

# The Design of Puzzle Selection Strategies for ESP-like GWAP Systems

Ling-Jyh Chen, *Member, IEEE*, Bo-Chun Wang, and Wen-Yuan Zhu

**Abstract**—The ‘Games With A Purpose’ (GWAP) genre is a type of ‘Human Computation’ that outsources certain steps of the computational process to humans. Although most GWAP studies focus on the design and analysis of GWAP systems, a systematic and thorough evaluation of existing systems is lacking. We address the issue in this paper. Taking the ESP game as an example, we propose a metric, called *system utility*, for evaluating the performance of GWAP systems, and use analysis to study the properties of the ESP game. We argue that GWAP systems should be designed and played with strategies. To this end, based on our analysis, we implement an Optimal Puzzle Selection Strategy (OPSA) to improve GWAP systems. Using a comprehensive set of simulations, we show that the proposed OPSA approach can improve the system utility of the ESP game significantly. In addition, we implement a quasi ESP game, called *ESP Lite*, which embeds three puzzle selection algorithms transparently and records the complete game trace for evaluation and further research. During a one-month experiment, we have investigated the inner properties of the three strategies in real-world GWAP systems, and verified that the OPSA scheme achieves the best system utility for the ESP game. The results of this study demonstrate that GWAP systems are more efficient if they are designed and played with strategies.

**Index Terms**—Games with a Purpose; Human Computation; Tagging.

## I. INTRODUCTION

*Games With A Purpose* (GWAP) [35, 37] represent a new paradigm of applications that leverage people’s desire to be entertained and also outsource certain steps of the computational process to humans [23, 25, 27, 34]. By exploiting “human cycles” in computation, GWAP not only attract people to play voluntarily, but also produce useful metadata as a by-product. The genre has shown promise in solving a variety of problems, such as image annotation [31, 36, 41], audio annotation [8, 9, 11, 26], and commonsense reasoning [28, 40], which computer programs have been unable to resolve completely thus far.

Several GWAP systems have been proposed in recent years [28, 31, 36, 38–41]. Among them, the ESP Game [36] was the first to successfully realize the advantages of GWAP systems.

A preliminary version of this study was published in the IEEE/WIC/ACM International Conference on Web Intelligence, 2008 [18]. In this extended version paper, we implemented a ‘quasi’ ESP game, called *ESP Lite*, and performed comprehensive evaluations of the three puzzle selection strategies in real-world systems. Hence, this manuscript is a much more thorough and authoritative presentation of our study on playing strategies of GWAP systems.

Manuscript received XXX XX, 2009.

L.-J. Chen and W.-Y. Zhu are with the Institute of Information Science, Academia Sinica, 128, Sec. 2, Academia Road, Taipei 11529, Taiwan.

B.-C. Wang is with the Department of Computer Science, University of Southern California. This work is done when he was affiliated with the Institute of Information Science, Academia Sinica.

The rationale behind the ESP game is to motivate people to label images because it is fun. It has been shown that the image labels collected through the ESP game are typically of good quality. Moreover, the game results allow more accurate image retrieval, help users block inappropriate (e.g., pornographic) images, and improve web accessibility (e.g., the labels can help visually impaired people surf web pages [14]).

To be effective, the ESP game should satisfy two goals. First, it should take as many labels as possible for each puzzle, so that fewer distinct puzzles will be played. Second, each puzzle should only be played once in order to maximize the number of puzzles played. Clearly, both factors are critical to the performance of GWAP systems, but unfortunately they do not complement each other.

We believe that, to better accommodate both factors, GWAP systems should be *designed and played with strategies*. Specifically, in this work, using the ESP game as an example, we define a metric, called *system utility*, to evaluate the performance of GWAP systems. The proposed metric considers two performance aspects: *the number of puzzles that have been played in the system*, and *the average aggregated score of the agreements reached in each puzzle*. Based on our analysis, we propose an *Optimal Puzzle Selection Algorithm* (OPSA) that can maximize the system utility by properly accommodating the two opposing factors. Using a set of simulations, we investigate the properties of the ESP game, and evaluate the proposed OPSA scheme against two other schemes, namely, the *Random Puzzle Selection Algorithm* (RPSA) and the *Fresh-first Puzzle Selection Algorithm* (FPSA). The results demonstrate that, under the OPSA scheme, the system utility of the ESP game is much better than under the two compared schemes. In addition, to evaluate the three playing strategies in real-world GWAP systems, we implement a system called *ESP Lite*, which mimics the ESP game while embedding the three playing strategies transparently, and recording the complete game trace for evaluation and further research. We perform real-world experiments using the *ESP Lite* system, and we verify that the OPSA scheme can improve the system utility and better accommodate both performance objectives in the ESP game.

The contribution of this work is three-fold. First, to the best of our knowledge, this is the first GWAP study that proposes, implements and evaluates an analytical model on real-world GWAP systems. Second, our evaluation results demonstrate that GWAP systems are more efficient if they are designed and played with strategies. Finally, the proposed analysis is simple and applicable to other ESP-like GWAP systems, and the proposed puzzle selection strategy shows promise for use

in the design and implementation of future GWAP systems.

The remainder of this paper is organized as follows. Section II contains a review of related works on GWAP systems. In Section III, we describe the rules of the ESP game and present our analysis. In Section IV, we compare three puzzle selection algorithms for the ESP game, namely the RPSA, FPSA, and OPSA schemes. Section V contains the simulation results. In Section VI, we present the design and implementation of the *ESP Lite* system. In Section VII, we provide a comprehensive set of experiment results, which we analyze and explain in detail. We then summarize our conclusions in Section VIII.

## II. BACKGROUND

The concept of GWAP was pioneered by Luis von Ahn and his colleagues, who created *games with a purpose* [35], which people play voluntarily and produce useful metadata as a by-product. By taking advantage of people’s desire to be entertained, GWAP has shown promise in solving some problems that computer computation cannot currently resolve completely. In recent years, a substantial and increasing amount of research effort has been invested in the area, and several GWAP systems have been developed for a variety of purposes, e.g., [6, 8, 9, 11–13, 17, 21, 26, 28, 30, 31, 33, 36, 38–41, 43].

Among them, the online ESP Game<sup>1</sup> [36] was the first GWAP system, and it was subsequently adopted as the *Google Image Labeler* [6]. The game is fast-paced, enjoyable, and competitive<sup>2</sup> [37]. Moreover, it has been shown that the collected labels facilitate more accurate image retrieval, help users block inappropriate images, and improve web accessibility.

In addition, the *Peekaboom* system [41] can help determine the location of objects in images; and the *Squigl* system [7] and the *LabelMe* system [31] can provide complete outlines of the objects in an image. *Phetch* [38, 39] can provide image descriptions that improve web accessibility and image searches, while the *Matchin* system [7] can help image search engines rank images based on which ones are deemed the best. The concept of the ESP Game has been applied to other problems. For instance, the *Herd It* [8, 11], *Major Miner* [9], and *TagATune* [26] systems, which provide annotation for sounds and music, can improve audio searches. The *Verbosity* system [40] and the *Common Consensus* system [28] collect “common-sense” knowledge that is valuable for common-sense reasoning and enhancing the design of interactive user interfaces. In [13], Bennett et al. collect individual users’ preferences for image-search results, and extract consensus rankings from the preferences for the results of a query. The *Context-Aware Recognition Survey* (CARS) system [43] uses ubiquitous sensors to monitor activities in the home, while [12, 17, 21, 30] employ mobile social gaming techniques for geospatial tagging. Moreover, [33] applies human computation to ontology alignment and web content annotation for the

Semantic Web using a set of games, such as *OntoPronto*, *SpotTheLinks*, *OntoTube*, and *OntoBay*. Finally, Shenoy and Tan [32] showed that it is possible to design environments in which humans cannot avoid processing some of the tasks (and producing some useful outcomes), even though they are not actively trying to do so.

In addition to designing new GWAP systems, several studies have investigated the performance aspect of human computation [20, 22, 24, 37, 42]. For example, to solve the coalition problem, Ho et al. [22] proposed integrating both collaborative and competitive elements in image labeling games. Gentry et al. [20] proposed a framework of vote-based human computation and provided a probabilistic analysis of the reliability of the voting mechanism and design principles on the payout function. Moreover, in [42], Weber et al. presented a machine learning-based model that can play the ESP game without looking at the image. Based on the model, the authors proposed an enhanced scoring system for the ESP game to encourage users to contribute less predictable labels and thereby improve the quality of the collected labels. Jain and Parkes [24] conducted game theoretic analysis of the ESP game. They also investigated the equilibrium behavior under different incentive mechanisms and provided guidelines for the design of incentive mechanisms. Von Ahn [37] proposed a set of evaluation metrics, such as throughput, lifetime play, and expected contribution, to determine whether ESP-like GWAP systems are successful.

## III. OVERVIEW OF THE ESP GAME

### A. Game Description

After logging into the ESP game, the user is automatically matched with a random partner. The two players do not know each other’s identity because they cannot communicate. Initially, a randomly selected image is displayed to both players simultaneously. The players then input possible words to label the image until an “agreement” is reached (i.e., the same word is entered by both players); or they can decide to pass over a puzzle if they both think it is too difficult. Once an agreement has been reached, a bonus score is awarded to each player based on the ‘*quality*’ of the agreed word. In practice, the ‘*quality*’ of a word is measured by its popularity; generally, words that are more popular receive lower scores. After the players agree on a word, they are shown another image. In each game, they have two and a half minutes to label 15 images.

The word that the two players agree may become the official label of the image if enough people agree (the threshold of which depends on the game’s statistics). Note that the word (called a “taboo” word of the image) cannot be used the next time that image is displayed in another game. The rationale for using taboo words is to ensure that each image is labeled with a variety of words.

To be effective, the ESP game tries to collect the outcomes with *the largest possible aggregated score for each puzzle (image)*; hence, it needs *as many distinct puzzles as possible to be played*. There is a trade-off between these two objectives. First, it should take as many labels as possible for each puzzle, so that fewer distinct puzzles will be played. Second, each

<sup>1</sup>The game is named ‘ESP’ because the players have to work together to solve the tasks without talking to each other, i.e., using so called *Extra-Sensory Perception* (ESP) [36].

<sup>2</sup>As of July 2008, more than 200,000 players had contributed more than 50 million labels. Each player plays for a total of 91 minutes on average, and the throughput is about 233 labels per player per hour (i.e., one label every 15 seconds) [37].

puzzle should only be played once in order to maximize the number of puzzles played. Thus, an optimal puzzle selection 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 [15, 16, 29], and propose a metric to evaluate the *system utility* of the ESP game. We present an analysis of the puzzle selection problem in the following subsections.

### B. Game Analysis

Let  $N$  be the number of the puzzles that have been played at least once in the system, and let  $S$  be the average aggregated score of the agreements reached in each puzzle. We define the system utility,  $U$ , of the ESP game as follows:

$$U = \ln(N) \times \ln(S). \quad (1)$$

Specifically,  $\ln(N)$  and  $\ln(S)$  denote, respectively, the two performance aspects of the ESP game described earlier<sup>3</sup>. The metric  $U$  is designed to evaluate how well the ESP game accommodates both performance aspects simultaneously.

Clearly, the system utility increases as the number of the games played increases, and as the average aggregated score (per puzzle) increases. Suppose that  $N$  puzzles have been played  $T$  rounds in total (one puzzle per round), each puzzle has been played  $r$  times on average, and  $E[L]$  is the expected score of each agreed label. Since we know that  $N = T/r$  and  $S = E[L] \times r$ , we can rewrite Equation 1 as follows:

$$\begin{aligned} U &= \ln(T/r) \times \ln(E[L] \times r) \\ &= (\ln(T) - \ln(r)) \times (\ln(E[L]) + \ln(r)) \\ &= -(\ln(r))^2 + (\ln(T) - \ln(E[L]))\ln(r) + \ln(T)\ln(E[L]) \\ &= -\left(\ln(r) - \frac{\ln(T) - \ln(E[L])}{2}\right)^2 + C, \end{aligned} \quad (2)$$

where  $C$  is fixed with a value equal to  $\ln(T)\ln(E[L]) + \frac{(\ln(T) - \ln(E[L]))^2}{2}$ . Note that  $C$  also represents the largest possible system utility, which occurs when

$$r = e^{\frac{\ln(T) - \ln(E[L])}{2}}. \quad (3)$$

## IV. PUZZLE SELECTION ALGORITHMS

In this section, we compare three puzzle selection algorithms for the ESP game, namely, the *Random Puzzle Selection Algorithm* (RPSA), the *Fresh-first Puzzle Selection Algorithm* (FPSA), and the proposed *Optimal Puzzle Selection Algorithm* (OPSA). We take the RPSA scheme's performance as the baseline (in terms of system utility). The heuristics-based FPSA scheme tries to maximize the first component of Equation 1 (i.e.,  $\ln(N)$ ), while the OPSA scheme tries

<sup>3</sup>Note that we use the *natural logarithmic scale* for both factors in  $U$  because it has been shown that the *logarithmic scale* is more intuitive and appropriate in number-space mapping [19]. Moreover, the *natural logarithmic scale* has several properties (e.g., derivatives and Taylor series) that could be useful for further analysis.

---

**Algorithm 1** The Random Puzzle Selection Algorithm (RPSA).

---

- 1: **Function** RPSA
  - 2:  $p \leftarrow \text{Select\_Random}(P)$
  - 3: **Return**  $p$
- 

---

**Algorithm 2** The Fresh-first Puzzle Selection Algorithm (FPSA).

---

- 1: **Function** FPSA
  - 2:  $p \leftarrow \text{Select\_Fresh}(P)$
  - 3: **Return**  $p$
- 

to achieve the largest possible system utility based on our analysis<sup>4</sup> (discussed in Sec. III).

We use  $P$  to denote the set of all puzzles in the system, and define the following three functions used by the puzzle selection algorithms: 1) *Select\_Random*( $P$ ), which randomly selects a puzzle from the input puzzle set  $P$ ; 2) *Select\_Played*( $P$ ), which selects the puzzle in the input puzzle set  $P$  that has been played most frequently; and 3) *Select\_Fresh*( $P$ ), which selects the puzzle in the input puzzle set  $P$  that has been played least frequently.

### A. RPSA and FPSA

The steps of the *Random Puzzle Selection Algorithm* (RPSA) and the *Fresh-first Puzzle Selection Algorithm* (FPSA) are shown in Algorithms 1 and 2 respectively. RPSA selects a puzzle at random from the puzzle pool  $P$  in each round.<sup>5</sup> As mentioned earlier, it provides the baseline performance of the ESP game in this study. FPSA, on the other hand, selects the puzzle that has been played least frequently in the system. It is a greedy, heuristics-based approach that tries to maximize the first component of Equation 1.

### B. The Proposed Scheme: OPSA

In the proposed *Optimal Puzzle Selection Algorithm* (OPSA) for ESP games,  $N$  denotes the number of puzzles that have been played in the system,  $E$  denotes the expected score of each label, and  $T$  is the total number of rounds that have been played. In addition,  $r$  denotes the optimal number of rounds (discussed in Sec. III); and for each entry  $p$  of  $P$ ,  $p.r$  represents the number of rounds that the puzzle  $p$  was played. Suppose the puzzle set  $P_0$  contains all the puzzles that have not been played;  $P_1$  contains all the puzzles that have been played at least once, but less than  $r$  rounds; and the set  $P_2 = P - P_0 - P_1$  contains the other puzzles. We detail the steps of the OPSA algorithm in Algorithm 3.

<sup>4</sup>In this paper, we only consider two factors to evaluate the system utility  $U$  (i.e., the number of puzzles played and the average score of the agreed words). The proposed OPSA algorithm is designed to only accommodate the two factors. However, the analytical framework presented in this work can be easily applied to derive an 'optimal' solution as long as the notion of 'optimality' is known and can be clearly defined.

<sup>5</sup>The random puzzle selection algorithm is implemented in the ESP game [36], but it was called RPSA for the first time in [18].

**Algorithm 3** The Optimal Puzzle Selection Algorithm (OPSA).

---

```

1: Function OPSA
2:  $T \leftarrow T + 1$ 
3:  $r' \leftarrow \lceil e^{\frac{\ln(T) - \ln(E)}{2}} \rceil$ 
4: if  $r' > r$  then
5:   for each  $p$  in  $P_2$  do
6:     if  $p.r < r'$  then
7:       Move  $p$  from  $P_2$  to  $P_1$ 
8:     end if
9:   end for
10:   $P_1 \leftarrow P_1 \cup P_2$ 
11:   $r \leftarrow r'$ 
12: end if
13: if  $\{P_1\}$  is NOT empty then
14:   $p \leftarrow \text{Select\_Played}(P_1)$ 
15:   $p.r \leftarrow p.r + 1$ 
16:  if  $p.r = r$  then
17:    Move  $p$  from  $P_1$  to  $P_2$ 
18:  end if
19:  Return  $p$ 
20: else
21:  if  $\{P_0\}$  is NOT empty then
22:     $p \leftarrow \text{Select\_Random}(P_0)$ 
23:     $p.r \leftarrow 1$ 
24:    if  $p.r < r$  then
25:      Move  $p$  from  $P_0$  to  $P_1$ 
26:    else
27:      Move  $p$  from  $P_0$  to  $P_2$ 
28:    end if
29:    Return  $p$ 
30:  else
31:     $p \leftarrow \text{Select\_Fresh}(P_2)$ 
32:     $p.r \leftarrow 1$ 
33:    Return  $p$ 
34:  end if
35: end if

```

---

## V. SIMULATIONS

In this section, we discuss the simulations performed to investigate the properties of the ESP game based on our analysis. We also evaluate the *system utility* of the three puzzle selection strategies. All the results are based on the average performance of 200 simulations.

### A. The Optimal $r$ Value

In the first set of simulations, we evaluated the accuracy of our analytical model in determining the optimal  $r$  value for the ESP game. We assumed that the number of puzzles in the system was infinite, and all of them were unsolved at the beginning of the simulation (i.e., no labels were discovered for any puzzles). Moreover, we set the total number of game rounds played ( $T$ ) at 10,000. Figure 1 shows the evaluation results in terms of *system utility* for  $r$  values between 1 and 200, when the score value  $E[L]$  was fixed at 10.3353, i.e., the same as the value used in [18]. In the figure, the analysis curve

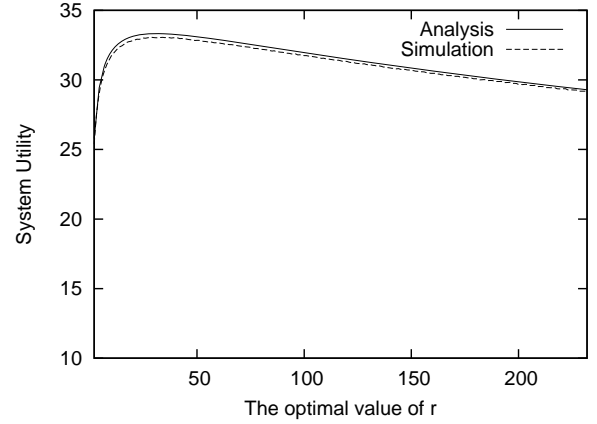


Fig. 1. Comparison of the system utility under various  $r$  settings in both the simulations and the analysis. ( $E[L] = 10.3353$  and  $T = 10,000$ )

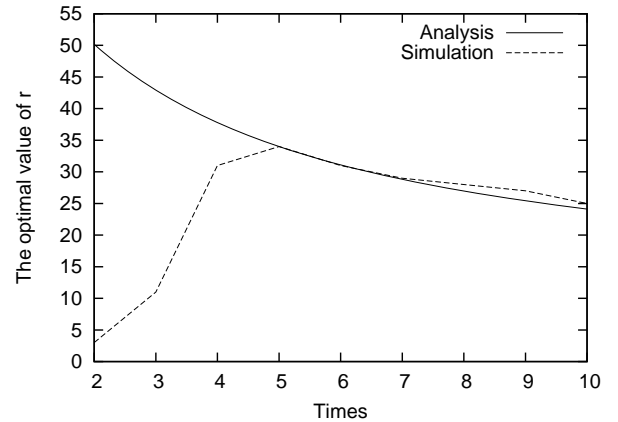


Fig. 2. Comparison of the optimal  $r$  values derived by simulations and analysis, where  $T = 10,000$  and  $E[L]$  varies from 1 to 10 times of 10.3353.

is derived by Equation 2. We observe that the curve matches the simulation curve very well, and the optimal  $r$  values (i.e., those that yielded the largest *system utility*) of the two curves are also comparable.

Additionally, we compared the derived optimal  $r$  values with different  $E[L]$  values varying from 1 to 10 times of 10.3353 using both simulations and analysis, as shown in Figure 2. We observe that, the optimal  $r$  value decreases as the  $E[L]$  value increases, which is consistent with Equation 3. We find that if more scores are generated in each round, fewer rounds of each puzzle need be played to achieve better overall *system utility*.

### B. The Relationship between $T$ , $N$ , and $r$

Next, we evaluate the relationship between the total number of game rounds  $T$ , the number of played puzzles  $N$ , and the number of game rounds required to maximize the *system utility*  $r$  in the proposed OPSA scheme. Figures 3 and 4 show the comparison results of  $r$  and  $N$  with various  $T$  values in the range 200 to 20,000 ( $E[L]$  is fixed at 10.3353).

The results show that our analytical model matches the simulation results very well in all cases. In addition, we observe that the values of  $r$  and  $N$  increase as the value of  $T$

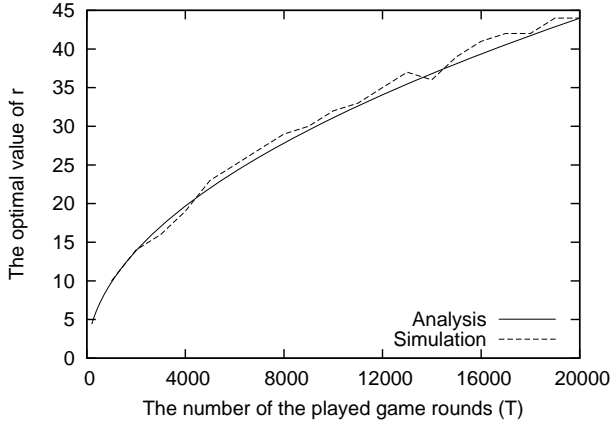


Fig. 3. Comparison of the optimal  $r$  values derived by simulations and analysis, where  $E[L] = 10.3353$  and  $T$  varies between 200 and 20,000.

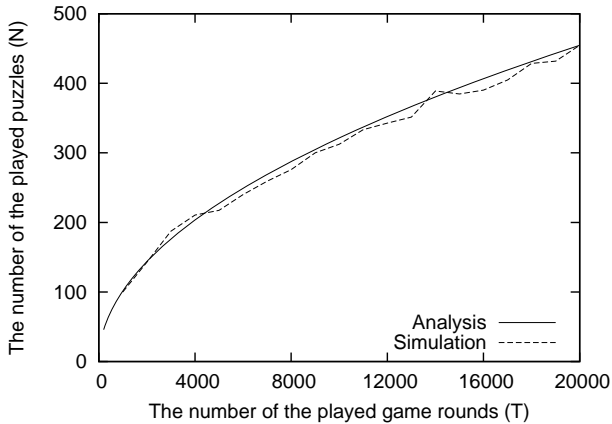


Fig. 4. Comparison of the  $N$  values derived by the simulations and analysis, where  $E[L] = 10.3353$  and  $T$  varies between 200 and 20,000.

increases. There are two reasons for this phenomenon: a) as the total number of game rounds increases, each puzzle tends to take more labels from the system; and b) more puzzles are played. Since  $N = T/r$ , the results confirm that the proposed OPSA approach can effectively balance the two goals, i.e., maximize the number of games played, while identifying as many labels per puzzle as possible.

### C. Comparison of RPSA, FPSA, and OPSA

Here, we present the evaluation of the three puzzle selection algorithms in the ESP game. In the simulation, we set  $T = 10,000$ , and  $E[L] = 10.3353$ ;  $M$  denotes the total number of puzzles in the system. The simulation results are shown in Figure 5.

The results show that, when  $M$  is small (say, smaller than a threshold  $M'$ ), the three algorithms are comparable in terms of the *system utility* achieved. However, when  $M$  is larger than  $M'$ , the *system utility* of OPSA remains consistent regardless of the changes in the values of  $M$ . In contrast, the *system utility* of FPSA and RPSA degrades as the value of  $M$  increases, and RPSA slightly outperforms FPSA when  $M$  is very large. More precisely, the threshold  $M'$  represents the minimal number of puzzles required to achieve the maximum *system utility* (i.e.,

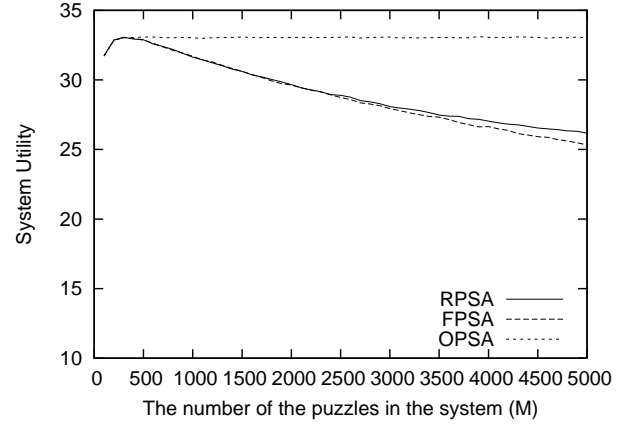


Fig. 5. Comparison of the system utility achieved by the OPSA, FPSA and RPSA schemes with various numbers of puzzles, where  $T$  is fixed at 10,000, and  $E[L]$  is set to 10.3353.

$M' = N = T/r$ ). Since  $T = 10,000$ , and  $E[L] = 10.3353$ , we know that  $r = 31$ . Therefore,  $M' = 10000/31 \approx 321$  in this case. The results indicate that, under the OPSA scheme, the ESP game must maintain at least a certain number of puzzles to achieve the maximum *system utility*; otherwise, it will favor the RPSA and FPSA schemes because their performance is comparable to that of OPSA and they are easy to implement. Fortunately, this is not a problem in general, since the number of the puzzles can be easily increased by adding new puzzles from the Internet.

### D. Discussion

We have presented an analytical model that evaluates the system utility of the ESP game. In addition, we have proposed a puzzle selection algorithm (i.e., OPSA) that can strategically select the puzzles that should be played to achieve the largest possible system utility. However, there are two issues that have yet to be addressed.

- First, the proposed analytical model only considers the number of the games played and the quantity/quality of the game outcomes when measuring the system utility. However, the speed of generating metadata varies a great deal among different puzzles, and even different rounds of the same puzzle. Thus, the proposed model may not be sufficiently representative to measure the “*productivity*” of the ESP game (i.e., the average quantity and quality of the outcomes in each time unit).
- Second, in this study, we assume that each round in a game produces an outcome with a positive score. However, this may not always be the case, since it is possible to have a round *without* any positive outcomes (e.g., both players agree to pass a puzzle, or they fail to agree about the labels within the time limit). Thus, it is necessary to further investigate the proposed model in such scenarios.

To tackle the above issues, a systematic and thorough evaluation of the three playing strategies in real-world GWAP systems is desirable. In an attempt to address this research

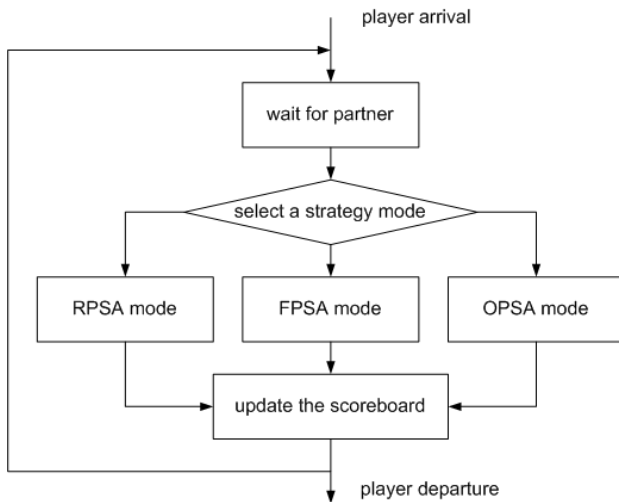


Fig. 6. The flowchart of the *ESP Lite* system.

gap, we implement a system called *ESP Lite*<sup>6</sup>, which mimics the ESP game while embedding the three playing strategies transparently, and recording the complete game trace for evaluation and further research. We present the design of the *ESP Lite* system in the next section.

## VI. IMPLEMENTATION: THE ESP LITE SYSTEM

### A. System Architecture

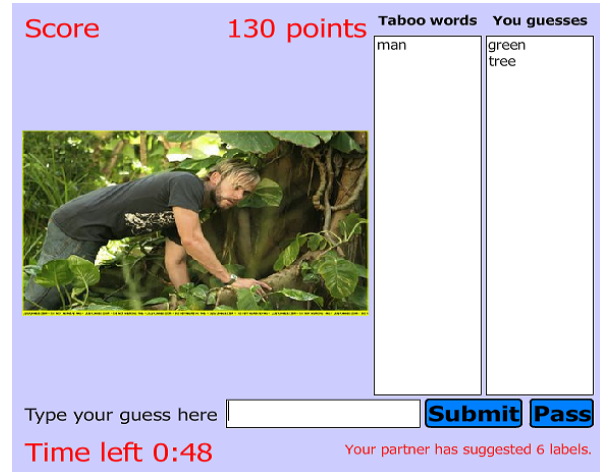
The *ESP Lite* system is a quasi ESP game that mimics the ESP game, but it differs in two respects: 1) once an agreement is reached, the agreed word becomes a taboo word (i.e., the threshold for defining a taboo word is set to one in *ESP Lite*); and 2) it embeds the three playing strategies proposed transparently and records the complete game trace for evaluation and further research. Figure 6 shows the flowchart of the *ESP Lite* system.

*ESP Lite* follows the client-server architecture. Specifically, the *ESP Lite* client is a Flash plug-in that can be played on popular web browsers; and it connects to the game server, which is implemented in Java language and hosted by Academia Sinica, Taiwan<sup>7</sup>. As shown in Figure 6, after logging into the system, the user is automatically matched with a random partner. If there are no players to match with the user, the system will enter the *single-player* mode, and the user will be paired with a game *bot* that mimics human playing behavior by generating guesses based on the puzzle’s historical data. Then, *ESP Lite* selects one of the three playing strategies and creates a new game. The strategy selection process gives priority to the strategy *that has been used least in terms of the number of puzzles played previously*.

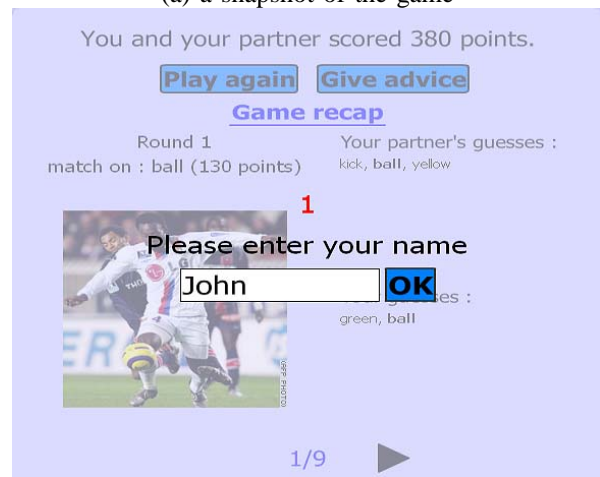
Figure 7 shows the user interface of the *ESP Lite* client. During the game, the client displays the puzzle image, the

<sup>6</sup>Note that, although the game dataset of the ESP game has been released [3], it only contains the agreement words for each puzzle. It does not provide the detailed play history (e.g., the number of play rounds in each puzzle, the pass rate of each puzzle, and play time of each play round). As result, the dataset was insufficient for this study, so we decided to implement a ‘quasi’ ESP system to collect the required data.

<sup>7</sup>*ESP Lite*, <http://nrl.iis.sinica.edu.tw/GWAP/ESPLite/>.



(a) a snapshot of the game



(b) review of the game play

Fig. 7. The user interface of the *ESP Lite* client.

taboo words of the puzzle, the player’s input words, the remaining time, the score, and the number of words that have been input to the puzzle by the other player, as shown in Figure 7-a.

When the game finishes, the *ESP Lite* system calculates the game score, and updates the scoreboard if necessary. The client also displays the history of the played game, as shown in Figure 7-b, so that the players can review each other’s inputs for each played puzzle, and hopefully improve their playing skills. We note that a recent study reported that using the scoreboard and the review mechanism can strengthen a player’s motivation to stay in the game, because they introduce challenges, competition, variety, and communication to the system [37].

### B. Puzzle Dataset

The *ESP Lite* system imports the ESP game dataset as the puzzle dataset. The dataset contains 100,000 images, each of which has approximately 15 labels collected from the ESP game. The CDF of the number of labels per image in the dataset is shown in Figure 8.

We use the ESP game dataset for two reasons. First, using the same set of images allows us to compare the quality of the

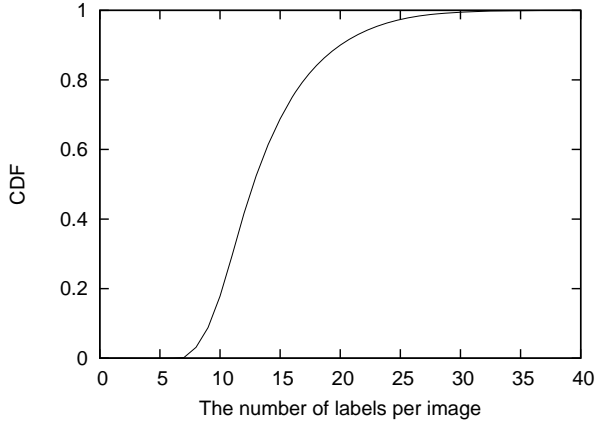


Fig. 8. The CDF of the number of labels per image in the ESP game dataset.

outcomes of the ESP game and the *ESP Lite* system. Second, with the high-quality labels provided by the ESP game dataset, the game bot of the *ESP Lite* system is better able to mimic users' behavior, and thus make single-player games possible.

We should also mention that, to make single-player games enjoyable, it is very important that game bots can make appropriate guesses during the game. If the pre-installed labels of the images are provided actively by their owners (e.g., Flickr [4]), the number of labels for each image is usually small, and the provided labels usually contain specialized information, such as the owners' names and geographical locations; hence, it is difficult to reach an agreement in a single-player game. On the other hand, if the labels of the images are provided collaboratively by general users (e.g., Fotki [5]), it is very likely that the labels will not be meaningful, or they will be commercial words; for example, we found that most images on Fotki were labeled with the two words 'contest' and 'fotki'. Once again, it is difficult to reach agreements in single-player games.

### C. Agreement Scoring

Next, we describe *ESP Lite's* scoring system. As mentioned earlier, when an agreement is reached in the game, a bonus score is awarded to both players based on the quality of the agreed word. For simplicity, the quality of a word is measured by its popularity; that is, words that are more popular generally receive lower scores.

In *ESP Lite*, we import the word frequency list from *Edict Virtual Language Centre* [2] to measure the popularity of words. The list contains the 5,000 most popular words, as well as their frequency, in the *Brown Corpus* [1], a general corpus used in the field of corpus linguistics. Then, we apply the *Porter Stemming Algorithm* [10] to remove common morphological and inflectional endings of English words. The objective is twofold: 1) to prevent words with the same root, such as *determinant* and *determine*, receiving different scores; and 2) to reduce the plural form to the singular form, such as *experiments* to *experiment*.

Suppose there are  $n$  distinct words after applying the *Porter Stemming Algorithm*. We sort the  $n$  words in monotonically

descending order of their frequency. Let  $W$  denote the list of sorted  $n$  words,  $w_i$  denote the  $i$ -th word in  $W$ , and  $f_i$  denote the frequency of  $w_i$ . We know that  $f_1 \geq f_2 \geq f_3 \geq \dots \geq f_n$ .

Next, we divide the  $n$  words into  $k$  levels based on their frequency, such that each level of words has a comparable word frequency sum. Then, we can obtain the level number,  $l_i$ , of  $w_i$  by

$$l_i = \left\lceil \frac{\sum_{j=1}^i f_j}{\sum_{j=1}^n f_j} \cdot k \right\rceil. \quad (4)$$

We define  $S(w_i)$  as the score function of  $w_i$  as follows:

$$S(w_i) = S_{base} + (l_i - 1) \cdot S_{offset}, \quad (5)$$

where  $S_{base}$  is the minimum value of  $S(w_i)$  in the system, and  $S_{offset}$  is the score offset between each adjacent level of words. Note that if the agreed word,  $w_{agreed}$ , is not on the word list (i.e.,  $w_{agreed} \notin W$ ), *ESP Lite* will give it the highest score, i.e.,  $S(w_{agreed}) = S_{base} + k \cdot S_{offset}$ . Moreover, in *ESP Lite*, we set  $k = 9$ ,  $S_{base} = 60$ , and  $S_{offset} = 10$ .

## VII. EXPERIMENT

The *ESP Lite* system was released on March 9, 2009. Within one month (by April 9), it created 3,130 games, played 9,376 distinct puzzles, and reached 12,312 agreements. The OPSA scheme was used in 1,444 of the games, which played 575 distinct puzzles and reached 3,418 agreements; the RPSA scheme was used in 812 games, which played 4,380 distinct puzzles and reached 4,473 agreements; and the FPSA scheme was used in 874 games, which played 4,421 distinct puzzles and reached 4,421 agreements. The players were from 606 distinct IP addresses (588 in the Taiwan, 12 in USA, 2 in Hong Kong, and 4 in various other countries). Most of the players in Taiwan were students. We present the analysis and the discussion of the experiment results in the following subsections.

### A. General description

Figure 9 shows the histogram of players' arrival times during the experiment. The results confirm our intuition that most Internet activities take place from the afternoon until midnight. We observe that there are two peak hours in the figure: 5pm-6pm, and 12am-1am. The results indicate that players like to play the system either for relaxation (i.e., to be entertained after finishing work) or for self-satisfaction (i.e., to see their names on the scoreboard, since the system resets the scoreboard regularly at midnight).

We also observe that, after the warm-up period, the number of games created by OPSA is consistently higher than in the FPSA and RPSA games, as shown in Figure 10. The reason is quite simple: under the OPSA scheme, the assigned puzzle usually has a number of taboo words, so the players are forced to invest more time in order to reach an agreement (or they simply pass the puzzle, as shown in Figure 11). Hence, the number of puzzles solved in each OPSA game is smaller than in the FPSA and RPSA games, as shown in Figure 12. Since the *ESP Lite* system gives priority to the playing strategy that

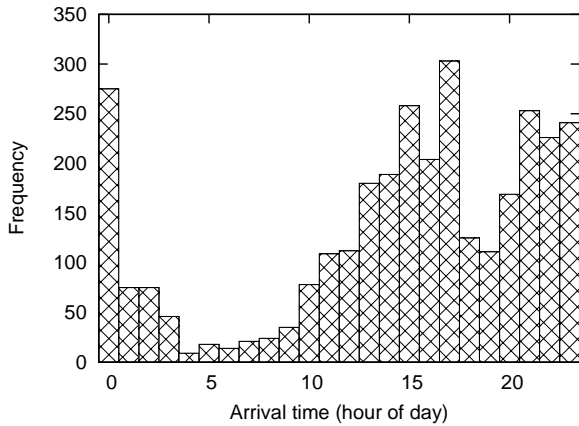


Fig. 9. The histogram of players' arrival times during the experiment.

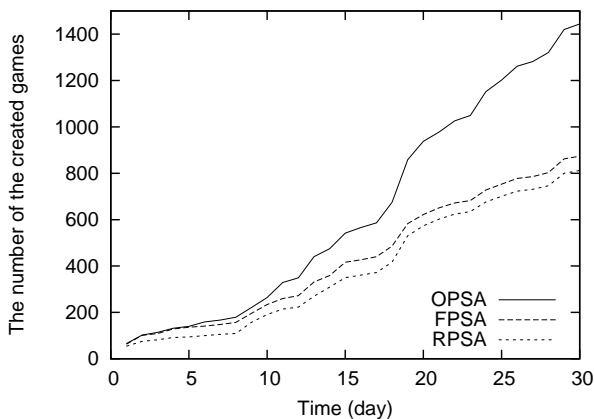


Fig. 10. The number of games played for each playing strategy in the experiment.

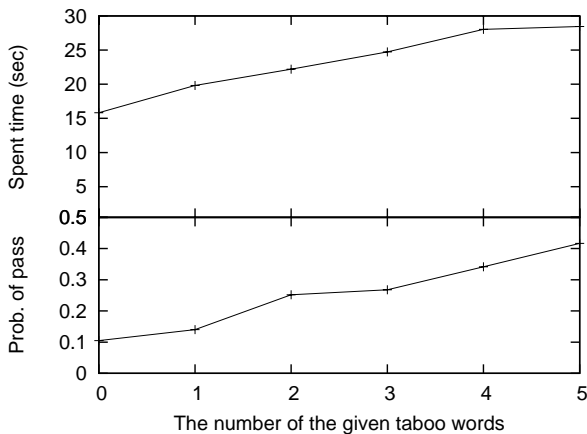


Fig. 11. Comparison of the average time required to reach an agreement per puzzle and the average pass rates for different numbers of taboo words in each puzzle.

has the lowest number of played puzzles, it tends to create more OPSA games in order to balance the number of puzzles played for each playing strategy.

The results in Figures 10 and 12 also show that the performances of the FPSA and RPSA schemes are comparable in the experiment. The reason is that, when the number of the puzzles

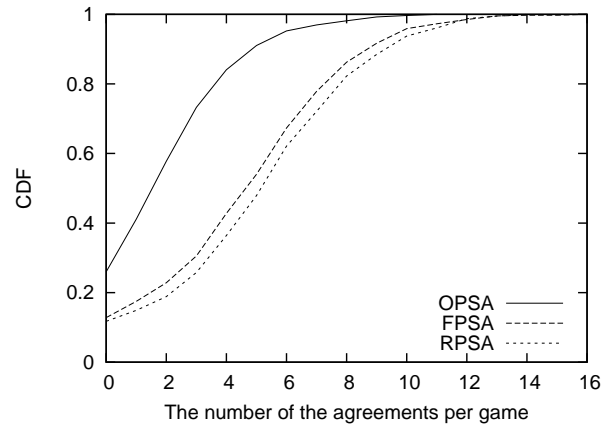


Fig. 12. The CDF of the number of agreements reached for each playing strategy in the experiment.

TABLE I  
THE 10 MOST FREQUENT LABELS IN THE ESP GAME DATASET AND THE  
*ESP Lite* EXPERIMENT RESULTS.

ESP			ESP Lite		
Word	Frequency	Score	Word	Frequency	Score
white	40,349	110	man	1,138	100
black	31,529	120	woman	889	120
blue	24,011	120	girl	810	110
man	21,987	100	black	564	120
red	20,224	120	red	540	120
woman	17,535	120	sky	497	130
hair	16,344	130	light	316	110
green	15,229	130	tree	192	130
girl	14,772	110	trees	164	130
sky	13,814	130	water	159	110

in the input pool is extremely large, the RPSA scheme tends to select a 'fresh' (i.e., unplayed) puzzle every time, which is exactly the same as the FPSA scheme. Therefore, since the number of available puzzles in our dataset is much larger than the number of the game rounds played in the experiment, the behavior of the two schemes is similar.

Figure 13 shows that when the OPSA strategy is used, the optimal value of  $r$  (i.e., the number of rounds that each puzzle is played in the system) increases with the total number of rounds played, which confirms our previous findings (cf. Figure 3). Moreover, when we apply the same scoring function to the two datasets compiled from the ESP game and the *ESP Lite* experiment results, we find that the score distribution of the agreed words collected in the experiment is consistently and slightly lower than that of agreed words in the ESP dataset, as shown in Figure 14. The reason may be that most players in our experiment are not fluent in English as it is not their mother tongue. By contrast, most players of the ESP game are native English speakers. We also observe that most of the game outcome scores are higher than 90 points (i.e., higher than the 4-th level), which again confirms previous reports that the agreed words in the ESP game are typically of good quality. Table I shows the 10 most frequent agreed words in the ESP game data set and in the *ESP Lite* experiment results.



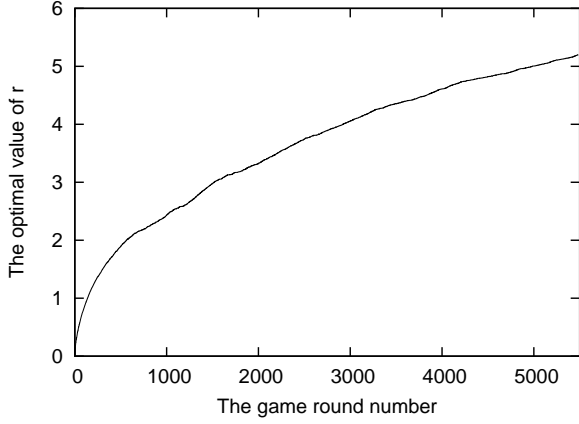


Fig. 13. The optimal value of  $r$  over time when the OPSA strategy is used in the system.

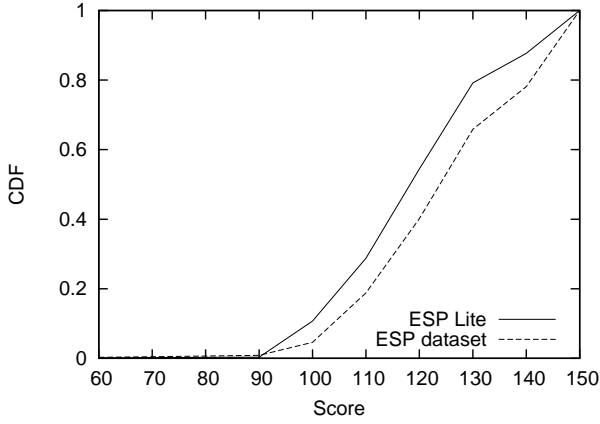


Fig. 14. The score distributions of the agreed words collected in the experiment and those in the ESP dataset.

### B. Comparison of the pass rates

Next, we compare the ‘pass rates’ of the three playing strategies compared in the experiment, i.e., the likelihood that both players will decide to pass the assigned puzzle. The results in Figure 15 show that, after the warm-up period, the OPSA scheme has a much higher pass rate (more than double) than the other two schemes, i.e., above 0.2 in the experiment. This is because, under the OPSA scheme, there are at most  $r - 1$  taboo words for each puzzle. As a result, the puzzle is more difficult to solve, and the players are more likely to pass it (as shown in Figure 11).

We also compare the time required for players to either reach an agreement or decide to pass a puzzle. As shown in Figure 16, players tend to pass a puzzle either immediately (24% of the passed rounds were finished within 10 seconds) or when they finally realize the puzzle is too difficult after considering it for more than 30 seconds (players spent more than 30 seconds on 30% of the passed rounds). On the other hand, for game rounds that resulted in agreements, only 5% of the rounds were finished in 5 seconds, and 12% required more than 30 seconds. Interestingly, if players can reach an agreement very quickly, most of the agreed words are frequent words, such as ‘man’ and ‘woman’.

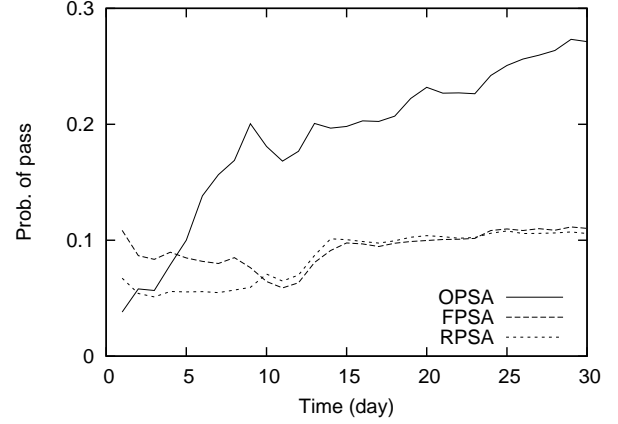


Fig. 15. Comparison of the average pass rates of the three playing strategies, given different numbers of taboo words in the puzzle.

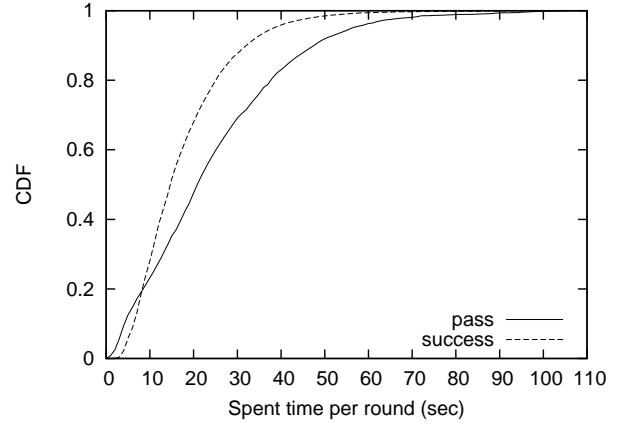


Fig. 16. The CDF of the time spent per game round, where the players either reached an agreement or decided to pass the puzzle.

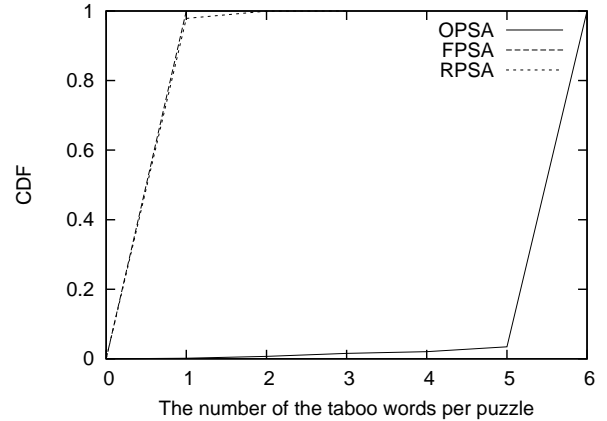


Fig. 17. The distribution of agreements reached under different playing strategies.

### C. Comparison of the system performance

Finally, we compare the system performance of the three playing strategies. We consider two metrics: the number of agreements reached per puzzle and the system utility (i.e., Equation 1).

Figure 17 compares the number of agreements reached per

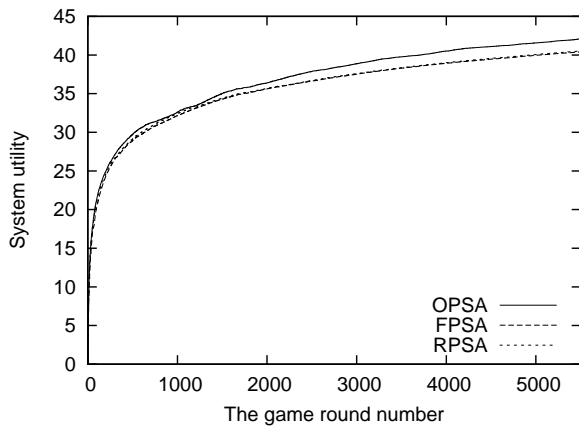


Fig. 18. Comparison of the system utility achieved by the three playing strategies in the experiment.

puzzle under the three playing strategies. All puzzles only reach one agreement when FPSA is used, and only 2% of the puzzles reach more than one agreement when RPSA is used. As mentioned earlier, this is because the behavior of the RPSA scheme is equivalent to that of the FPSA scheme when the number of the available puzzles in the system is very large. In contrast, when OPSA is used, more than 97% of the puzzles reach at least 5 agreements. This is because the optimal value of  $r$  is larger than 4 after 2,921 game rounds, as shown in Figure 13. The results indicate that the OPSA scheme yielded more agreements for each puzzle in the experiment.

In addition, the results in Figure 18 show that the OPSA scheme consistently outperforms the other schemes in terms of system utility<sup>8</sup>, and the difference between the OPSA scheme and the other schemes increases with the number of game rounds. The results confirm our previous analysis results, which demonstrate that *GWAP systems can better accommodate both system performance objectives (namely the number of the distinct puzzles played, and the average aggregated score of the agreements reached in each puzzle) if they are designed and played with strategies.*

## VIII. CONCLUSION

In this paper, we have studied the ESP game, an emerging GWAP system, and proposed a metric called *system utility* to evaluate the game's performance. Moreover, we argue that GWAP systems need to be designed and played with strategies so that human intelligence can be collected in a more efficient manner. Based on our analysis, we propose a new puzzle selection algorithm (i.e., OPSA) that provides guidelines for improving the ESP game. Using analysis and a comprehensive set of simulations, we have investigated the intrinsic properties of the ESP game, and demonstrated that the proposed OPSA scheme significantly outperforms other schemes in all test cases. To investigate the proposed scheme

<sup>8</sup>Note that it is possible to adapt a strategy that employs a fixed  $r$  when selecting puzzles. The scheme's performance would be somewhere in-between that of the OPSA and the FPSA/RPSA schemes. However, the 'fixed- $r$ ' scheme cannot always achieve the highest system utility because the optimal value of  $r$  increases over time (as shown in Figure 13).

in real-world systems, we have implemented a quasi ESP game, called *ESP Lite*, which mimics the ESP game while embedding the three playing strategies transparently. During a one-month experiment, we verified that the OPSA scheme can achieve higher system utility than the other two schemes. The contribution of this work is three-fold. First, to the best of our knowledge, this is the first GWAP study that proposes, implements and evaluates an analytical model on real-world GWAP systems. Second, our evaluation results demonstrate that GWAP systems are more efficient if they are designed and played with strategies. Finally, the proposed analysis is simple and applicable to other ESP-like games, and the proposed puzzle selection strategy shows promise for use in the design and implementation of future GWAP systems.

## ACKNOWLEDGEMENTS

We are grateful to the editors and anonymous reviewers for their insightful comments. This research was supported in part by the National Science Council of Taiwan under Grants: NSC 98-2221-E-001-014-MY3 and NSC 98-2631-H-001-013.

## REFERENCES

- [1] Brown Corpus Manual. <http://khnt.aksis.uib.no/icame/manuals/brown/>.
- [2] Edict Virtual Language Centre. <http://www.edict.com.hk/textanalyser/wordlists.htm>.
- [3] ESP Dataset. <http://www.hcomp2009.org/Data.html>.
- [4] Flickr - Photo Sharing. <http://www.flickr.com/>.
- [5] Fotki: Share and Print Your Photos. <http://www.fotki.com/>.
- [6] Google Image Labeler. <http://images.google.com/imagelabeler/>.
- [7] GWAP. <http://www.gwap.com/gwap/>.
- [8] Herd It. <http://www.herdit.org/>.
- [9] Major Miner. <http://majorminer.org/>.
- [10] Porter Stemming Algorithm. <http://tartarus.org/martin/PorterStemmer/>.
- [11] L. Barrington, D. Turnbull, D. O'Malley, and G. Lanckriet. Herd It: Designing A Social Game to Tag Music. In *Human Computation Workshop*, 2009.
- [12] M. Bell, S. Reeves, B. Brown, S. Sherwood, D. MacMillan, J. Ferguson, and M. Chalmers. Eyespy: Supporting navigation through play. In *ACM SIGCHI*, 2009.
- [13] P. N. Bennett, D. Maxwell, and A. Mityagin. Learning Consensus Opinion: Mining Data from a Labeling Game. In *WWW*, 2009.
- [14] J. P. Bigham, R. S. Kaminsky, R. E. Ladner, O. M. Danielsson, and G. L. Hempton. WebInSight: making web images accessible. In *ACM SIGACCESS*, 2006.
- [15] S. H. Bokhari. *Assignment Problems in Parallel and Distributed Computing*. Springer, 1987.
- [16] 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, February 1988.
- [17] 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.
- [18] L.-J. Chen, B.-C. Wang, K.-T. Chen, I. King, and J. H.-M. Lee. An analytical study of puzzle selection strategies for the esp game. In *IEEE/WIC/ACM Web Intelligence Conference*, 2008.
- [19] S. Dehaene, V. Izard, E. Spelke, and P. Pica. Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene Cultures. *Science*, 320(5880):1217–1220, May 2008.
- [20] C. Gentry, Z. Ramzan, and S. Stubblebine. Secure distributed human computation. In *ACM Electronic Commerce Conference*, 2005.
- [21] 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.
- [22] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-J. Hsu, and K.-T. Chen. KissKissBan: A Competitive Human Computation Game for Image Annotation. In *Human Computation Workshop*, 2009.
- [23] J. Howe. The Rise of Crowdsourcing. *WIRED Magazine*, 14(6), June 2006.
- [24] S. Jain and D. C. Parkes. A Game-Theoretic Analysis of Games with a Purpose. In *Workshop on Internet and Network Economics*, 2008.
- [25] A. Kosorukoff. Human based genetic algorithm. In *IEEE SMC*, 2001.
- [26] E. Law and L. von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *ACM SIGCHI*, 2009.
- [27] Y. Li, C.-J. Hu, and X. Yao. Innovative batik design with an interactive evolutionary art system. *Journal of Computer Science and Technology*, 24(6):1035–1047, 2009.
- [28] H. Lieberman, D. Smith, and A. Teeters. Common Consensus: a web-based game for collecting common-sense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.
- [29] V. M. Lo. Heuristic algorithms for task assignment in distributed systems. *IEEE Transactions on Computers*, 37(11):1384–1397, November 1988.
- [30] 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.
- [31] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008.
- [32] P. Shenoy and D. S. Tan. Human-aided computing: utilizing implicit human processing to classify images. In *ACM SIGCHI*, 2008.
- [33] K. Siorpaes and M. Hepp. Games with a Purpose for the Semantic Web. *IEEE Intelligent Systems*, 23(3):50–60, May/June 2008.
- [34] J. Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wis-*
- dom Shapes Business, Economies, Societies and Nations*. Doubleday, May 2004.
- [35] L. von Ahn. Games with a Purpose. *IEEE Computer*, 39(6):92–94, June 2006.
- [36] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *ACM SIGCHI*, 2004.
- [37] L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.
- [38] L. von Ahn, S. Ginosar, M. Kedia, and M. Blum. Improving Image Search with PHETCH. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.
- [39] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. Improving accessibility of the web with a computer game. In *ACM SIGCHI*, 2006.
- [40] L. von Ahn, M. Kedia, and M. Blum. Verbosity: A Game for Collecting Common-Sense Facts. In *ACM SIGCHI*, 2006.
- [41] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A Game for Locating Objects in Images. In *ACM SIGCHI*, 2006.
- [42] I. Weber, S. Robertson, and M. Vojnovic. Rethinking the ESP Game. Technical report, Microsoft Research MSR-TR-2008-132, 2008.
- [43] D. H. Wilson, A. C. Long, and C. Atkeson. A context-aware recognition survey for data collection using ubiquitous sensors in the home. In *ACM SIGCHI*, 2005.



**Ling-Jyh Chen** received the B.Ed. degree in information and computer education from National Taiwan Normal University in 1998, and the M.S. and Ph.D. degrees in computer science from University of California at Los Angeles in 2002 and 2005 respectively. He joined the Institute of Information Science as assistant research fellow in 2005. His research interests are opportunistic networks, wireless and mobile networks, network protocols, network measurements, and social computing. He is a member of IEEE and ACM.



**Bo-Chun Wang** received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University in 2004 and 2006. He entered University of Southern California to study computer science for Ph.D. degree in 2009. His research interests are network measurement, social computing, and web 2.0.



**Wen-Yuan Zhu** received the B.S. degree in Computer and Information Science from Aletheia University in 2007, and the M.S. degree from the Department of Computer Science and Information Engineering, National Taiwan Normal University in 2009, respectively. His research interests are web 2.0 and social computing.