Exploiting Puzzle Diversity in Puzzle Selection for ESP-like GWAP Systems

Yu-Song Syu, Hsiao-Hsuan Yu, and Ling-Jyh Chen
Institute of Information Science, Academia Sinica
GWAP = Games with a Purpose
GWAP = Games with a Purpose
ESP Game - the 1st GWAP system
ESP Game - the 1st GWAP system

PLAYER 1

PLAYER 2
ESP Game - the 1st GWAP system

PLAYER 1

PLAYER 2
ESP Game - the 1st GWAP system

PLAYER 1

GUESSING: CAR

GUESSING: HAT

GUESSING: KID

PLAYER 2
ESP Game - the 1st GWAP system

PLAYER 1
GUESSING: CAR
GUESSING: HAT
GUESSING: KID

PLAYER 2
GUESSING: BOY
GUESSING: CAR
ESP Game - the 1st GWAP system

PLAYER 1

GUESSING: CAR

GUESSING: HAT

GUESSING: KID

PLAYER 2

GUESSING: BOY

GUESSING: CAR

Agreement reached: CAR
Why is it important?

• Some statistics (July 2008)
  – 200,000+ players have contributed 50+ million labels.
  – Each player plays for a total of 91 minutes.
  – The throughput is about 233 labels/player/hour (i.e., one label every 15 seconds)

• Google bought a license to create its own version of the game in 2006 (called Google Image Labeler).
The Ideas behind the ESP Game
The Ideas behind the ESP Game

• **Human Computation** – a new paradigm of applications
  – Outsource computational process to human
  – Exploit “human cycles” to solve the problems that are easy to humans but difficult to computer programs
The Ideas behind the ESP Game

• Human Computation – a new paradigm of applications
  – Outsource computational process to human
  – Exploit “human cycles” to solve the problems that are easy to humans but difficult to computer programs

• Games With A Purpose (GWAP)
  – Pioneered by Dr. Luis von Ahn, CMU
  – Take advantage of people’s desire to be entertained
  – Motivate people to play voluntarily
  – Produce useful metadata as a by-product
The Ideas behind the ESP Game

• **Human Computation** – a new paradigm of applications
  – Outsource computational process to human
  – Exploit “human cycles” to solve the problems that are easy to humans but difficult to computer programs

• **Games With A Purpose (GWAP)**
  – Pioneered by Dr. Luis von Ahn, CMU
  – Take advantage of people’s desire to be entertained
  – Motivate people to play voluntarily
  – Produce useful metadata as a by-product

• **Question: how to evaluate the performance of GWAP systems?**
The state of the art
The state of the art

• The ESP Game has two goals
  – To collect as many labels per puzzle as possible (i.e., *quality*)
  – To solve as many puzzles as possible (i.e., *throughput*)
The state of the art

• The ESP Game has two goals
  – To collect as many labels per puzzle as possible (i.e., *quality*)
  – To solve as many puzzles as possible (i.e., *throughput*)

• Both factors are critical to the performance of the ESP game, but unfortunately they do not complement each other.
The state of the art

• The ESP Game has two goals
  — To collect as many labels per puzzle as possible (i.e., quality)
  — To solve as many puzzles as possible (i.e., throughput)

• Both factors are critical to the performance of the ESP game, but unfortunately they do not complement each other.

• In [14], we formulated the problem as a variant of classic scheduling problems, and proposed an Optimal Puzzle Selection Algorithm (OPSA).
What’s the problem?
What’s the problem?

- The OPSA scheme determines the optimal number of agreements required for all puzzles based on an analytical model [14].
What’s the problem?

• The OPSA scheme determines the optimal number of agreements required for all puzzles based on an analytical model [14].

• It does not consider the puzzle diversity (some puzzles are more productive, and some are hard to solve), which may result in the equality of outcomes problem.
Contribution

• Using realistic game traces, we identify the *puzzle diversity* issue in ESP-like GWAP systems.

• We propose the *Adaptive Puzzle Selection Algorithm* (APSA) to cope with *puzzle diversity* by promoting *equality of opportunity*.

• We propose the *Weight Sum Tree* (WST) to reduce the computational complexity and facilitate the implementation of APSA in real-world systems.

• We show that APSA is more effective than OPSA in terms of the number of agreements reached and the system gain.
Adaptive Puzzle Selection Algorithm

- APSA is inspired by the *Additive Increase Multiplicative Decrease (AIMD)* model of *Transmission Control Protocol (TCP)*.

- APSA selects a puzzle to play based on a weighted value $w_k$, and the probability that the $k$-th puzzle will be selected is $p_k = \frac{w_k}{\sum_{i=1}^{K} w_i}$.
Adaptive Puzzle Selection Algorithm

- APSA is inspired by the Additive Increase Multiplicative Decrease (AIMD) model of Transmission Control Protocol (TCP).

- APSA selects a puzzle to play based on a weighted value $w_k$, and the probability that the $k$-th puzzle will be selected is

$$p_k = \frac{w_k}{\sum_{i=1}^{K} w_i}$$

$$w_k = \begin{cases} 
1 & \text{the initial value,} \\
\frac{w_k + 1}{2} & \text{if agreements are reached,} \\
\frac{w_k}{2} & \text{if no agreements are reached.}
\end{cases}$$

The more productive a puzzle is, the higher probability it will be selected in the next game round.
Implementation Method (1/3)
• The *scalability* issue:
  – The computational complexity increases linearly with the number of puzzles played, i.e., $O(K)$. 
Implementation Method (1/3)

• The *scalability* issue:
  – The computational complexity increases linearly with the number of puzzles played, i.e., \( O(K) \).

• Our solution:
  – We propose a new data structure, called *Weight Sum Tree* (WST), which is a *complete binary tree of partially weighted sums*.
The scalability issue:

- The computational complexity increases linearly with the number of puzzles played, i.e., $O(K)$.

Our solution:

- We propose a new data structure, called Weight Sum Tree (WST), which is a complete binary tree of partially weighted sums.

$$s_i = \begin{cases} w_{i-2^h+1}, & \text{when } 2^{h-1} \leq i < 2^h; \\ s_{2i} + s_{2i+1}, & \text{when } 0 < i < 2^{h-1}. \end{cases}$$

$s_i$: the $i$-th node in the tree

$h$: the height of the tree

$L.\,J.\,Chen$, Exploiting Puzzle Diversity in Puzzle Selection for ESP-like GWAP Systems, WI 2010, Toronto, Canada
Implementation Method (2/3)

• Three cases to maintain the WST
  – After the *k*-th puzzle is played in a game round
    • Update the \( w_k \) and its ancestors: \( O(\log K) \)
  – After a puzzle has been removed (say, the k-th puzzle)
    • Set the \( w_k \) to 0 (to become a virtual puzzle): \( O(\log K) \)
  – After adding a new puzzle (say, the k-th puzzle)
    • Set the \( w_k \) to 1
    • Replace the first (leftmost) virtual puzzle or rebuild the WST:
      \( O(\log K) \) or \( O(K) \)
Three cases to maintain the WST

- After the $k$-th puzzle is played in a game round
  - Update the $w_k$ and its ancestors: $O(\log K)$

- After a puzzle has been removed (say, the $k$-th puzzle)
  - Set the $w_k$ to 0 (to become a virtual puzzle): $O(\log K)$

- After adding a new puzzle (say, the $k$-th puzzle)
  - Set the $w_k$ to 1
  - Replace the first (leftmost) virtual puzzle or rebuild the WST: $O(\log K)$ or $O(K)$
• Determine a random number $r \ (0 \leq r \leq 1)$, and call the function $Puzzle\_Selection(0,r)$

**Algorithm 1** The proposed puzzle selection implementation based on the APSA scheme and the weight sum tree data structure.

1: **Function** $Puzzle\_Selection(k, r)$
2:    **if** $k \geq 2^{h-1}$ **then**
3:        Return the $(k - 2^{h-1} + 1)$th puzzle;
4:    **end if**
5:    **if** $r \leq \frac{s_{2k}}{s_1}$ **then**
6:        $Puzzle\_Selection(2k, r)$;
7:    **else**
8:        $Puzzle\_Selection(2k + 1, r - \frac{s_{2k}}{s_1})$;
9:    **end if**
Evaluation

• We evaluated the APSA scheme using trace-based simulations.

• The game trace was collected by the ESP Lite system.
  – The trace was one-month long (from 2009/3/9 to 2009/4/9).
  – The OPSA scheme was used in 1,444 games comprised of 6,326 game rounds. In total, 575 distinct puzzles were played and 3,418 agreements were reached.
  – The dataset is available at: http://hcomp.iis.sinica.edu.tw/dataset/
Evaluation – Puzzle Diversity

It is more difficult to reach the \((i+1)\text{-th}\) agreement than the \(i\text{-th}\) agreement.
Evaluation – Puzzle Diversity

• The differences exist among the puzzles.

It is more difficult to reach the \((i+1)\)-th agreement than the \(i\)-th agreement.
Evaluation – Puzzle Diversity

• The differences exist among the puzzles.

• It is more difficult to reach the \((i+1)\)-th agreement than the \(i\)-th agreement.

• It is important to consider puzzle diversity!
Simulation Results (1)

APSA scheme is superior in terms of reducing the number of the passed rounds.

\( N_T \): # of distinct puzzles with at least one agreement reached
\( N_P \): # of distinct puzzles played
Simulation Results (2)

APSA scheme yields more agreements with better per-puzzle throughput.

APSA scheme can better accommodate puzzle diversity than the OPSA scheme.

L.-J. Chen, Exploiting Puzzle Diversity in Puzzle Selection for ESP-like GWAP Systems, WI 2010, Toronto, Canada
• APSA always achieves a better system gain than the OPSA scheme (about 5% improvement).
  – The system gain could be improved further by modifying the second part of the metric (e.g., by introducing competition into the system [17]).
Summary

• We identify the *puzzle diversity* issue in ESP-like GWAP systems.

• We propose the Adaptive Puzzle Selection Algorithm (APSA) to consider *individual differences* by promoting *equality of opportunity*.

• We design a data structure, called *Weight Sum Tree* (WST) to reduce the computational complexity of APSA.

• We evaluate the APSA scheme and show that it is more
• **GWAP API** ([http://hcomp.iis.sinica.edu.tw/GWAP_API/](http://hcomp.iis.sinica.edu.tw/GWAP_API/))
  - JAVA-based API source codes released
  - ESP Lite: an example of GWAP API
  - ESP Lite dataset (v2010.01.01)

L.-J. Chen, Exploiting Puzzle Diversity in Puzzle Selection for ESP-like GWAP Systems, WI 2010, Toronto, Canada
Thank You!

Network Research Lab: http://nrl.iis.sinica.edu.tw/
GWAP API: http://hcomp.iis.sinica.edu.tw/GWAP_API/