Bringing Reputation-awareness into Crowdsourcing

Han Yu, Zhiqi Shen
School of Computer Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
Email: yuhan@pmail.ntu.edu.sg, zqshen@ntu.edu.sg

Cyril Leung
Department of Electrical and Computer Engineering
The University of British Columbia
Main Mall V6T 1Z4, Vancouver, BC, Canada
Email: cleung@ece.ubc.ca

***Abstract*—In crowdsourcing systems, worker selection is a challenging issue. Researchers have advocated the incorporation of reputation management into crowdsourcing systems to address this issue. However, current reputation-based decisions often result in concentrating the delegation of human intelligence tasks (HITs) to a small number of highly reputable workers to reduce risk. It conflicts with the main objective of crowdsourcing systems which is to promote mass collaboration. In this paper, we propose a situation-aware approach - SWORD - to enable existing reputation models to work in crowdsourcing systems. We holistically consider the objectives of all stake-holders in a crowdsourcing system (including requesters, workers, and system operators) to formulate the HIT allocation problem as a trade-off between quality and timeliness, and propose an efficient approach via constraint optimization to produce solutions to this problem with low computational complexity. Extensive simulations designed based on the actual conditions from Amazon's Mechanical Turk system demonstrates significant advantage of SWORD compared to existing approaches in improving social welfare.**

I. Introduction

In recent years, the philosophy of using computers is shifting. Instead of trying to push for further automation with ever more sophisticated computing systems to tackle problems that human beings find easy to do while computers struggle with (e.g., transcribing a video, completing a survey), a new breed of computing platforms that outsource them to a diverse group of people have emerged. This category of computing platforms is known as crowdsourcing systems. Some of the well-known examples include Amazon's Mechanical Turk (AMT), 99designs, and Mob4hire, etc [1].

Crowdsourcing systems are a new breed of e-commerce systems. Many crowdsourcing systems operate in a similar fashion as informal online job placement agencies. Each of them has a registered user base consisting of *requesters* who have projects that need to be outsourced, and *workers* who are willing to offer their time and effort to complete these tasks in exchange for monetary gain. A requester breaks down a project into small human intelligence tasks (HITs) using proprietary tools provided by the crowdsourcing systems, then publishes for workers take up. A monetary reward is associated to each HIT.

In such an open system, workers may have different levels of reliability or even act strategically when working on HITs due to ulterior motives. Reputation management has been shown to be a promising approach to address this problem [1]. Recent works studied how to elicit fair evaluations on the quality of HIT results [2] and how to minimize errors when updating workers' reputation [3]. However, there is one fundamental difference between crowdsourcing systems and system characteristics under which existing trust management models are proposed: existing trust models assume no limitation on the capacity of a trustee, while in crowdsourcing systems, trustees are human beings who are inherently capacity-constrained. This mismatch has significant implications on the applicability of existing reputation models in crowdsourcing. The success of an HIT depends on both the quality and the timeliness of its result. Uncoordinated interaction decisions made by existing reputation models have been found to result in over-utilization of a small number of highly reputable workers and reduced the business throughput of a crowdsourcing system, thus making reputation management unattractive [4].

To enable existing reputation models to work in crowdsourcing systems, we propose a social welfare optimizing reputation-aware decision-making (SWORD) HIT allocation approach to dynamically manage the workforce in a crowdsourcing system. It helps the system make efficient HIT allocation decisions, in view of the reputation of and current situation facing individual workers, to unleash their productive capacity by strategically adjusting the level of greediness in individual requester's HIT allocation decisions in real-time.

This paper makes the following contributions:

- We propose a novel constraint optimization based formulation that balances quality guarantee with mass collaboration to enable reputation management to be benefit crowdsourcing systems users.
- We design an efficient approach to make situation-aware HIT allocation decisions to produce solutions to the optimization problem in the proposed system model with *linearithmic* time and linear storage complexities.
- We prove the existence of theoretical performance guarantees for SWORD when its recommendations are fully complied (additional contribution compared to our previous work in [5]).
- We demonstrate that SWORD outperforms the existing approaches in a highly dynamic simulated environment which has been constructed based on actual system conditions in AMT.

II. Related Work

Studies on how to design incentive mechanisms in crowdsourcing systems to induce workers to behave cooperatively have been carried out [6]. In this paper, we study the issue without modifying existing crowdsourcing pricing models.

Reputation-aware task delegation decision-making approaches can be classified as either *greedy* or *dynamic* according to their strategies [7]. In a typical *greedy approach* (e.g., in [8]), a truster (a requester in our case) explores for trustees (workers in our case) with desired reputation through either relying on supporting infrastructure (e.g., peer recommendation, social network analysis, etc.) or random exploration. The reputations of the candidate trustees are evaluated using a trust model of choice, and tasks are delegated to the one with the highest reputation. Such an approach is currently the most widely adopted in computational trust literature [9], [7]. From an individual truster's point of view, in order to maximize its own long term wellbeing, it is locally rational to select the best possible option that can be found as often as possible.

Compared to the greedy approach, *dynamic approaches* are not as widely used by existing trust models. The dynamism in such approaches is related to how they decide when to delegate tasks to known reputable trustees and when to explore for potentially better alternatives. Such decisions are made often by measuring the changes in the behavior of known reputable trustees. It can be achieved through statistical methods as in [10], [11]. The dynamic approaches eventually also settle in the greedy strategy which is for each truster to always select the known trustee with the highest reputation to delegate tasks. In [12], the authors recognized the potential problem that can be caused by uncoordinated selfish task delegation decisions in a system where trustees have limited capacity. They proposed a Global Consideration (GC) approach based on heuristics for varying the ranking of trustees based on not only their reputation values but also their current workload.

III. Preliminaries

In this section, we propose improvements to the existing crowdsourcing system operation to allow worker reputation values and the overall system wellbeing to be considered simultaneously when allocating HITs to workers.

A crowdsourcing system consists of a set of requesters, R, and a set of workers, W. A requester r, $r \in R$, can propose a set of HITs (a.k.a. an HIT group), $H_r(t)$, and publish them in the common HIT queue, $Q(t)$, in the crowdsourcing system at any time step t. Usually, the general effort level required for completing an HIT h as well as its maximum possible payoff u are the same for all $h \in H_r(t)$. When a worker, $w \in W$, with the right qualification for the job who has noticed the newly published $H_r(t)$, w can accept one or more h from $H_r(t)$ into his pending HIT queue and work on them. In practical crowdsourcing systems such as AMT, this is currently accomplished on a *first-come-first-served* basis. As w's capacity is limited, he can only complete up to a maximum of μ_w^{max} HITs per unit time. μ_w^{max} can be regarded as a constant over a given duration. At each time step t, the actual number of HITs completed by a worker w is denoted as $\mu_w(t)$, where

$$\mu_w(t) \in \{0, ..., \mu_w^{max}\} \text{ for } \forall w \in W, \forall t \tag{1}$$

$$\mu_w(t) \leq q_w(t) \text{ for } \forall w \in W. \tag{2}$$

$q_w(t)$ is a variable representing the number of HITs in w's pending HIT queue at time step t. Constraint (1) ensures that only an integer number of no more than μ_w^{max} HITs can be completed by a w per unit time. Constraint (2) ensures that the number of HITs completed by w per unit time is less than or equal to the total number of HITs currently waiting for w to process. The exact value of $\mu_w(t)$ depends solely on the behavior of w.

The outcome from r delegating an HIT h to w at time step t is denoted by $O^h_{r\to w}(t)$. $O^h_{r\to w}(t)$ can take the value of either 1 (representing success) or 0 (representing failure). $O^h_{r\to w}(t) = 1$ only if the HIT result produced by w is of satisfactory quality for r and within the specified deadline. If $O^h_{r\to w}(t) = 0$, r can decide whether to put h into $Q(t)$ for allocation again or not. Typically, in practical crowdsroucing systems such as AMT, an HIT Group is specified by a requester to be active for a certain duration after being published. Any HIT results received within this duration are regarded as completed on time. r is only obliged to pay w if $O^h_{r\to w}(t) = 1$.

The trustworthiness of a worker w can be estimated through calculating his reputation, $\gamma_w(t)$, based on records of $O^h_{r\to w}(t)$ for all r who have delegated HITs to w in the past. There are a lot of trust models that can produce $\gamma_w(t)$ values within the range $[0, 1]$ where 0 means w is completely untrustworthy, and 1 means w is completely trustworthy [7]. The SWORD approach focuses on how task delegation decisions are made after the $\gamma_w(t)$ values are obtained.

IV. The *SWORD* Approach

The overall goal of a crowdsourcing system is to maximize the time averaged social welfare in the system which is the sum of utility derived from HITs of the same type:

$$\overline{U} \triangleq \frac{1}{T}\sum_{t=0}^{T-1}\sum_{h} u \cdot O^h_{r\to w}(t). \tag{3}$$

To achieve this goal, successful HIT allocation decisions need to minimize the potential *risk* for the requesters to get low quality HIT results and the overall *congestion* experienced by them. This objective can be stated as minimizing a $(risk+drift)$ expression. Assuming the underlying reputation evaluation model can produce reliable reputation values, the risk exposure for r to trust w can be gauged through w's reputation. The higher the reputation of w, the less risk r needs to take on by delegating to w. Therefore,

$$risk \triangleq a_w(t)[u \cdot (1 - \gamma_w(t))]. \tag{4}$$

$a_w(t)$ is the actual number of new HITs allocated to a worker w at time step t.

If mass collaboration is efficiently utilized, the waiting time experienced by requesters should be short in general. To minimize the *congestion*, two conditions should be avoided as far as permitted by the situation: 1) *queuing instability*: the sizes of the workers' pending HIT queues, $q_w(t)$, for all w should not be allowed to grow indefinitely at any time step t, and 2) *capacity under-utilization*: the workers' pending HIT queues should not be allowed to become too short subject to considerations about the workers' reputation and the available new HITs for allocation at t. In essence, the workload for each w should be maintained as close to a target value as allowed by the physical limitations in the environment. The target workload of w, θ_w, is positively correlated to w's innate characteristics represented by the 2-tuple $\langle \mu_w^{max}, \gamma_w^{max} \rangle$. γ_w^{max} is the maximum reputation value achieved by w over a given period, and it can be assumed to reflect the behavior of w. As the behavior pattern of w may change over time, the value of γ_w^{max} needs to be updated from time to time. We express θ_w as a combination of w's capacity and reputation:

$$\theta_w \triangleq \lfloor \mu_w^{max} + V \cdot \gamma_w^{max} \rfloor \tag{5}$$

where V is a non-negative control variable that determines the relative importance given to quality and timeliness. The potential waiting time in the system can be measured by the overall drift away from the target workloads. Large *drift* indicates inefficiency in HIT allocation. Thus, to minimize waiting time, the overall drift needs to be minimized.

The overall deviation from the target workload in a crowdsourcing system at any given time can be expressed as a *Lyapunov function* [13]:

$$L(t) \triangleq \frac{1}{2} \sum_{w \in W} (q_w(t) - \theta_w)^2. \tag{6}$$

Therefore, the overall drift can be expressed as a conditional *Lyapunov drift*:

$$\triangle(t) \triangleq \mathbb{E}\{L(t+1) - L(t) | \forall q_w(t), w \in W\}. \tag{7}$$

The drift in w's workload at the beginning of t depends on both the number of HITs completed by w during the previous time step $(t-1)$ and the number of new HITs to be allocated to w at t. Following the principles of *Lyapunov drift*, the *drift* is defined as:

$$drift \triangleq a_w(t)[q_w(t) - \theta_w] \tag{8}$$

where the queuing dynamics of $q_w(t)$ is:

$$q_w(t+1) \leftarrow \max[q_w(t) - \mu_w(t), 0] + a_w(t). \tag{9}$$

With expressions for both the *risk* (Eq. (4)) and the *drift* (Eq. (8)), the objective of minimizing $(risk + drift)$ can be expressed as

Minimize:

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{w \in W} \{V \cdot a_w(t)[u(1 - \gamma_w(t))] + a_w(t)[q_w(t) - \theta_w]\} \tag{10}$$

Subject to:

$$a_w(t) \in \{0, ..., \theta_w\} \tag{11}$$

$$a_w(t) \leq Q(t) \tag{12}$$

$$\gamma_w(t) \geq Th, \text{for all } w \text{ and } t \tag{13}$$

Constraint (11) ensures that $a_w(t)$ will only be of integer values not exceeding θ_w. Constraint (12) ensures that no more than the total number of HITs currently pending allocation in $Q(t)$ will be assigned to w. Constraint (13) ensures that only workers who have obtained at least the minimum reputation, Th, as preferred by the requesters will be considered.

By re-arranging the terms, we can re-write Eq. (10) as:

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{w \in W} a_w(t)\{u[1 - \gamma_w(t)]V + q_w(t) - \theta_w\} \tag{14}$$

Two distinct concepts are represented by Eq. (14): 1) $a_w(t)$, which is the HIT allocation decision for each $w \in W$, and 2) $\{u[1 - \gamma_w(t)]V + q_w(t) - \theta_w\}$, which represents the innate characteristics of w as well as the external situation facing w. We refer to the second concept as the *undesirability score* of w at time step t, and denote it as $d_w(t)$. The value of $d_w(t)$ varies for each w over time with changes in the behavior of w and the situation facing w.

The SWORD approach produces an HIT allocation plan, $P(t) = \{a_w(t) | \forall w \in W\}$, which specifies the number of HITs to be assigned to each worker in a crowdsourcing system at time step t. For $P(t)$ to minimize Eq. (10), SWORD follows a process as illustrated in Algorithm 1. The time complexity for SWORD to produce $P(t)$ is $O(|W|)$ with a single pass through the for loop. However, the computational complexity of the entire process also depends on the sorting algorithm used to rank the workers according to their undesirability scores. Assuming an efficient sorting algorithm such as *Quicksort* is employed, SWORD can achieve time complexity of $O(|W| log |W|)$ (*linearithmic* time) and storage complexity of $O(|W|)$. This makes SWORD computationally efficient even for large scale crowdsourcing systems.

Algorithm 1 SWORD

Require: μ_w^{max}, γ_w^{max} and $\gamma_w(t)$ values for all $w \in W$, the common HIT queue $Q(t)$.

```
Re-evaluate d_w(t) for ∀w ∈ W
Rank ∀w ∈ W in ascending order of their d_w(t) values
for each w ∈ W do
    if d_w(t) < 0 then
        a_w(t) = min[μ_w^max, Q(t)]
        Q(t) ← Q(t) − a_w(t)
    else
        a_w(t) = 0
    end if
end for
Return P(t)
```

V. Analysis

Suppose $q_w(t) > \theta_w$ for a worker w at a particular time step t, then $q_w(t) - \theta_w > 0$. Since $V > 0$, $u \in \mathbb{R}^+$, and $0 \leq \gamma_w(t) \leq 1$, the expression $\{u[1-\gamma_w(t)]V + q_w(t) - \theta_w\}$ is strictly positive. By choosing $a_w(t) = 0$, we can obtain a strictly smaller value for Eq. (10) than choosing any other $a_w(t) > 0$. Since, at each time step t, only up to μ_w^{max} new HITs can be assigned to w, and if $q_w(t) > \theta_w$, $a_w(t) = 0$, it can be deduced that $q_w(t) \leq \theta_w + \mu_w^{max}$ for all t if $q_w(0) \leq \theta_w + \mu_w^{max}$. Thus, we have $q_w(t) \leq \lfloor 2\mu_w^{max} + V \cdot \gamma_w^{max} \rfloor$. This proves that $P(t)$ produced by SWORD guarantees an *upper bound* to the sizes of the workers' pending HIT queues at any time step t.

From Eq. (9), we have:

$$
\begin{aligned}
&(q_w(t+1) - \theta_w)^2 \\
&\leq (q_w(t) - \theta_w)^2 - 2(q_w(t) - \theta_w)(\mu_w(t) - a_w(t)) \\
&+ (\mu_w(t) - a_w(t))^2.
\end{aligned}
$$

By re-arranging the above inequality and dividing both sides by 2, we have:

$$
\begin{aligned}
&\frac{1}{2}(q_w(t+1) - \theta_w)^2 - \frac{1}{2}(q_w(t) - \theta_w)^2 \\
&\leq \frac{1}{2}(\mu_w(t) - a_w(t))^2 - (q_w(t) - \theta_w)(\mu_w(t) - a_w(t)).
\end{aligned}
$$

By substituting Eq. (6) into the above inequality, summing over $\forall w \in W$, and taking conditional expectation on both sides, we have:

$$
\begin{aligned}
\triangle(t) &\leq \frac{1}{2}\sum_{w \in W} \mathbb{E}\{(\mu_w(t) - a_w(t))^2 | \forall q_w(t)\} \\
&- \sum_{w \in W} (q_w(t) - \theta_w)\mathbb{E}\{\mu_w(t) - a_w(t) | \forall q_w(t)\} \\
&\leq B - \sum_{w \in W} (q_w(t) - \theta_w)\mathbb{E}\{\mu_w(t) - a_w(t) | \forall q_w(t)\}
\end{aligned}
$$

where B is a finite constant that satisfies:

$$
\frac{1}{2}\sum_{w \in W} \mathbb{E}\{(\mu_w(t) - a_w(t))^2 | \forall q_w(t)\} \leq B \leq \frac{1}{2}\sum_{w \in W} (\mu_w^{max})^2.
$$

This proves that $P(t)$ produced by SWORD guarantees an *upper bound* on the drift away from the target workload at any time step t when there are enough new HITs in $Q(t)$.

VI. Evaluation

In order to evaluate the performance of SWORD, we implemented a simulated crowdsourcing environment based on past studies on the system characteristics of the popular AMT crowdsourcing system. The objective of our experiments is to compare the performance of SWORD with existing approaches under different operating environment conditions.

A. Experiment Design

In the test-bed, requester agents put up HITs with associated rewards for worker agents to complete. Worker agents work on HITs in order to earn artificial rewards from the requester agents. These agents are designed to simulate the limitations of human workers. The worker agents are designed with the *quality-quantity tradeoff* (i.e., a worker who produces high quality HIT results can only do so for a small number of HITs per unit time and vice versa). According to studies conducted by [14] on the AMT system from 2009 to 2010, the ratio between the worker and the requester populations is about 20:1 and the average HITs in an HIT Group is about 40. We use these key statistics to design our experiments so as to be as close to real crowdsourcing systems as possible.

Six simulated crowdsourcing systems, each equipped with different HIT allocation approaches, are compared under worker agent population configurations from highly unproductive to highly productive in the proposed test-bed environment. They are: AMT, in which requester agents who follow AMT's *first-come-first-served* model for HIT allocation; BRS2002 (which is a *greedy approach*), M2009, H2010, and GC (which are *dynamic approaches*) in which HITs are allocated following [8], [11], [10], [12] respectively; and SWORD in which the HITs proposed by the requester agents are distributed to the worker agents following the SWORD approach. The value of the control variable V is varied from 0 to 100 to analyze the perofrmance sensitivity of SWORD to it.

B. Results and Discussions

Figure 1 illustrates the performance of the six approaches in terms of $\overline{U}$, which is calculated according to Eq. (3). The $\overline{U}$ value for each approach is averaged under different worker population configurations (from 10% trustworthy workers to 100% trustworthy workers in the simulated worker populations). The performance landscape further divides $\overline{U}$ into two component factors: 1) success rate, s, which represents the percentage of all HITs that has been completed with acceptable quality on time, and 2) business volume, b, which denotes the total number of HIT groups processed by a population of worker agents per time step. The performance landscape is divided into four quadrants: $Q1$ (low s, low b), $Q2$ (high s, low b), $Q3$ (low s, high b), and $Q4$ (high s, high b). With no concerns for reputation, AMT can achieve high b. However, it offers no guarantee on s. Thus, AMT occupies $Q3$. On the other hand, BRS2002, M2009 and H2010 primarily focuses on improving the average quality of the HITs through allocating more of them to reputable workers. The three of them occupy $Q2$. GC attempts to balance s and b by adjusting the workers' reputation values partially according to their current workload. However, the approach taken is not effective under the conditions where workers' behaviors follow the *quality-quantity tradeoff*. Thus, GC occupies $Q1$. SWORD (with V=10) achieves the highest $\overline{U}$ through effective in balancing quality with business volume among the six approaches. It occupies $Q4$ which is the most desirable quadrant.

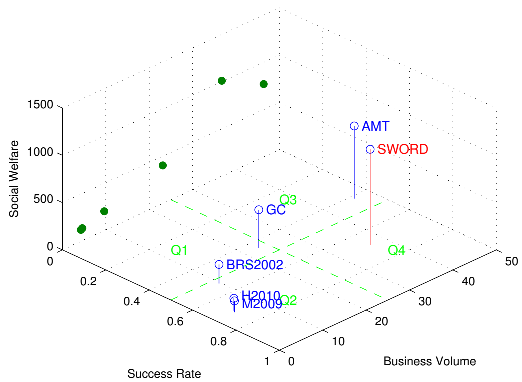

Fig. 1. Summary of performance of various approaches.

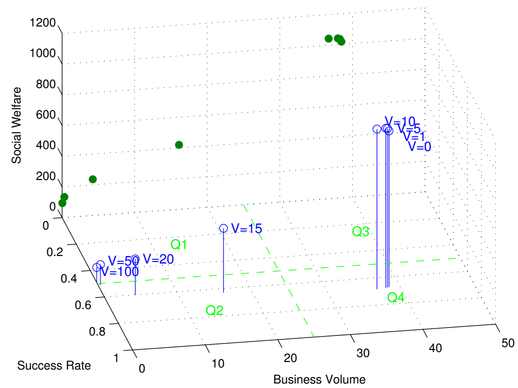

Fig. 2. Sensitivity analysis for SWORD.

The control variable V allows the system administrator to determine how much s he is willing to sacrifice to gain more b and vice versa. The larger the value of V, the more preference in given to s. As shown in Figure 2, increasing V from 0 to 10 improves $\overline{U}$ while still ensuring that SWORD occupies $Q4$. However, further increasing V results in over concentration of HITs to highly reputable workers and pushes SWORD into $Q2$ while significantly decreasing $\overline{U}$. Thus, increasing V when its value is small can improve social welfare. However, its value should not be increased to such an extent as to exceed the physical limitations of the workers.

VII. Conclusions

In this paper, we propose a situation and reputation aware HIT allocation approach - SWORD - to support efficient HIT distribution in crowdsourcing systems. It continually adjusts the distribution of HITs to workers using the difference between the target workload and the current workload of the workers to guide its decisions. It improves the social welfare of a given crowdsourcing system by striking a balance between maintaining the quality of the HIT results and promoting mass collaboration to shorten waiting time. Empirical evaluations show that SWORD is able to achieve comparable success rate to existing reputation-based approaches under various system conditions while significantly increase the volume of business. SWORD enables crowdsourcing systems to reap the benefit of reputation management without sacrificing mass collaboration, thereby enhancing the long term sustainability of these systems.

Acknowledgment

This research is supported in part by Interactive and Digital Media Programme Office (IDMPO), National Research Foundation (NRF) hosted at Media Development Authority (MDA) of Singapore (Grant No.: MDA/IDM/2012/8/8-2 VOL 01).

References

[1] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[2] M. Allahbakhsh, A. Ignjatovic, B. Benatallah, S.-M.-R. Beheshti, E. Bertino, and N. Foo, "Reputation management in crowdsourcing systems," in *Proceedings of the 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'12)*, 2012, pp. 664–671.

[3] Y. Xiao, Y. Zhang, and M. van der Schaar, "Socially-optimal design of crowdsourcing platforms with reputation update errors," in *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP'13)*, 2013, pp. 5263–5267.

[4] H. Yu, Z. Shen, C. Miao, and B. An, "Challenges and opportunities for trust management in crowdsourcing," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'12)*, 2012, pp. 486–493.

[5] ——, "A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems," in *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'13)*, 2013, pp. 1315–1316.

[6] Y. Zhang and M. v. d. Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12)*, 2012, pp. 2140–2148.

[7] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, "A survey of multi-agent trust management systems," *IEEE Access*, vol. 1, no. 1, pp. 35–50, 2013.

[8] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of 15th Bled Electronic Commerce Conference*, 2002, pp. 41–55.

[9] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1755–1772, 2010.

[10] M. Hoogendoorn, S. W. Jaffry, and J. Treur, "Exploration and exploitation in adaptive trust-based decision making in dynamic environments," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (IAT'10)*, 2010, pp. 256–260.

[11] V. Muñoz, J. Murillo, B. López, and D. Busquets, "Strategies for exploiting trust models in competitive multi-agent systems," in *Proceedings of the 7th German Conference on Multiagent System Technologies (MATES'09)*, 2009, pp. 79–90.

[12] A. Grubshtein, N. Gal-Oz, T. Grinshpoun, A. Meisels, and R. Zivan, "Manipulating recommendation lists by global considerations," in *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART'10)*, 2010, pp. 135–142.

[13] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[14] P. G. Ipeirotis, "Analyzing the amazon mechanical turk marketplace," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, pp. 16–21, 2010.