T. Ricken, S. A. Grandhi, R. P. Schuler, and Q. Jones. MarkIt: Community Play and Computation to Generate Rich Location Descriptions through a Mobile Phone Game. In *Hawaii International Conference on System Sciences*, 2010.
29. C. Schlieder. Representing the Meaning of Spatial Behavior by Spatially Grounded Intentional Systems. *Lecture Notes in Computer Science*, 3799:30–44, 2005.
30. C. Schlieder, P. Kiefer, and S. Matyas. Geogames: Designing Location-Based Games from Classic Board Games. *IEEE Intelligent Systems*, 21(5):40–46, 2006.
31. J. Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday, May 2004.
32. L. von Ahn. Games with a Purpose. *IEEE Computer*, 39(6):92–94, June 2006.

33. L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *ACM SIGCHI*, 2004.
34. L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.
35. L. von Ahn, M. Kedia, and M. Blum. Verbosity: A Game for Collecting Common-Sense Facts. In *ACM SIGCHI*, 2006.
36. M.-C. Yuen, L.-J. Chen, and I. King. A survey of human computation systems. In *IEEE Symposium on Social Computing Applications*, 2009.