An Analytical Model for Generalized ESP Games

Bo-Chun Wang, Chien-Wei Lin, Kuan-Ta Chen, Ling-Jyh Chen

Institute of Information Science, Academia Sinica, Taiwan

Abstract

The ESP game belongs to the genre called *Games with a Purpose* (GWAP), which leverage people's desire to be entertained and also outsource certain steps of the computational process to humans. The games have shown promise in solving a variety of problems, which computer computation has been unable to resolve completely thus far. In this study, we consider generalized ESP games with two objectives. First, we propose an analytical model for computing the utility of generalized ESP games, where the number of players, the consensus threshold, and the stopping condition are variable. We show that our model can accurately predict the stopping condition that will yield the optimal utility of a generalized ESP game under a specific game setting. A service provider can therefore utilize the model to ensure that the hosted generalized ESP games produce high-quality labels efficiently. Second, we propose a metric, called *system gain*, for evaluating the performance of ESP-like GWAP systems, and also use analysis to study the properties of generalized ESP games. We believe that GWAP systems should be designed and played with strategies. To this end, we implement an Optimal Puzzle Selection Strategy (OPSA) based on our analysis. Using a comprehensive set of simulations, we demonstrate that the proposed OPSA approach can effectively improve the *system gain* of generalized ESP games, as long as the number of puzzles in the system is sufficiently large.

Keywords: Games with a Purpose, Human Computation, Tagging

1. Introduction

Games With A Purpose (GWAP) represents a new paradigm that exploits people's desire to be entertained by outsourcing certain steps of the computational process to the players [1, 2]. By taking advantage of people's desire to be entertained and exploiting “human cycles” in computation, GWAP not only attract people to play voluntarily, but also produce useful metadata as a by-product. The paradigm has shown promise in solving a variety of problems, such as image annotation [3, 4, 5], audio annotation [6, 7, 8, 9], and commonsense reasoning [10, 11], which computer programs have been unable to resolve completely thus far.

Several GWAP systems have been proposed in recent years [12, 13, 14]. Among them, the ESP Game [3] was the first to successfully realize the advantages of GWAP systems, and it was subsequently adopted as the *Google Image Labeler* [15]. 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 [16]).

In this paper, we have two objectives, both dealt with the ESP game. One objective is to model the performance of the ESP game and optimize its utility by redefining the criteria for finishing a game. The ESP game proposed in [3] only allows two players to participate. To play, in each round, the randomly matched players keep suggesting appropriate labels to describe an image until an agreement is reached (i.e., the same word is entered by both players). Once the players achieve a consensus, the current image is considered solved and the game continues with the next image. In our study, we consider a more generalized ESP game that incorporates the following extensions:

1. The number of players, n, can be greater than 2.
2. The consensus threshold, m, can be any positive integer that is not larger than n; that is, a label is considered a consensus decision if it is proposed by m out of n players. The purpose of this threshold is to control the quality of the reached consensus, as labels may reach consensus coincidentally even if they are irrelevant to the image.
3. The stopping condition, k, can be any positive integer; that is, an image is considered correctly labeled if k consensuses have been reached. The intention of this extension is to improve the efficiency of consensus reaching, as players’ suggestions (i.e., labels) can be accumulated over time.

In our framework for generalized ESP games, the game proposed by Ahn and Dabbish [3] corresponds to an instance where $n = 2$, $m = 2$, and $k = 1$. Hereafter, we use "ESP games" or "games" to refer to the generalized version we propose. As some variants of ESP games ask players to label audio clips instead of images, we use the term "puzzle" to denote the target object that players label to achieve consensus.

In our model, we assume that the number of appropriate labels for each puzzle is limited, and all remaining words are considered inappropriate. For example, to label an image containing a red car beside a river, "car," "river," "red" are considered appropriate or good. Other words are considered inappropriate or bad, even if there is a consensus among the players. For example, players may input typos like "cra," "rive," or "rde" by mistake, or words that are too vague or general, such as "picture," "photo," "sea" and still achieve a consensus occasionally. In such cases, we deem that the current game yields a bad label and the quality of the game's output is reduced.

We model the *utility* of ESP games, i.e., the throughput rate of good labels for the puzzles and its relationship with the game's parameters, i.e., the number of players, the consensus threshold, and the stopping condition. We find that a tradeoff exists between the efficiency of the consensus achieved and the quality of matched labels. Given a fixed number of players and a consensus threshold, our model can predict the optimal stopping condition that will ensure the maximal possible utility for an ESP game. Our contribution in this part is three-fold:

1. We present a generalized ESP game in which the number of players, the consensus threshold, and the stopping condition are variable.
2. We propose a probabilistic model that is able to predict the efficiency, quality, and utility of an ESP game.
3. Via extensive simulations, we show that the proposed model can accurately predict the optimal stopping condition, which facilitates the maximal utility of a generalized ESP game. This feature can be used by game service providers to maximize the outcome of games, given that the number of players willing to invest time and effort in the game is limited.

For the second objective of this study, we propose a metric, called *system gain*, to evaluate the performance of ESP-like GWAP systems. The proposed metric considers two factors: the number of puzzles that have been played in the system, and the average outcomes produced by each puzzle. Both factors are critical for GWAP systems, but unfortunately they do not complement each other. We believe that GWAP systems should be *played with a strategy*. Specifically, based on our analysis, we propose an *Optimal Puzzle Selection Algorithm* (OPSA) that can maximize the *system gain* by properly accommodating the two opposing factors. Using a set of simulations, we investigate the properties of ESP games, and evaluate the proposed OPSA scheme on two widely used schemes, namely the *Random Puzzle Selection Algorithm* (RPSA) and the *Fresh-first Puzzle Selection Algorithm* (FPSA). The results demonstrate that, with the OPSA scheme, ESP games yield a much better *system gain* than the two compared schemes. In addition, the presented analysis is simple and applicable to other ESP-like GWAP systems.

The remainder of this paper is organized as follows. Section 2 contains a review of related works on human computation systems. In Section 3, we describe the rules of generalized ESP games. In Section 4, we present our analysis and compare three puzzle selection algorithms for ESP games, namely the RPSA, FPSA, and OPSA schemes. Section 5 presents a comprehensive set of simulation results, which we analyze and explain in detail. In Section 6, we consider several issues arising from this work. We then summarize our conclusions in Section 7.

2. Background

"Human Computation" was pioneered by Luis von Ahn and his colleagues, who created *Games With A Purpose* (GWAP) [1, 2] that people play voluntarily and produce useful metadata as a by-product. By taking advantage of people's desire to be entertained, Human Computation 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 [12, 13, 14].

Among them, the online *ESP Game* [3] was the first GWAP system (which was subsequently adopted as the *Google Image Labeler* [15]); and it has been shown that the collected labels facilitate 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 [16]). In addition, the *Peekaboom* system [4] can help determine the location of objects in images; and the *Squigl* system [17] and the *LabelMe* system [5] can provide the complete outlines of the objects in an image. *Phetch* [18, 19] can provide image descriptions that improve web accessibility and image searches, while the *Matchin* system [17] can help image search engines rank images based on which ones look the best.

The concept of GWAP has been applied to other problems in a variety of forms[1]. For instance, Xiao and Houser propose the use of a *Ultimatum Game* to evaluate the 'emotion' behind the messages [20], and they design a *Coordination Game* to classify natural language messages [21]. Unlike the ESP game that collects independent guesses from randomly paired participants, [20, 21] use a set of candidate labels known by all participants and promote coordination among participants to complete game tasks. In addition, the *Herd It* [6, 8], *Major Miner* [9], and *TagATune* [7] systems, which provide annotation for sounds and music, can improve audio searches. The *Verbosity* system [11] and the *Common Consensus* system [10] collect "common-sense" knowledge that is valuable for commonsense reasoning and enhancing the design of interactive user interfaces. The *GUIDE* system [22] allows users to explore UML designs interactively by playing games, and resolves the incompleteness and informal definition of UML models. Moreover, Metoyer et al. [23] use *Reat-Time Strategy* (RTS) games to collect better annotation of users' real-time decisions within specific spatial and temporal constraints in dynamic environments. Ferreira and Atkinson [24] propose the use of a feedback component in the loop for intelligent tutoring systems of foreign languages, and [25, 26, 27, 28] employ mobile social gaming for geospatial tagging. Finally, Shenoy and Tan [29] 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 [2, 30, 31, 32, 33]. For example, Ho et al. [33] proposed solving the *coalition* problem by integrating both collaborative and competitive elements in image labeling games. Gen-

[1] In [2], Von Ahn and Dabbish have proposed to classify GWAP systems into three types based on their game structures, namely, 1) *output-agreement games*: players are given the same input and must agree on an appropriate output (e.g., *ESP* [3] and *Squigl* [17]); 2) *input-agreement games*: players must determine whether they have been given the same input (e.g., *TagATune* [7]); and 3) *inverse-problem games*: one player produces an output based on a given input, and the other player guesses the input (e.g., *Peekaboom* [4] and *Phetch* [18, 19]).

try et al. [30] 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. In [31], 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 predicable labels and thereby improve the quality of the collected labels. Houser and Xiao [21] consider the ESP game as a coordinate game with multiple equilibria, and they argue that a *coordination equilibrium* will occur under certain condition (i.e., the agreement achieved in an ESP game is the equilibrium). Jain and Parkes [32] applied game theoretic analysis to the ESP game. They investigated the equilibrium behavior under different incentive mechanisms and provided guidelines for the design of such mechanisms. Von Ahn [2] proposed a set of evaluation metrics, namely, throughput, lifetime play, and expected contribution, to determine whether ESP-like GWAP systems are successful.

3. Generalized ESP Games

After entering 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.[2] Initially, a randomly selected image is displayed to both players. Both players then independently input words to label the image until an agreement is reached (i.e., the same word is entered by both players), and a bonus score is awarded to each player based on the *quality* of the agreed word. In practice, the *quality* of a word can be partially determined by its popularity; generally, words that are more popular receive lower quality scores because they contain less information about the puzzle. 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

[2] The game is named "ESP" because the players have to work together to solve the tasks without communicating with each other, i.e., using the so-called *Extra-Sensory Perception* (ESP) ability [3].

for using taboo words is to ensure that each image is labeled with a variety of words.

In this study, we consider generalized ESP games that are based on the following assumptions:

1. *Round-based play.* We assume that the game play is round-based rather than continuous. In each round, a player can only make one guess about the current puzzle, and the system checks whether the players' guesses match at the end of each round.
2. *Independent guess.* For model tractability, we assume that guesses made by a player are independent and identically distributed[3]; that is, a player's current guess is not affected by his/her guesses in previous rounds. Although this assumption somewhat simplifies realistic user behavior, it does not affect the model's accuracy significantly. We discuss this issue further in Section 6.1.
3. *Good and bad words.* We assume that the number of "good" labels for each puzzle is limited, so that all remaining words are considered "bad", i.e., inappropriate. The good words are not known by the game system or the participants a priori. We expect that players will do their best to guess good words in the vocabulary. However, there is a possibility that they will fail to pick the right words; instead, they may make a guess from the bad vocabulary due to a spelling error, a memory error, a misunderstanding, or a deliberate ploy.
4. *Uniform guess.* How human beings conceptualize puzzles has not yet to be statistically modeled. Therefore, we assume that players' guesses are drawn uniformly from both the good and bad vocabulary pools.

We assume that n players participate in a game. In addition, the consensus threshold is set to m, and the stopping condition is set to k. For a certain puzzle, the size of the good vocabulary is denoted by v_{good}, while that of the bad vocabulary is denoted by v_{bad}. Thus, the total number of words that players can choose from is $d = v_{good} + v_{bad}$. The probability that a player will guess a word in the good vocabulary is $prob_{good}$; and the probability that a player will guess a bad word is $prob_{bad}$, which is equal to $1 - prob_{good}$. The variables used in the model are summarized in Table 1; All variables are

[3] We note that some GWAP systems use non-independent inputs to improve game efficiency (e.g., [20, 21]), and we do not consider these games in this study.

Table 1: Variable Definitions

Name	**Meaning**
n	number of players
m	number of guesses required to reach a consensus
k	number of labels required to solve a puzzle
v_{good}	size of the good vocabulary
v_{bad}	size of the bad vocabulary
d	total size of the vocabularies
$prob_{good}$	probability of choosing good words in a round
$prob_{bad}$	probability of choosing bad words in a round

positive integers, except $prob_{good}$ and $prob_{bad}$ that both range between 0 and 1.

4. Modeling of Generalized ESP Games

4.1. Probabilistic Modeling of Generalized ESP Games

In this subsection, we describe the proposed probabilistic model for generalized ESP games. First, we estimate the number of rounds required to solve a puzzle, as well as the number of good and bad labels suggested by participants before a puzzle is solved. Then, based on our model, we evaluate the productivity of an ESP game in three aspects, namely, efficiency, quality, and utility.

4.1.1. Time Required to Solve Puzzles

We begin by modeling the number of rounds, S, required to solve a puzzle. The terms “consensus” and “match” are used interchangeably to indicate that a label has been proposed by m players in a game. The probability mass function of S is as follows:

$$\begin{aligned} f_S(s) &= \Pr(\text{a puzzle is solved in the } s_{th} \text{ round}) \\ &= \Pr(\text{no. of matches} \geq k \text{ in the } s_{th}\text{round}) \\ &= \Pr\left(\text{no. of matches} \geq k \text{ in the first } s \text{ rounds}\right) - \\ &\quad \Pr\left(\text{no. of matches} \geq k \text{ in the first } (s-1) \text{ rounds}\right). \end{aligned} \tag{1}$$

We assume that the probability that exactly i matches will occur in the first s rounds is $P(i;s)$, and among the i matches, i_{good} matches are from good words and i_{bad} matches are from bad words. We now derive the probability of i_{good} matches in the first s rounds. On average, each player in the first s rounds proposes $s_{good} = s \cdot prob_{good}$ good words and $s_{bad} = s \cdot prob_{bad}$ bad words. A match in the first s rounds indicates that at least m players suggest the same label in a total of $n \cdot s$ guesses. Moreover, if the matched label is a good word, it indicates that at least m players propose the label with a total of $n \cdot s_{good}$ guesses. We can model the probability of one good match occurring in the first s rounds as

$$\begin{aligned} &\Pr(\text{one good match in the first } s \text{ rounds}) \\ &= P_{good}(1) \\ &= 1 - \sum_{q=0}^{m-1} \left(\begin{array}{c} n \cdot s_{good} \\ q \end{array} \right) \left(\tfrac{1}{v_{good}} \right)^{q} \left(1 - \tfrac{1}{v_{good}} \right)^{n \cdot s_{good} - q} . \end{aligned} \tag{2}$$

Next, we model the probability of i_{good} good matches occurring in the first s rounds. Since the i_{good} good matches indicate that exactly i_{good} words have been matched, we have a total of $C_{i_{good}}^{v_{good}}$ combinations of matched labels. The probabilities of the combinations are equivalent because each good word has an equal probability, $1/v_{good}$, of being selected. Therefore, the probability of i_{good} good matches in the first s rounds can be computed by

$$P_{good}(i_{good}) = C_{i_{good}}^{v_{good}} P_{good}(1)^{i_{good}} [1 - P_{good}(1)]^{v_{good} - i_{good}}. \tag{3}$$

Similarly, the probability of i_{bad} bad matches in the first s rounds can be computed by

$$P_{bad}(i_{bad}) = C_{i_{bad}}^{v_{bad}} P_{bad}(1)^{i_{bad}} [1 - P_{bad}(1)]^{v_{bad} - i_{bad}}. \tag{4}$$

Combining Equations 3 and 4, we can derive the probability of i matches in the first s rounds as

$$P\left(i; s\right) = \sum_{i_{good}=0}^{\min(i, v_{good})} P_{good}\left(i_{good}\right) P_{bad}\left(i_{bad}\right), \tag{5}$$

where i_{good} must be in the range 0 and $min(i, v_{good})$ and the sum of i_{good} and i_{bad} must be i. After rewriting the probability mass function of S, the number of rounds needed to solve a puzzle becomes

$$f_S(s) = \sum_{q=1}^{s} f_S\left(q\right) - \sum_{q=1}^{s-1} f_S\left(q\right) = \left[1 - \sum_{i=0}^{k-1} P\left(i; s\right) \right] - \left[1 - \sum_{i=0}^{k-1} P\left(i; s-1\right) \right]. \tag{6}$$

Finally, we obtain the expected number of rounds needed to solve a puzzle as

$$E\left(s\right) = \sum_{s=1} s \cdot f_S\left(s\right). \tag{7}$$

4.1.2. Number of Matches

Next, we model the composition of the matched labels, i.e., how many good labels and bad labels are matched. We first derive the expected number of good matches. By assuming that the puzzle is solved in the s_{th} round, on average, $n \cdot s_{good}$ guesses will be made by n players, and each of the guesses will be drawn from the v_{good} good words. We treat the question of whether a certain word is a match or not as a Bernoulli event, where "success" indicates that the label is matched and "fail" indicates a non-match. Consequently, the sum of the Bernoulli random variable of each good word will be a binomial random variable with a success probability equal to Equation 2. It can be computed as

$$\sum_{v_i \in V_{good}} I(v_i \text{ matched}), \tag{8}$$

where V_{good} denotes the set of good words, and $I(\cdot)$ denotes the indicator function. Let $N_{good}(s)$ be the expected value of Equation 8, i.e., the expected number of good matches in the first s rounds, it can be derived by

$$\begin{aligned} N_{good}(s) &= v_{good} \cdot P_{good}(1) \\ &= v_{good}\left\{1 - \left[\sum_{q=0}^{m-1} \binom{n \cdot s_{good}}{q} \left(\tfrac{1}{v_{good}}\right)^q \times \left(1 - \tfrac{1}{v_{good}}\right)^{n \cdot s_{good} - q}\right]\right\}. \end{aligned} \tag{9}$$

$N_{bad}(s)$, the expected number of bad matches in the first s rounds, can be derived similarly by

$$\begin{aligned} N_{bad}(s) &= v_{bad} \cdot P_{bad}(1) \\ &= v_{bad}\left\{1 - \left[\sum_{q=0}^{m-1} \binom{n \cdot s_{good}}{q} \left(\tfrac{1}{v_{bad}}\right)^q \times \left(1 - \tfrac{1}{v_{bad}}\right)^{n \cdot s_{bad} - q}\right]\right\}. \end{aligned} \tag{10}$$

Note that both $N_{good}(\cdot)$ and $N_{bad}(\cdot)$ are functions of S, the number of rounds required to solve a puzzle. In other words, for puzzles that require a different number of rounds to find a solution, the expected number of good matches and bad matches will also be different.

4.1.3. Probabilistic Analysis of Match Quality

Here we introduce a bounding on the match quality, i.e., the upper bound of probability for having a bad match when a match happens.

Theorem 1. *Let $p_{good} = k_1 \times p_{bad}$ and $v_{bad} = k_2 \times v_{good}$, the probability of matching on bad words under the condition matches happen in the s round satisfies*

$$P_s(\textit{bad match}|\textit{match}) \leq \frac{1}{1 + k_1^m k_2^{m-1}(1 - (n \cdot s - m)p_{good}/v_{good})}, \tag{11}$$

where $s \leq \frac{v_{good}+mp_{good}}{np_{good}} \simeq \frac{v_{good}}{np_{good}}$.

This theorem is useful only when s is small. This is intuitive since the probabilities of matching on good/bad words are both high when s is large.

Typically, the size of bad vocabulary is much larger than that of good vocabulary (a variety of possible typos vs. limited number of appropriate labels), i.e., $v_{bad} \gg v_{good}$, and players would be more likely to choose good words, i.e., $p_{good} > p_{bad}$. Assuming $p_{good} = 0.8$, $v_{good} = 10$, $k_1 = 4$, $k_2 = 5$, and the match condition m and the number of players n are set to 2, the probability of matching on bad words when match happens in the first round would be less than 1/65. If match happens in the 5th round, the probability would be less than 1/24.04. This theorem is proved as follows.

Proof: According to the *independent guess* assumption, a s round play with n players are identical to single round play with $n \cdot s$ players. Thus the probability of matching on good words in round s can be written as

$$\begin{aligned} P_s(\text{good match}) &= v_{good} \textstyle\sum_{i=m}^{n\cdot s} \binom{n\cdot s}{i} (p_{good}\frac{1}{v_{good}})^i (1 - p_{good}\frac{1}{v_{good}})^{n\cdot s - i} \\ &= \textstyle\sum_{i=m}^{n\cdot s} \binom{n\cdot s}{i} \frac{p_{good}^i}{v_{good}^{i-1}} (1 - \frac{p_{good}}{v_{good}})^{n\cdot s-i} \\ &= \textstyle\sum_{i=m}^{n\cdot s} \binom{n\cdot s}{i} f_{1,good}(i) f_{2,good}(i), \end{aligned} \tag{12}$$

where $f_{1,good}(i) = p_{good}^i / v_{good}^{i-1}$ and $f_{2,good}(i) = (1 - p_{good}/v_{good})^{n\cdot s-i}$. Accordingly, we obtain the value of P_s(bad match) by

$$\begin{aligned} P_s(\text{bad match}) &= \textstyle\sum_{i=m}^{n\cdot s} \binom{n\cdot s}{i} \frac{p_{bad}^i}{v_{bad}^{i-1}} (1 - \frac{p_{bad}}{v_{bad}})^i \\ &= \textstyle\sum_{i=m}^{n\cdot s} \binom{n\cdot s}{i} f_{1,bad}(i) f_{2,bad}(i). \end{aligned} \tag{13}$$

The probability of matching on bad words under the condition matches happen in the s-th round is

$$
\begin{aligned}
P_s(\text{bad match}|\text{match}) &= \frac{P_s(\text{bad match})}{P_s(\text{good match})+P_s(\text{bad match})} \\
&= \frac{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i} f_{1,bad}(i) f_{2,bad}(i)}{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i}(f_{1,bad}(i)f_{2,bad}(i)+f_{1,good}(i)+f_{2,good}(i))}.
\end{aligned} \tag{14}
$$

Before continuing on the proof, we simplifies two ratios:

$$
\frac{f_{1,good}(i)}{f_{1,bad}(i)} = \frac{p_{good}^i / v_{food}^{i-1}}{p_{bad}^i / v_{bad}^{i-1}} = k_1^i k_2^{i-1}. \tag{15}
$$

$$
\begin{aligned}
\frac{f_{2,good}(i)}{f_{2,bad}(i)} &= \frac{(1-p_{good}/v_g ood)^{n\cdot s-i}}{(1-p_{bad}/v_{bad})^{n\cdot s-i}} \\
&\geq (1-p_{good}/v_g ood)^{n\cdot s-i} \\
&= 1-(n\cdot s-i)p_{good}/v_{good}+\binom{n\cdot s-i}{2}(p_{good}/v_{good})^2-\ldots \\
&\geq 1-(n\cdot s-i)p_{good}/v_{good}, \text{if } s \leq \frac{v_{good}+ip_{good}}{np_{good}}.
\end{aligned} \tag{16}
$$

According to the simplifications above, we obtain

$$
\begin{aligned}
\frac{f_{1,bad}(i)f_{2,bad}(i)}{f_{1,bad}(i)f_{2,bad}(i)+f_{1,good}(i)f_{2,good}(i)} &= \frac{1}{1+\frac{f_{1,good}(i)}{f_{1,bad}(i)}\frac{f_{2,good}(i)}{f_{2,bad}(i)}} \\
&\leq \frac{1}{1+k_1^i k_2^{i-1}(1-(n\cdot s-i)p_{good}/v_{good})}.
\end{aligned} \tag{17}
$$

Let $K_i = \frac{1}{1+k_1^i k_2^{i-1}(1-(n\cdot s-i)p_{good}/v_{good})}$, then $f_{1,bad}(i)f_{2,bad}(i)+f_{1,good}(i)f_{2,good}(i) \geq K_i \cdot f_{1,bad}(i)f_{2,bad}(i)$. Under the assumption that $s \leq \frac{v_{good}+mp_{good}}{np_{good}}$, we can conclude that

$$
\begin{aligned}
P_s(\text{bad match}|\text{match}) &\leq \frac{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i} f_{1,bad}(i)f_{2,bad}(i)}{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i} K_i \cdot f_{1,bad}(i)f_{2,bad}(i)} \\
&\leq \frac{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i} f_{1,bad}(i)f_{2,bad}(i)}{\sum_{i=m}^{n\cdot s}\binom{n\cdot s}{i} K_m \cdot f_{1,bad}(i)f_{2,bad}(i)} \\
&= \frac{1}{1+k_1^m k_2^{m-1}(1-(n\cdot s-i)p_{good}/v_{good})}.
\end{aligned} \tag{18}
$$

4.1.4. Efficiency, Quality, and Utility

Here we explain how we evaluate the productivity of an ESP game. We define *the efficiency of an ESP game as the rate that labels are matched for the given images*. If the number of participants remains the same, higher efficiency indicates that the system is more productive given the same amount of resources. Thus, we consider game settings that lead to higher efficiency as more desirable. In addition, we define *the quality of an ESP game as the proportion of good labels among all the matched labels.* Higher quality indicates that the matched labels are more likely to be appropriate descriptions of the target puzzle. Thus, we naturally seek game settings that yield high-quality matched labels.

However, there is often a trade-off between efficiency and quality in a real system because configurations that yield higher efficiency often lead to lower quality; conversely, settings that yield higher quality may impact on the level of efficiency. For this reason, we define *the utility of an ESP game as the product of the game's efficiency and quality.* This definition enables us to explain utility as *the throughput rate of good labels produced by an ESP game.*

Based on the probabilistic model presented in this section, we can write the formula of the efficiency, quality, and utility of an ESP game as follows:

$$Efficiency = \frac{E\left(N_{good}\left(s\right) + N_{bad}\left(s\right)\right)}{E\left(s\right)}; \tag{19}$$

$$Quality = \frac{E\left(N_{good}\left(s\right)\right)}{E\left(N_{good}\left(s\right) + N_{bad}\left(s\right)\right)}; \tag{20}$$

$$Utility = Efficiency \times Quality = \frac{E\left(N_{good}\left(s\right)\right)}{E\left(s\right)}. \tag{21}$$

4.2. System Gain Modeling of Generalized ESP Games

To be effective, generalized ESP games try to collect outcomes with the largest possible aggregated score for each puzzle (image), and thus needs as many distinct puzzles as possible to be played. There is a trade-off between these two aspects. On the one hand, the system prefers to take as many labels as possible for each puzzle, which will result in the playing of fewer distinct puzzles; on the other hand, the system prefers that each puzzle is played only once, which can lead to the playing of the maximum number of puzzles. 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 [34, 35, 36], and propose a metric to evaluate the *system gain* of generalized ESP games. Then, we analyze the puzzle selection problem. We discuss the analysis in this subsection.

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 gain*, G, of generalized ESP games as follows:

$$G = ln(N) \times ln(S). \tag{22}$$

Specifically, $ln(N)$ and $ln(S)$ denote, respectively, the two performance aspects of the games described earlier[4]. The metric G is designed to evaluate how well the games accommodate both performance aspects simultaneously.

Clearly, the *system gain* increases as the number of the games played increases, and/or as the average total score (per puzzle) increases. In the system, each label is associated with positive score value V, and in previous subsection 4.1, we define k, number of labels per round. Thus, we know that $S = V \times k \times r$. Suppose the N puzzles have been played T rounds in total (one puzzle per round), and each puzzle has been played r times on average ($N = T/r$). We can then rewrite Equation 22 as follows:

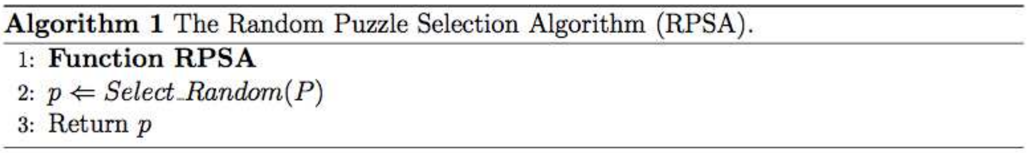
$$\begin{aligned} G &= ln(T/r) \times ln(V \times k \times r) \\ &= (ln(T) - ln(r)) \times (ln(V \times k) + ln(r)) \\ &= -(ln(r))^2 + (ln(T) - ln(V \times k))ln(r) + ln(T)ln(V \times k) \\ &= -\left(ln(r) - \frac{ln(T) - ln(V \times k)}{2}\right)^2 + C, \end{aligned} \tag{23}$$

where C is a constant with a value equal to $ln(T)ln(V \times k) + \left(\frac{ln(T)-ln(V\times k)}{2}\right)^2$. Note that C also represents the largest possible *system gain*, which occurs when

$$r = e^{\frac{ln(T)-ln(V\times k)}{2}}. \tag{24}$$

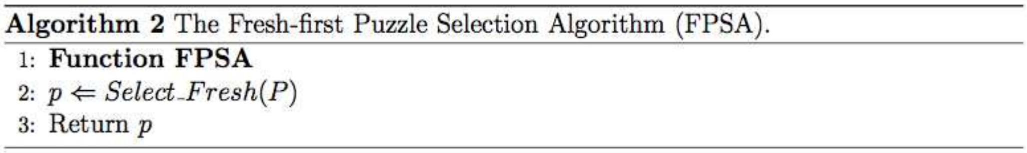
Next, we design three puzzle selection algorithms for the generalized ESP game, namely the *Random Puzzle Selection Algorithm* (RPSA), the *Fresh-first Puzzle Selection Algorithm* (FPSA), and the proposed *Optimal Puzzle*

[4] Note that we use the *natural logarithmic scale* for both factors in G because it has been shown that the *logarithmic scale* is more intuitive and appropriate in number-space mapping [37]. 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).

```
Function RPSA
$p \Leftarrow Select\_Random(P)$
Return $p$
```



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

```
Function FPSA
$p \Leftarrow Select\_Fresh(P)$
Return $p$
```

Selection Algorithm (OPSA). Specifically, we take the RPSA scheme's performance as the baseline (in terms of *system gain*). Meanwhile, the heuristics-based FPSA scheme tries to maximize the first component of Equation 22 (i.e., $ln(N)$), and the OPSA scheme tries to achieve the largest possible *system gain* based on our analysis.

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. We present the three algorithms in the following.

- **RPSA and FPSA**

  We present the *Random Puzzle Selection Algorithm* (RPSA) and the *Fresh-first Puzzle Selection Algorithm* (FPSA) in Algorithms 1 and 2 respectively. RPSA selects a puzzle at random from the puzzle pool P in each round.[5] 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 22.

- **The Proposed Scheme: OPSA**

[5] The random puzzle selection algorithm is implemented in the ESP game [3], but it was called RPSA for the first time in [38].

Table 2: Default Values of Variables

Name	**Default value**
n	2
m	2
d	1000
T	$10,000$
v_{good}	20
v_{bad}	$d - v_{good}$
$prob_{good}$	0.8
$prob_{bad}$	$1 - prob_{good}$

In the proposed Optimal Puzzle Selection Algorithm (OPSA), 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. 4.2); and for each entry p of P, $p.r$ represents the round number in which 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 OPSA algorithm in Algorithm 3.

5. Evaluation

5.1. Probabilistic Model Validation

In this subsection, we describe the simulations used to validate our model. After explaining the simulation setup, we compare the utility computed by our model with that derived in the simulations. The effects of various game parameters on the game's utility are also considered.

5.1.1. Simulation Setup

We designed our simulator based on the rules of ESP games. In each round, there are n players, each of which randomly selects a good word with probability $prob_{good}$, and a bad word with probability $prob_{bad}$. At the end

Algorithm 3 The Optimal Puzzle Selection Algorithm (OPSA).

```
**Function OPSA**
$T \Leftarrow T + 1$
$r' \Leftarrow \lceil e^{\frac{ln(T)-ln(V \times k)}{2}} \rceil$
**if** $r' > r$ **then**
  **for** each $p$ in $P_2$ **do**
    **if** $p.r < r'$ **then**
      Move $p$ from $P_2$ to $P_1$
    **end if**
  **end for**
  $P_1 \Leftarrow P_1 \bigcup P_2$
  $r \Leftarrow r'$
**end if**
**if** $\{P_1\}$ is NOT empty **then**
  $p \Leftarrow Select\_Played(P_1)$
  $p.r \Leftarrow p.r + 1$
  **if** $p.r = r$ **then**
    Move $p$ from $P_1$ to $P_2$
  **end if**
  Return $p$
**else**
  **if** $\{P_0\}$ is NOT empty **then**
    $p \Leftarrow Select\_Random(P_0)$
    $p.r \Leftarrow 1$
    **if** $p.r < r$ **then**
      Move $p$ from $P_0$ to $P_1$
    **else**
      Move $p$ from $P_0$ to $P_2$
    **end if**
    Return $p$
  **else**
    $p \Leftarrow Select\_Fresh(P_2)$
    $p.r \Leftarrow 1$
    Return $p$
  **end if**
**end if**
```

of each round, the simulator checks the number of matches to determine whether the current puzzle has been solved. If m matches are found, all the players' guesses are erased to simulate that the participants are trying to solve a new puzzle; otherwise, the simulator just advances to the next round. The simulator assumes the number of puzzles is infinite, and that there are always n players ready to participate in a game. The simulations end after running for T rounds, no matter how many puzzles have been solved. We then compute the average efficiency, quality, and utility of the matches based on the time taken to solve each puzzle and the number of good and bad matches recorded during the simulations.

To investigate the accuracy of our model under different settings, we change the parameters and observe whether the simulated quantity of good and bad matches is identical to or close to that computed by our analytical model. Specifically, we change the four major variables, i.e., the number of players, n; the consensus threshold, m; the size of the good vocabulary, v_{good}; and the probability that the players will guess a good word, $prob_{good}$. When evaluating the effect of one variable, the other three are set to their default values[6], as shown in Table 2. Moreover, when we adjust the consensus threshold, we set the number of players at 20, as the consensus threshold must be no greater than the number of players.

5.1.2. Validation by Utility Curves

Although we have defined three key characteristics of generalized ESP games, namely, the efficiency, quality, and utility, we only validate the accuracy of our model by a game's utility. This is because the magnitude of the utility depends on the efficiency and the quality; thus, the utility is unlikely to be correct if the values of the other two characteristics are incorrect. Since our objective is to optimize the utility of ESP games by changing the game settings, the model's accuracy in predicting a game's utility should be examined more carefully.

In the following, we investigate how the utility of ESP games changes under different stopping conditions, k. As shown in Figure 1, the utility

[6] Note that, since the objective of this study is to investigate the intrinsic properties of generalized ESP games, rather than those specific to the original ESP game, we decide to configure the parameters of our simulation by ourselves; however, in practice, some of the parameters, such as n, T, and $prob_{good}$, can be calculated directly from the ESP game dataset [39].

reaches its maximum when $n = 2$ and $k = 10$. As the number of participants increases, the shapes of the utility curves change slightly, and the optimal stopping condition shifts slightly to the lower k values. The concave shape of the utility curve indicates that, as k increases, there should be a tradeoff between the efficiency and quality of ESP games such that the utility curve is not monotonic. To demonstrate the tradeoff between efficiency and quality, we plot the values of all three characteristics in Figure 2. Clearly, the game's efficiency increases as k increases, while its utility decreases. The utility reaches the highest point when k is around 15.

We now consider the effects of the other parameters on the utility curves of ESP games, and check the correspondence between the results derived by our model and those of the simulations. The effects of the consensus threshold, the size of the good vocabulary, and the probability that players will guess a good word are investigated. However, because of space limitations, Figure 3 only shows the effect of the consensus threshold. For all the parameters, the utility curves computed by our model are very close to those derived by the simulations. We observe that the consensus threshold and the size of the good vocabulary have a strong effect on the optimal stopping condition, while the number of participants and the probability of choosing good words have relatively little effect.

5.1.3. Effect of Game Settings

We now examine the effect of various game settings on the game's utility. The relationships between the utility and different game parameters are shown in Figure 4. Figure 4(a) shows that if more players participate in a game simultaneously, the matching rate of good words increases faster than linearly, as the number of guess-pairs grows quadratically. In contrast, if the consensus threshold is raised, as shown in Figure 4(b), the game's utility declines exponentially, but the quality of the matched results increases. The size of the good vocabulary also has a substantial impact on the game's utility. Figure 4(c) shows that the utility gradually decreases as the size of the good vocabulary increases because of the lower probability that two participants will guess the same good word. Finally, as expected, the game's utility increases linearly as the probability of guessing good words rises. Note that, in all the graphs, the utility scores computed via simulations and by our model match closely, which demonstrates the accuracy of our analytical model.

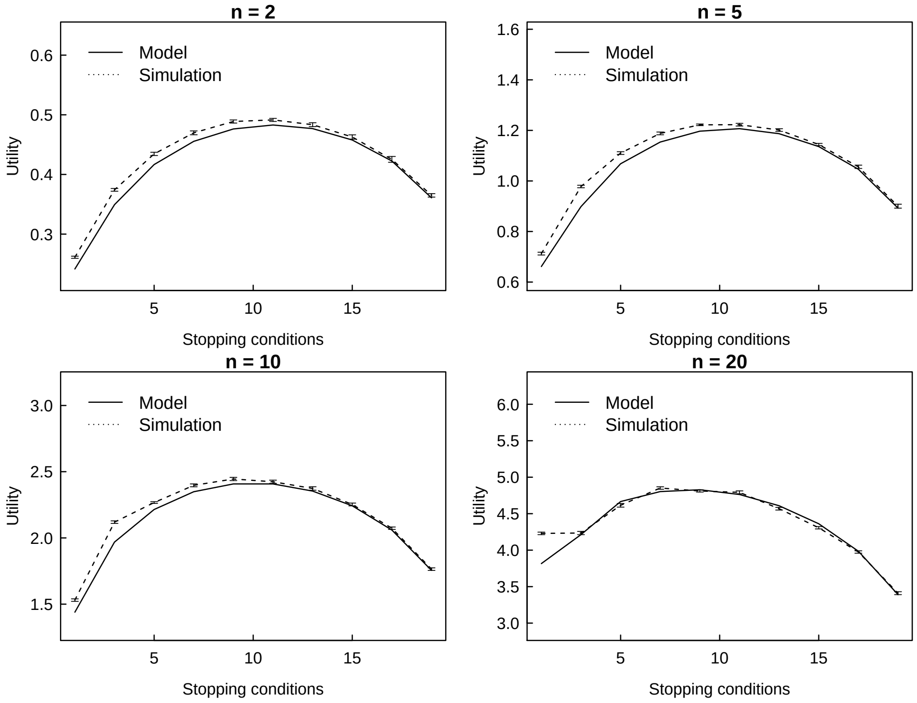


Figure 1: The relationships between utility and stopping conditions under different n. The vertical bars on the simulation curve denote the 95% confidence band of the average utility.

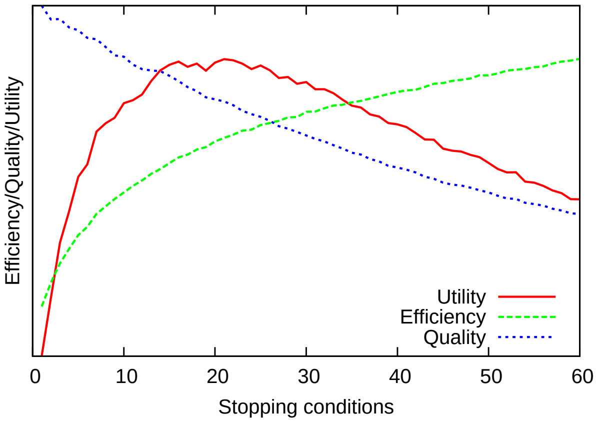


Figure 2: The relationships between efficiency, quality, and utility in an ESP game.

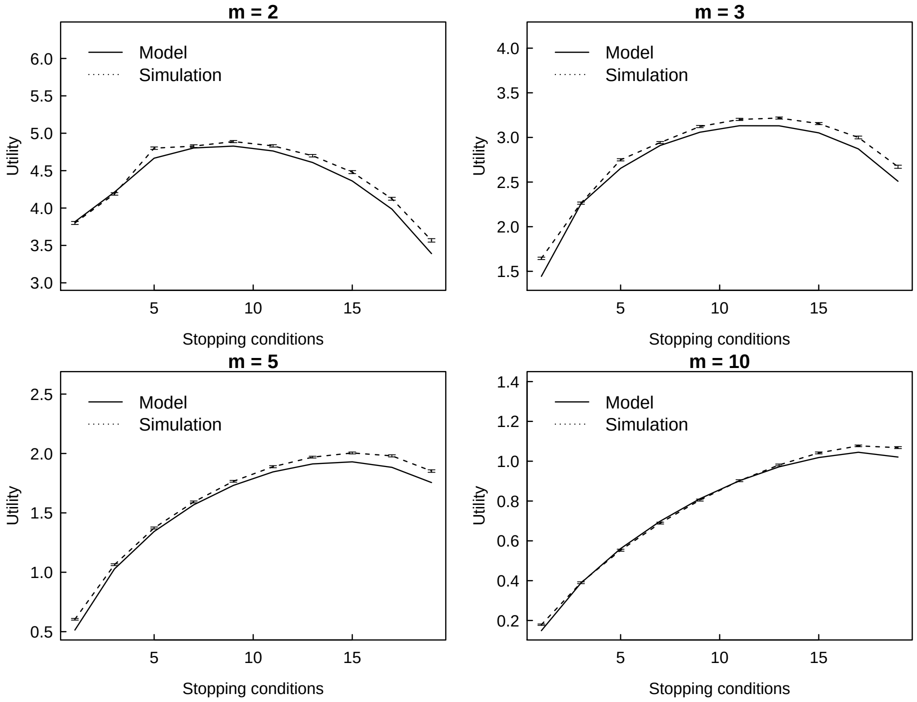


Figure 3: The relationships between utility and stopping conditions under different m. The vertical bars on the simulation curve denote the 95% confidence band of the average utility.

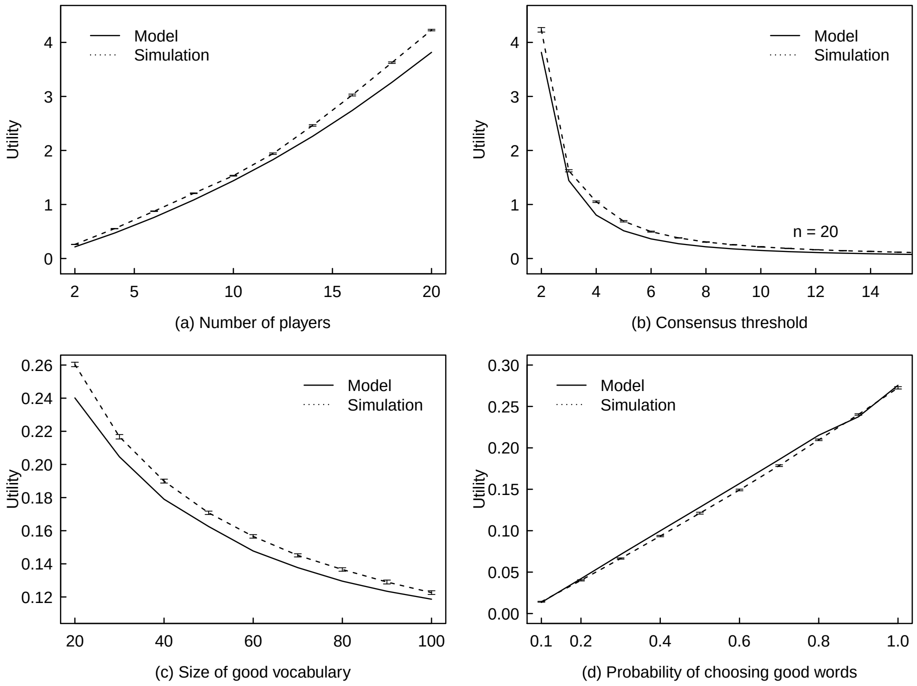


Figure 4: The effect of other parameters on utility. The vertical bars on the simulation curve denote the 95% confidence band of the average utility.

5.2. Optimal Stopping Conditions

In this subsection, we focus on how to set the stopping condition to maximize an ESP game's utility. We explain the derivation of the optimal stopping conditions, and discuss how they change under different configurations. In addition, we examine how our optimization method improves the game's utility.

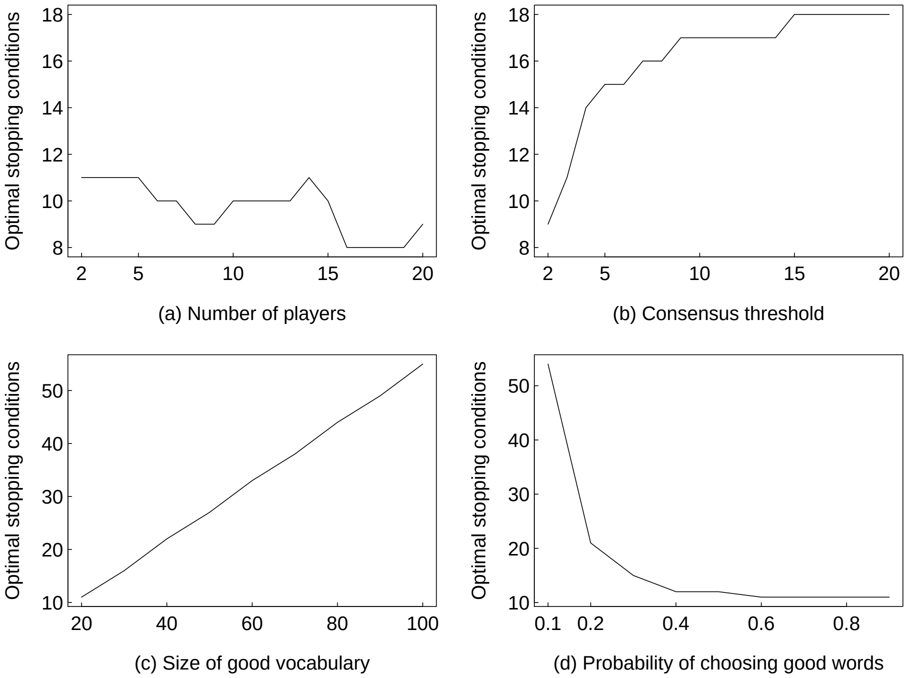
5.2.1. Computation

The utility equation of our model (Equation 21) is a discontinuous function, so we cannot obtain its optimal point by differentiating the function with respect to the stopping conditions. Therefore, we derive it in a numerical way. From Section 5.1.2, we know that the utility function that takes the stopping condition, k, as the only parameter is a unimodal function. In addition, the domain of k is a positive integer, which is usually small (less than 100 in most of our scenarios). Thus, we use an exhaustive search to find the maximum utility within a reasonable range, say, from 1 to 200. In our implementation, this exhaustive search process takes only a few seconds on a commodity PC (with Intel Pentium Dual-Core 2.0GHz and 1 GB RAM).

5.2.2. Effect of Parameters

Here, we consider the effect of different parameters on the optimal stopping conditions. Interestingly, the number of participants does not affect the optimal stopping conditions, as shown in Figure 5(a). This is reasonable because the probability of good matches and bad matches remains the same regardless of the number of players, which only affects the rate of label matching. The consensus threshold, on the other hand, affects the optimal stopping conditions significantly when it increases, as shown in Figure 5(b). This behavior can be explained by the occurrence probability of good matches relative to that of bad matches. Raising the consensus threshold makes label matching more difficult; however, the advantage is that matching bad labels will become relatively more difficult than matching good labels. Therefore, when the consensus threshold increases, the matching rate of good labels will grow faster than that of bad labels; consequently, the optimal stopping condition is deferred to allow more good words to be matched before finishing the puzzle.

The size of the good vocabulary and the probability of choosing good words have similar impacts on the optimal stopping conditions. Both increasing the number of good words and reducing the probability of choosing good



(a) Number of players

(b) Consensus threshold

(c) Size of good vocabulary

(d) Probability of choosing good words

Figure 5: The effect of the parameters on the optimal stopping conditions

words increase the optimal stopping conditions because they make matching good labels more difficult. Thus, a relatively late stopping condition is required in order to increase the proportion of good matches.

5.2.3. Benefit of Optimization

To demonstrate how optimization improves the game's achieved utility, we examine the gain derived by adopting the optimal stopping condition suggested by our model. We define the *utility gain* as the ratio of the utility of an optimized game to that of a simple ESP game, i.e., with the stopping condition set to 1.

The relationships between the utility gain and various game parameters are shown in Figure 6. We observe that, the optimization achieved by adopting the optimal stopping condition generally provides a utility boost that is 2 or more times higher than that of the simple ESP game. Even if we consider a more conservative scenario, where only two participants play the game and the consensus threshold is set to 2, the utility gain will be around 2, assuming the number of good words is 20 and the probability of choos-

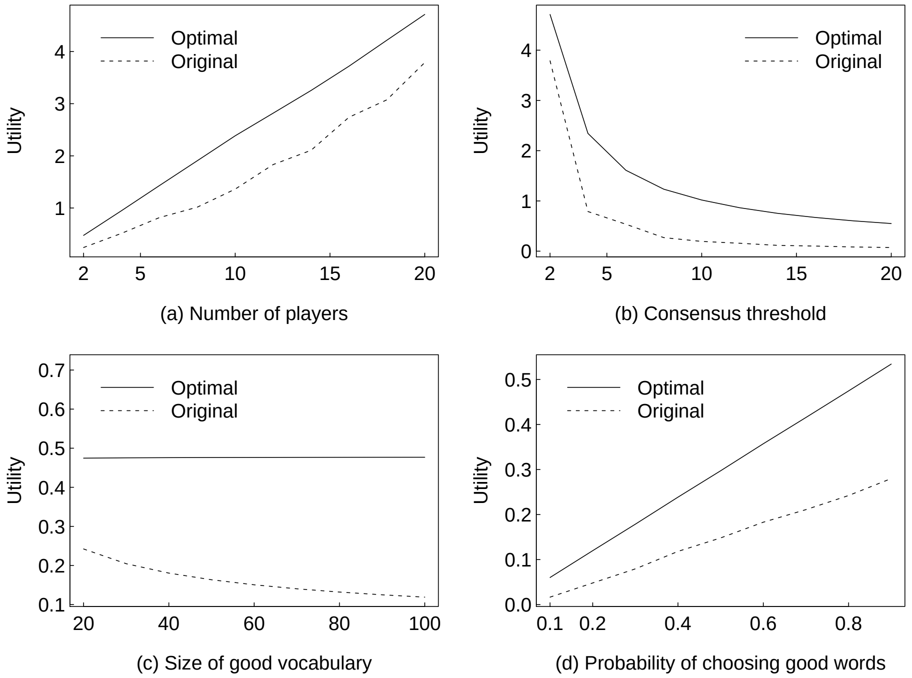


Figure 6: The effect of the parameters on the improvement in utility.

ing good words is 0.8. Moreover, the utility gain increases rapidly as either the consensus threshold or the size of the good vocabulary increases. The utility gain is only significantly lower than 2 when the number of participants is much higher than 2. However, we can still achieve a utility gain of around 1.3, even the number of players is as high as 20. These findings demonstrate that the utility optimization provided by our analytical model can generally provide twice as much utility as a non-optimized game, which stops immediately after a label has been matched.

5.3. Evaluation of the System Gain and Puzzle Selection Algorithm

In this subsection, we discuss the simulations performed to investigate the properties of the ESP game based on our analysis. We also evaluate the *system gain* of the three puzzle selection strategies. All the results are based on the average performance of 100 simulations.

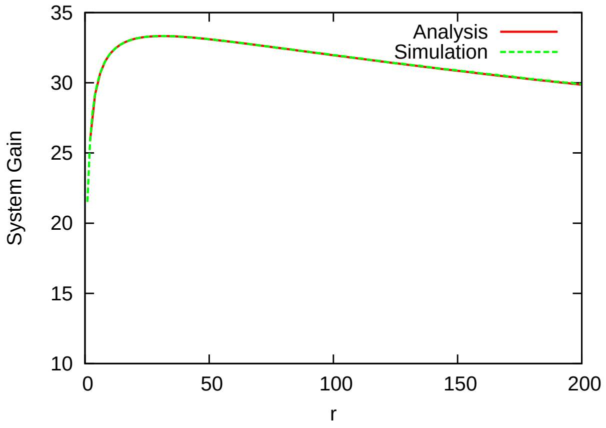


Figure 7: Comparison of the system gain under various r settings in both the simulations and the analysis. ($V = 10.3353$, $k = 1$ and $T = 10,000$)

5.3.1. The Optimal r

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 7 shows the evaluation results in terms of *system gain* for r values between 1 and 200, when k was fixed at 1 and the score value V was fixed at 10.3353, which was the same as the value used in [38]. In the figure, the analysis curve is derived by Equation 23. We observe that the analysis curve matches the simulation curve very well, and the optimal r values (i.e., those that yielded the largest *system gain*) of the two curves are also comparable.

Additionally, we compared the derived optimal r values with different V values using both simulations and analysis, as shown in Figure 8. We observe that, the optimal r value decreases as the V value increases. Then, we want to observe the effect caused by k to r. We changed k from 1 to 10 and compared the derived optimal r values using both simulations and analysis. The result is shown in Figure 9. When k value increases, the optimal r values will decrease at the same time. As a result, based on Equation 24, the optimal r value will decrease as V or k increases. We find that if there are more scores generated in per round, less rounds of each puzzle need be

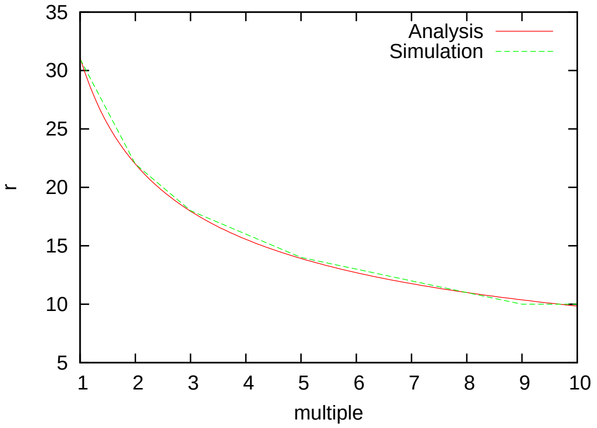


Figure 8: Comparison of the optimal r values derived by simulations and analysis, where $T = 10,000$, $k = 1$, and V varies from 1 to 10 multiples of 10.3353.

played in order to achieve a better overall *system gain*.

5.3.2. 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 gain* r in the proposed Optimal Puzzle Selection Algorithm. Figures 10 and 11 show the comparison results of r and N with various T values in the range 200 to 20,000 (V is fixed at 10.3353 and k is fixed at 1).

Figures 10 and 11 show that our analytical model matches the simulation results very well in all cases. In addition, we observe that both of the r and N values increase as the value of T 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) a larger number of 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.

5.3.3. 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 $T = 10,000$, $V = 10.3353$, and $k = 1$; M denotes the total number of puzzles in the system. The simulation results are shown in Figure 12.

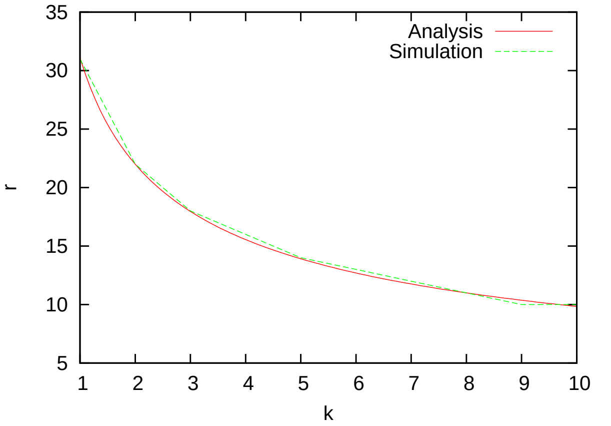


Figure 9: Comparison of the optimal r values derived by simulations and analysis, where $T = 10,000$, $V = 10.3353$, and k varies between 1 and 10.

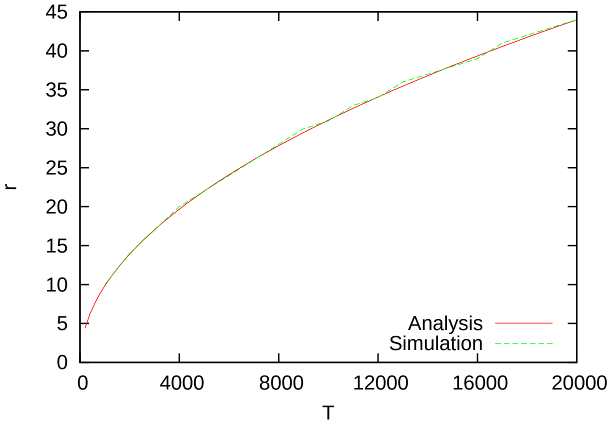


Figure 10: Comparison of the optimal r values derived by simulations and analysis, where $V = 10.3353$, $k = 1$ and T varies between 200 and 20,000.

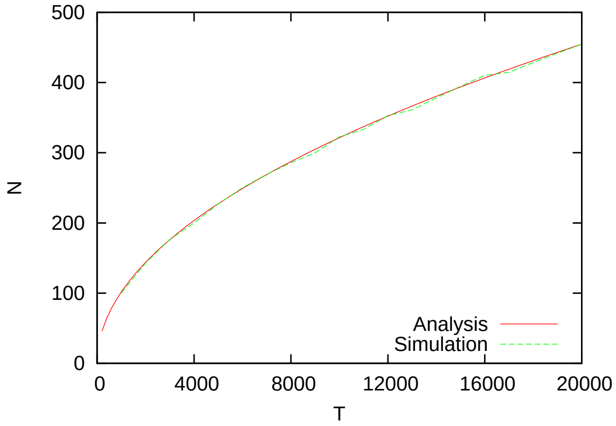


Figure 11: Comparison of the N values derived by the simulations and analysis, where $V = 10.3353$, $k = 1$ and T varies between 200 and 20,000.

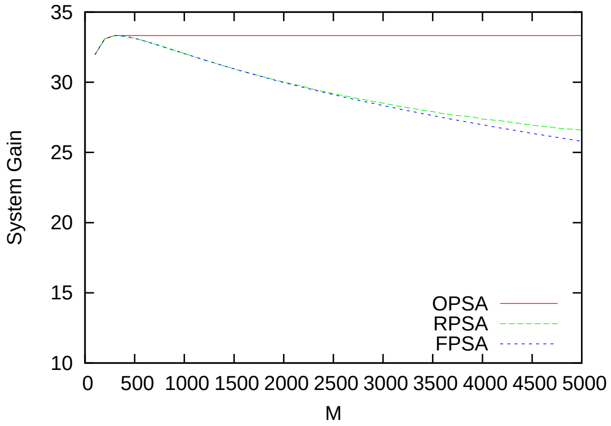


Figure 12: Comparison of the system gain achieved by the OPSA, FPSA and RPSA schemes with various numbers of puzzles, where T is fixed at 10,000, k is set to 1, and V is set to 10.3353.

The results in Figure 12 show that, when M is small (say, smaller than a threshold M'), the three algorithms are comparable in terms of the *system gain* achieved. However, when M is larger than M', the *system gain* of OPSA remains consistent regardless of the changes in the values of M. In contrast, the *system gain* 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 gain* (i.e., $M' = N = T/r$). Since $T = 10,000$, $V = 10.3353$, and $k = 1$, we know that $r = 31$. Therefore, $M' = 10000/31 \approx 321$ in this case. The results indicate that, when using the OPSA scheme, the ESP game must maintain at least a certain number of puzzles to achieve the maximum *system gain*[7]; otherwise, it will favor the RPSA and FPSA schemes because their performance is comparable to that of OPSA and they are easy to implement.

6. Discussion

We have presented the analysis and modeling of generalized ESP games, and performed a rich set of evaluation to verify the proposed model. However, there are some issues that have yet to be addressed. Here, we discuss the effects of the assumptions used by our analytical model and some issues that may occur when applying our analysis in real-life ESP games.

6.1. Model Assumptions

One major assumption of our analysis is that the guesses made by each player are independent of each other. In practice, players remember what labels they have already used and avoid submitting duplicate guesses; moreover, the taboo words provided in each puzzle give players hints implicitly in their guessing [31]. However, considering the "memory" and "hint" effects would make the analytical modeling too complicated to manage. Thus, we adopt the independent guess assumption and examine its impact on the model's accuracy by simulations.

To demonstrate that our model provides a reasonable solution for utility optimization, we show the optimal utility achieved by different models and simulations respectively in Figure 13. Because we do not actually construct

[7] 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.

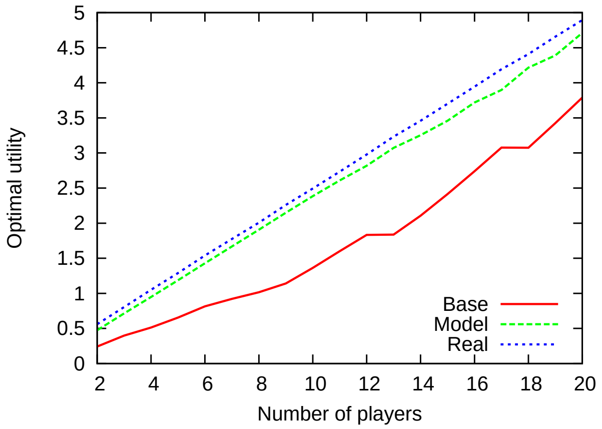


Figure 13: The optimal utility achieved by ideal modeling, independent modeling, and simple games without optimization.

an ideal model that takes the memory effect into consideration, we compute its output by simulations. On the graph, the three curves in the figure represent the optimal utility achieved by the ideal model, by our model with the independent guess assumption, and by simple games in which k is set to 1. The results show that both models yield much higher utility than the simple games. Even though our model does not provide as high utility as the ideal model, the games that adopt the stopping condition suggested by our model still achieve near optimal utility. In view of the complexity of modeling with the memory effect, we consider that our independent guess assumption is a reasonable tradeoff between the model's computational complexity and the degree of optimization we are pursuing.

Another assumption of our model is that players uniformly guess words in the vocabulary pool. In practice, players may guess according to some preferences. For example, they may prefer to guess more common, shorter words first, or guess more specific words first, because they think a particular strategy would lead to consensus more quickly. Besides, players may tend to guess similar words, i.e., tags with dependences. Players' strategies in prioritizing their label choices may significantly impact the outcomes of ESP games. In addition, the situation becomes more complex when tag dependences are considered or players with different strategies are assigned to the same game. In this paper, we leave these issues to future work because 1) it would make the analytical modeling intractable; 2) though previous work showed the existence of tag dependence, there is no established model of the correlation structure of the tags proposed by users. Modeling users' tagging behavior with temporal dependence is itself an open issue, which may require joint efforts from computer scientists, cognitive psychologists and linguistists. Thus, we leave this issue in the future work, and instead, we decide to perform an extensive simulation to examine the impact of the tag dependence issue on out model's accuracy.

6.2. Choice of Parameters

To put our model to real use, we must first address the problem of how to choose the model's parameters, especially the size of the good vocabulary and the probability that players will guess a good word. We believe that these parameters could be measured *empirically* from real-life observations. Specifically, one can take the average number of labels on which there has been a consensus in a large number of games as the size of the good vocabulary. Accordingly, one can compute the probability that players will guess a good word by the ratio of guesses that fall into the set of the good vocabulary. While the parameters may be different due to the types of puzzles and the composition of the participants, an empirical choice of parameters like this would be the most appropriate way to achieve accurate modeling results and thereby optimize the utility of games.

6.3. Time Consumption

The proposed analytical model only considers the number of the games played and the quantity/quality of the game outcomes when measuring the *system gain*. 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 ESP games (i.e., the average quantity and/or quality of the outcomes in each time unit). A possible solution to this issue is to consider an additional factor, i.e., the time consumption of the played games in Equation 22. We defer a detailed evaluation of this issue to a future work.

7. Conclusion

We have proposed a generalized ESP game in which the number of players, the consensus threshold, and the stopping condition are variable. In addition, we have presented an analytical model that computes the efficiency, quality, and utility of a generalized ESP game given the game's settings. Via extensive simulations, we show that by applying the optimal stopping condition predicted by our model, the game's utility will be usually be at least 2 times higher than that of a non-optimized game. This feature can be leveraged by game service providers to improve the utilization of finite player efforts in order to maximize both the efficiency and quality of the matched labels.

In addition, we have proposed an evaluation metric, called *system gain*, to evaluate the game's performance. Moreover, we argue that GWAP systems need to be *designed and played with strategies* in order to collect human intelligence in a more efficient manner. Based on our analysis, we propose and implement an Optimal Puzzle Selection Algorithm (OPSA) to provide guidelines for improving generalized ESP games. Using a comprehensive set of simulations, we have investigated the properties of ESP games, and demonstrated that the proposed OPSA scheme substantially outperforms other schemes in all test cases.

Acknowledgement

This research was supported by the National Science Council of Taiwan under grant numbers: NSC 98-2221-E-001-014-MY3, NSC 100-2631-H-001-013, and NSC 100-2631-S-003-006.

References

[1] L. von Ahn, Games with a Purpose, IEEE Computer 39 (6) (2006) 92–94.

[2] L. von Ahn, L. Dabbish, Designing games with a purpose, Communications of the ACM 51 (8) (2008) 58–67.

[3] L. von Ahn, L. Dabbish, Labeling Images with a Computer Game, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2004, pp. 319–326.

[4] L. von Ahn, R. Liu, M. Blum, Peekaboom: A Game for Locating Objects in Images, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2006, pp. 55–64.

[5] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, LabelMe: A Database and Web-Based Tool for Image Annotation, International Journal of Computer Vision 77 (1-3) (2008) 157–173.

[6] Herd It, http://www.herdit.org/.

[7] E. Law, L. von Ahn, Input-agreement: A New Mechanism for Data Collection using Human Computation Games, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2009, pp. 1197–1206.

[8] L. Barrington, D. Turnbull, D. O'Malley, G. Lanckriet, User-centered design of a social game to tag music, in: Proceedings of ACM Human Compuation Workshop, 2009, pp. 7–10.

[9] Major Miner, http://majorminer.org/.

[10] H. Lieberman, D. Smith, A. Teeters, Common Consensus: a web-based game for collecting commonsense goals, in: Proceedings of ACM Workshop on Common Sense for Intelligent Interfaces, 2007.

[11] L. von Ahn, M. Kedia, M. Blum, Verbosity: A Game for Collecting Common-Sense Facts, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2006, pp. 75–78.

[12] A. J. Quinn, B. B. Bederson, Human Computation: A Survey and Taxonomy of a Growing Field, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2011, pp. 1403–1412.

[13] S. Thaler, K. S. E. Simperl, C. Hofer, A Survey on Games for Knowledge Acquisition, Tech. Rep. STI TR 2011-05-01, Semantic Technology Institute (2011).

[14] M.-C. Yuen, L.-J. Chen, I. King, A Survey of Human Computation Systems, in: Proceedings of IEEE International Conference on Computational Science and Engineering, 2009, pp. 723–728.

[15] Google Image Labeler, http://images.google.com/imagelabeler/.

[16] J. P. Bigham, R. S. Kaminsky, R. E. Ladner, O. M. Danielsson, G. L. Hempton, WebInSight: making web images accessible, in: Proceedings

of ACM SIGACCESS conference on Computers and accessibility, 2006, pp. 181–188.

[17] GWAP, http://www.gwap.com/gwap/.

[18] L. von Ahn, S. Ginosar, M. Kedia, R. Liu, M. Blum, Improving accessibility of the web with a computer game, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2006, pp. 79–82.

[19] L. von Ahn, S. Ginosar, M. Kedia, M. Blum, Improving Image Search with PHETCH, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, pp. IV–1209 – IV–1212.

[20] E. Xiao, D. Houser, Emotion Expression in Human Punishment Behavior, Proceedings of the National Academy of Sciences of the United States of America 102 (20) (2005) 7398–7401.

[21] D. Houser, E. Xiao, Classification of Natural Language Messages using a Coordination Game, Experimental Economics 14 (1) (2011) 1–14.

[22] J. Tenzer, P. Stevens, GUIDE: Games with UML for Interactive Design Exploration, Knowledge-Based Systems 20 (7) (2007) 652–670.

[23] R. Metoyer, S. Stumpf, C. Neumann, J. Dodge, J. Cao, A. Schnabel, Explaining How to Play Real-Time Strategy Games, Knowledge-Based Systems 23 (4) (2010) 295–301.

[24] A. Ferreira, J. Atkinson, Designing a feedback component of an intelligent tutoring system for foreign language, Knowledge-Based Systems 22 (7) (2009) 496–501.

[25] M. Bell, S. Reeves, B. Brown, S. Sherwood, D. MacMillan, J. Ferguson, M. Chalmers, Eyespy: Supporting Navigation through Play, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2009, pp. 123–132.

[26] S. Casey, B. Kirman, D. Rowland, The gopher game: a social, mobile, locative game with user generated content and peer review, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology, 2007, pp. 9–16.

[27] L. Grant, H. Daanen, S. Benford, A. Hampshire, A. Drozd, C. Greenhalgh, MobiMissions: the game of missions for mobile phones, in: Proceedings of ACM SIGGRAPH 2007 educators program, 2007, p. Article No.: 12.

[28] S. Matyas, C. Matyas, C. Schlieder, P. Kiefer, H. Mitarai, M. Kamata, Designing Location-based Mobile Games With A Purpose: Collecting Geospatial Data with CityExplorer, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology, 2008, pp. 244–247.

[29] P. Shenoy, D. S. Tan, Human-aided computing: utilizing implicit human processing to classify images, in: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, 2008, pp. 845–854.

[30] C. Gentry, Z. Ramzan, S. Stubblebine, Secure distributed human computation, in: Proceedings of ACM conference on Electronic commerce, 2005, pp. 155–164.

[31] I. Weber, S. Robertson, M. Vojnovic, Rethinking the ESP Game, Tech. rep., Microsoft Research MSR-TR-2008-132 (2008).

[32] S. Jain, D. C. Parkes, A Game-Theoretic Analysis of Games with a Purpose, in: Proceedings of the International Workshop on Internet and Network Economics, 2008, pp. 342–350.

[33] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-J. Hsu, K.-T. Chen, KissKissBan: A Competitive Human Computation Game for Image Annotation, in: Proceedings of ACM Human Computation Workshop, 2009, pp. 11–14.

[34] S. H. Bokhari, Assignment Problems in Parallel and Distributed Computing, Springer, 1987.

[35] T. L. Casavant, J. G. Kuhl, A taxonomy of scheduling in general-purpose distributed computing systems, IEEE Transactions on Software Engineering 14 (2) (1988) 141–154.

[36] V. M. Lo, Heuristic algorithms for task assignment in distributed systems, IEEE Transactions on Computers 37 (11) (1988) 1384–1397.

[37] S. Dehaene, V. Izard, E. Spelke, P. Pica, Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene Cultures, Science 320 (5880) (2008) 1217–1220.

[38] L.-J. Chen, B.-C. Wang, K.-T. Chen, I. King, J. H.-M. Lee, An analytical study of puzzle selection strategies for the esp game, in: Proceedings of IEEE/WIC/ACM Web Intelligence Conference, 2008, pp. 180–183.

[39] ESP Dataset, http://www.hcomp2009.org/Data.html.