

Efficient Task Sub-delegation for Crowdsourcing

Han Yu¹, Chunyan Miao¹, Zhiqi Shen¹, Cyril Leung^{1,2}, Yiqiang Chen³, Qiang Yang⁴

¹School of Computer Engineering, Nanyang Technological University, Singapore

²Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC, Canada

³Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

⁴Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong

Abstract

Reputation-based approaches allow a crowdsourcing system to identify reliable workers to whom tasks can be delegated. In crowdsourcing systems that can be modeled as multi-agent trust networks consist of resource constrained trustee agents (i.e., workers), workers may need to further sub-delegate tasks to others if they determine that they cannot complete all pending tasks before the stipulated deadlines. Existing reputation-based decision-making models cannot help workers decide when and to whom to sub-delegate tasks. In this paper, we proposed a reputation aware task sub-delegation (RTS) approach to bridge this gap. By jointly considering a worker's reputation, workload, the price of its effort and its trust relationships with others, RTS can be implemented as an intelligent agent to help workers make sub-delegation decisions in a distributed manner. The resulting task allocation maximizes social welfare through efficient utilization of the collective capacity of a crowd, and provides provable performance guarantees. Experimental comparisons with state-of-the-art approaches based on the *Epinions* trust network demonstrate significant advantages of RTS under high workload conditions.

Introduction

Existing crowdsourcing systems can be modeled as multi-agent systems (MASs) where tasks need to be delegated to workers (i.e., agents). Currently, much research effort has been devoted into computing the agents' reputation and acting based on different levels of trust among agents (Jøsang, Ismail, and Boyd 2007; Yu et al. 2010; 2013c). Commonly used techniques for evaluating agents' reputation include fuzzy neural approaches (Miao et al. 2002; Song et al. 2009; Man et al. 2011) and inference-based approaches (Miao et al. 2001; Li et al. 2009). These approaches have been shown to reduce the potential harm caused by untrustworthy agents in various fields (Pan et al. 2009; Yu et al. 2011). In practice, crowdsourcing workers are resource constrained in terms of having limited productive capacity (Yu et al. 2012). To address this issue, a number of reputation-based task allocation approaches have been proposed to balance the workload among trustee agents with

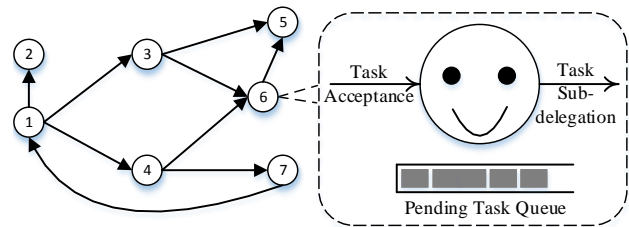


Figure 1: A trust network of resource constrained agents.

similar reputation (Grubshtein et al. 2010; Yu et al. 2013a; 2013b; 2014).

These existing approaches all assume that agents' reputation information is commonly known to and agreed by all agents in an MAS. However, in MASs where agents have different views about the trustworthiness of other agents, and form trust networks where only the reputation information of a limited number of agents is known to an agent (Massa and Avesani 2005), this assumption is no longer valid. An example of a trust network formed by resource constrained agents is shown in Figure 1. The nodes in the network represent agents (which can initiate new tasks and work on task requests from other agents) while the arrows represent trust relationships. Each agent can only spend limited effort in any given period of time to work on the tasks in its pending task queue. In this type of MAS environment, being able to sub-delegate tasks becomes crucial to the efficient utilization of the collective productivity of the MAS (Liu, Wang, and Orgun 2011; Burnett and Oren 2012). Each agent needs to make its own decisions on 1) *task acceptance*: how much new workload should be accepted and 2) *task sub-delegation*: how much pending workload should be sub-delegated based on the assessment of its current situation. Existing works only addressed the first challenge (Grubshtein et al. 2010; Yu et al. 2013a; 2013b; 2014). The second challenge remains open.

In practice, task sub-delegation is a useful technique for distributing workload in trust networks. For example, sub-delegation often occurs in hierarchical crowdsourcing (Pickard et al. 2011). In this paper, we propose a distributed reputation-aware task sub-delegation (RTS) approach to address both challenges simultaneously. By jointly considering the reputation and workload of an individual agent and

its trust relationships with other agents, RTS helps the agent with task sub-delegation decisions so as to minimize the agent’s cost associated with task sub-delegation while maximizing its time averaged reward in the long term.

To the best of our knowledge, RTS is the first approach to help an individual trustee agent make situation-aware task sub-delegation decisions for crowdsourcing systems which can be modeled as multi-agent trust networks. The resulting task allocation maximizes social welfare through efficient utilization of the collective productivity of the MAS, and provides provable performance bounds. RTS has been compared to four state-of-the-art approaches through extensive simulations based on the *Epinions* trust network dataset. The results demonstrate RTS achieves over 30% higher social welfare than the best performing existing approach under high workload conditions.

Related Work

Existing research works in the area of multi-agent trust focus heavily on improving the accuracy of evaluating agents’ reputation (Yu et al. 2013c). The question of how to utilize the reputation information to make efficient task delegation decisions among agents is has been less well studied area. In (Burnett, Norman, and Sycara 2011), a decision theoretic framework is proposed to help agents determine whom to delegate tasks to when reputation information is lacking. By employing explicit incentives, monitoring, and reputation incentives, an agent constructs a framework similar to a business contract to minimize its risk exposure when interacting with less well known agents. However, this framework is based on the traditional agent performance model in which agents’ limitations in terms of productivity (e.g., effort output rate) is not explicitly considered.

Recent works have proposed new agent performance models which explicitly consider limited productivity. In (Grubshtein et al. 2010), a Global Considerations (GC) approach to adjust the trustee agents reputation values based on their current workload levels is proposed. Under GC, the probability of task being delegated to a trustee agent is directly proportional to the agent’s reputation standing among its peers. This probability is further adjusted by the ratio between the agent’s effort output rate and the number of newly accepted tasks whenever this ratio falls below 1.0 (assuming each task requires unit effort).

In (Yu et al. 2013c), the centralized task allocation approach - SWORD - is proposed to dynamically balance the workload among trustee agents with similar trustworthiness in an MAS. A fully distributed variant of SWORD - the DRAFT approach - which helps trustee agents determine which incoming tasks should be accepted is proposed in (Yu et al. 2013a). In (Yu et al. 2014), the DRAFT approach is further extended to operate in environments where the trustee agents’ utility functions are not linearly related to the expected income that can be derived from completing tasks.

Nevertheless, these existing approaches suffer from two major limitations: 1) they assume each agent knows the reputation of all other agents (i.e. the trust network among the agents is fully connected), and 2) they assume that when

a task is assigned to an agent, it cannot be further sub-delegated. In practical multi-agent trust networks, these assumptions are not valid (Massa and Avesani 2005). In contrast, the RTS approach proposed in this paper is not constrained by these two limiting assumptions.

Preliminaries

In this paper, we consider the problem of delegating and further sub-delegating a task τ_j which is proposed by a *truster agent* j through multiple *trustee agents* forming a *delegation chain*. Each task may require different amount of effort. We assume that the workload of a task can be measured in effort units required. Each task is associated with a stipulated deadline before which it must be completed. A trustee agent i has a limited effort output rate of μ_i^{max} . Tasks accepted by i is stored in its pending task queue and are serviced on a *first-come-first-served* basis. In this work, we do not consider the cases in which an agent can reorder the tasks based on some priority criteria. $q_i(t)$ denotes the total workload (i.e. total number of effort units) in i ’s pending task queue at any given time step t . The queuing dynamics of $q_i(t)$ is expressed as:

$$q_i(t + 1) = \max[0, q_i(t) + \lambda_i(t) - \mu_i(t) - s_i(t)] \quad (1)$$

where $\lambda_i(t)$ denotes the amount of new workload accepted into $q_i(t)$ at t , $\mu_i(t)$ represents the actual amount of workload completed by i during t , and $s_i(t)$ denotes the amount of workload sub-delegated to other agents during t . As these variables are local to an agent, their values can be obtained through some monitoring mechanisms such as in (Heymann and Garcia-Molina 2011).

Based on past interactions, an agent can construct a set of known trustee agents, n_i^{sub} , to whom tasks may be delegated/sub-delegated. The past performance of an agent i in terms of the quality and timeliness in completing tasks can be used to evaluate i ’s reputation value, $r_i(t) \in (0, 1)$, using an existing reputation model such as (Jøsang and Ismail 2002). However, as tasks may be repeatedly sub-delegated through a delegation chain, all agents in the delegation chain should share responsibility for the final outcome of a task. In this paper, we adopt the *Decreasing Weighting* (DW) reputation update mechanism in (Burnett and Oren 2012) which has been shown to be a suitable strategy in this context. Under DW, decreasing proportions of weight are applied to consecutive agents in a delegation chain in the reverse order with the last agent in the chain, who actually worked on the task, assigned the highest weight. These weight values are then used to apportion the amount of change in each agent’s reputation based on the outcome of the task.

The price an agent charges for its service is denoted by $p_i(t)$. $p_i(t)$ is advertised by agent i for other agents to know and can be adjusted by agent i from time to time. In this paper, $p_i(t)$ is measured in “dollars per effort unit”. The reward an agent i derives from performing $\mu_i(t)$ effort units worth of tasks during time step t can be expressed as:

$$\xi_i(t) = \sum_{\tau_j} u(\tau_j) \quad (2)$$

where τ_j is a task in agent i 's pending task queue. $u(\tau_j) = \tau_j p_i(t)$ if τ_j is completed on time with quality acceptable to the truster agent j . Otherwise, $u(\tau_j) = 0$.

Given that a large number of trustees will be involved in interactions over a potentially infinite time horizon in an uncertain environment, it is intractable (even impossible) to compute the optimal (equilibrium) strategy for all the trustees. Alternatively, this paper optimizes the well-being of each trustee indirectly by maximizing the social welfare of all trustees. In an MAS with N agents, the *social welfare* (SW) is defined as the time-averaged sum of individual agent rewards:

$$SW = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^N \xi_i(t). \quad (3)$$

The RTS Approach

The RTS approach is designed for each individual trustee agent to use and takes only local knowledge as input. It helps an agent make the following decisions regarding workload management at every time step: 1) when and to whom to sub-delegate pending workload, and 2) the amount of new workload to accept into the pending task queue.

Sub-delegating Pending Workload

Intuitively, if an agent i determines it may not be able to complete all pending tasks before their stipulated deadlines, it should sub-delegate some of them to other agents it trusts. In this section, we illustrate how such an intuition can be translated into an actionable task sub-delegation decision approach.

In order to measure the need for i to sub-delegate its pending workload, we propose a conceptual queue $Q_i(t)$ which is updated together with $q_i(t)$ as follows:

$$Q_i(t+1) = \max[Q_i(t) - \mu_i(t) - s_i(t) + \bar{\lambda}_i 1_{[q_i(t) > 0]}, 0] \quad (4)$$

where $\bar{\lambda}_i = \frac{1}{T} \sum_{t=0}^{T-1} \lambda_i(t)$ is the time averaged new workload (in effort units) accepted by i in the past. $1_{[\text{condition}]}$ is a function that evaluates to 1 if [condition] is true, and 0 otherwise. The workload in $Q_i(t)$ is reduced by the same task servicing process $[-\mu_i(t) - s_i(t)]$ as in $q_i(t)$. However, the workload increasing process in $Q_i(t)$ is such that if the actual workload leftover in $q_i(t)$ is none zero at the time when $Q_i(t)$ is updated, $Q_i(t)$ increases by $\bar{\lambda}_i$. This results in the constant growth of $Q_i(t)$ if there are tasks in the physical pending task queue $q_i(t)$ which have not been serviced for an extended period of time. If we can find a way to ensure that both $q_i(t)$ and $Q_i(t)$ have finite upper bounds, the collective productivity of the MAS can be efficiently utilized.

Let $X_i(t) = (q_i(t), Q_i(t))$ be a concatenated vector of i 's physical and virtual pending task queues. The *Lyapunov* function (Neely 2010), which measures the level of congestion in both $q_i(t)$ and $Q_i(t)$, can be expressed as $L(X_i(t)) = \frac{1}{2}[q_i^2(t) + Q_i^2(t)]$. The conditional *Lyapunov* drift, which measures how drastically i 's pending workload changes, can

then be expressed as:

$$\begin{aligned} \Delta(X_i(t)) &= \mathbb{E}\{L(X_i(t+1)) - L(X_i(t)) | X_i(t)\} \\ &= \mathbb{E}\left\{\frac{1}{2}[q_i^2(t+1) - q_i^2(t) + Q_i^2(t+1) - Q_i^2(t)] | X_i(t)\right\}. \end{aligned} \quad (5)$$

Based on Eq. (4), we have:

$$\begin{aligned} \frac{1}{2}Q_i^2(t+1) - \frac{1}{2}Q_i^2(t) &\leq \frac{1}{2}[\bar{\lambda}_i - (\mu_i(t) + s_i(t))]^2 \\ &\quad + Q_i(t)[\bar{\lambda}_i - \mu_i(t) - s_i(t)] \\ &\leq \frac{1}{2}[(\lambda_i^{max})^2 + (\mu_i^{max} + s_i^{max})^2] \\ &\quad + Q_i(t)[\bar{\lambda}_i - \mu_i(t) - s_i(t)] \end{aligned} \quad (6)$$

where λ_i^{max} and s_i^{max} are the physical upper limits for $\lambda_i(t)$ and $s_i(t)$, respectively. Similarly, based on Eq. (1), we have:

$$\begin{aligned} \frac{1}{2}q_i^2(t+1) - \frac{1}{2}q_i^2(t) &\leq \frac{1}{2}[\lambda_i(t) - (\mu_i(t) + s_i(t))]^2 \\ &\quad + q_i(t)[\lambda_i(t) - \mu_i(t) - s_i(t)] \\ &\leq \frac{1}{2}[(\lambda_i^{max})^2 + (\mu_i^{max} + s_i^{max})^2] \\ &\quad + q_i(t)[\lambda_i(t) - \mu_i(t) - s_i(t)]. \end{aligned} \quad (7)$$

From Eq. (6) and Eq. (7), Eq. (5) can be expressed as:

$$\Delta(X_i(t)) \leq \mathbb{E}\{C_i + Q_i(t)[\bar{\lambda}_i - \mu_i(t) - s_i(t)] + q_i(t)[\lambda_i(t) - \mu_i(t) - s_i(t)] | X_i(t)\} \quad (8)$$

where $C_i = (\lambda_i^{max})^2 + (\mu_i^{max} + s_i^{max})^2$.

From agent i 's perspective, apart from minimizing drastic workload changes in its pending task queue, it also needs to minimize the cost incurred when sub-delegating tasks to other agents. These two objectives can be captured by the *drift-plus-cost* expression as follows:

$$\begin{aligned} \Delta(X_i(t)) + \rho_i \mathbb{E}\{\varphi_i(t) s_i(t) | X_i(t)\} &\leq \\ C_i + Q_i(t) \mathbb{E}\{\bar{\lambda}_i - \mu_i(t) - s_i(t) | X_i(t)\} &+ \\ q_i(t) \mathbb{E}\{\lambda_i(t) - \mu_i(t) - s_i(t) | X_i(t)\} &+ \\ \rho_i \mathbb{E}\{\varphi_i(t) s_i(t) | X_i(t)\}. & \end{aligned} \quad (9)$$

$\varphi_i(t) = \frac{1}{|n_i^{sub}|} \sum_{j=1}^{n_i^{sub}} p_j(t)$ is the average price charged by trustee agents known to agent i 's at time step t . The term $\rho_i > 0$ indicates agent i 's eagerness to work. The larger the ρ_i value, the more eager is agent i to work. RTS observes the current states of the physical and conceptual queues, as well as the current context tuple $\langle \mu_i(t), \lambda_i(t), \varphi_i(t) \rangle$, to determine the value of $s_i(t)$ so as to minimize the *drift-plus-cost* expression at every time step. By isolating the terms containing the decision variable $s_i(t)$ on the right hand side of Eq. (9), the objective function for agent i can be expressed as:

Minimize:

$$s_i(t)[\rho_i \varphi_i(t) - q_i(t) - Q_i(t)] \quad (10)$$

subject to:

$$0 \leq s_i(t) \leq q_i(t) \quad (11)$$

$$n_i^{sub} \neq \{\emptyset\} \quad (12)$$

$$\exists k \in n_i^{sub} \wedge \tau_j \in q_i(t) : r_k(t) \geq Th \wedge p_k(t) \leq p_{\tau_j} \quad (13)$$

where $Th \in [0, 1]$ is a predefined reputation threshold value. The threshold-based reputation sanctioning technique is widely used in multi-agent trust research and practical systems (Yu et al. 2013b). For example, in Amazon’s Mechanical Turk (mTurk)¹ crowdsourcing system, a task requester can specify the minimal “approval rate” a worker must achieve (which is similar to the worker’s reputation value) in order for him/her to be eligible to work on this requesters’ tasks. The term p_{τ_j} is the per-unit effort price agreed by agent i with the agent who delegated task τ_j to i when i accepted the task. In order to minimize Eq. (10), $\hat{s}_i(t)$, the ideal value of $s_i(t)$, can be expressed as:

$$\hat{s}_i(t) = \begin{cases} 0, & \text{if } \rho_i \varphi_i(t) - q_i(t) - Q_i(t) \geq 0 \\ q_i(t) - \mu_i(t), & \text{otherwise.} \end{cases} \quad (14)$$

Nevertheless, the actual $s_i(t)$ value still depends on Constraints (12) and (13) being satisfied. Constraint (12) requires that the list of known trustee agents to i must not be empty so that there are candidate agents to whom tasks may be sub-delegated. Constraint (13) requires that there exists at least one trustee agent k in n_i^{sub} with a reputation value higher than a pre-defined threshold and who can complete the task that is being sub-delegated at a cost no higher than what this task currently costs. Therefore, $s_i(t) \leq \hat{s}_i(t)$.

Eq. (14) can be interpreted as the following task sub-delegation policy: “the less *eager to work* an agent i is (indicated by small ρ_i values), the heavier its *current workload*, the longer the tasks in its task queue have been *pending*, and the lower the expected *cost* for sub-delegation, the more pending tasks agent i should sub-delegate”. Such a policy is rational and provides actionable guidance for an agent to determine the amount of work it should sub-delegate.

Accepting New Workload

Taking task sub-delegation into account, the expected profit for a trustee agent i at time step t can be expressed as $\mathbb{E}\{\xi_i(t)|p_i(t), r_i(t)\} = \lambda_i(t)p_i(t)r_i(t) - \varphi_i(t)s_i(t)$ where $\lambda_i(t)$ is a decision variable which controls the amount of new workload agent i can accept at t . Similar to Eq. (9), we can write a *profit-minus-drift* function as:

$$\begin{aligned} \rho_i \mathbb{E}\{\xi_i(t)|X_i(t)\} - \Delta(X_i(t)) = & \\ \rho_i \mathbb{E}\{\lambda_i(t)p_i(t)r_i(t) - \varphi_i(t)s_i(t)|X_i(t)\} & \\ - q_i(t)\mathbb{E}\{\lambda_i(t) - \mu_i(t) - s_i(t)|X_i(t)\} & \\ - C_i - Q_i(t)\mathbb{E}\{\bar{\lambda}_i - \mu_i(t) - s_i(t)|X_i(t)\} & \end{aligned} \quad (15)$$

which is to be maximized. By isolating the terms containing the decision variable $\lambda_i(t)$ on the right hand side of Eq. (15), we have:

Maximize:

$$\lambda_i(t)[\rho_i p_i(t)r_i(t) - q_i(t)] \quad (16)$$

¹<https://www.mturk.com/mturk/welcome>

Algorithm 1 The RTS Approach

Require: $q_i(t), Q_i(t), r_i(t), \rho_i, \varphi_i(t)$.

- 1: **if** $q_i(t) - \rho_i p_i(t)r_i(t) \leq 0$ **then**
 - 2: Accept up to a maximum of μ_i^{max} effort units’ worth of new tasks
 - 3: **else**
 - 4: Do not accept new tasks
 - 5: **end if**
 - 6: **if** $\rho_i \varphi_i(t) - q_i(t) - Q_i(t) \geq 0$ **then**
 - 7: $s_i(t) = 0$
 - 8: **else**
 - 9: $s_i(t) = q_i(t) - \mu_i(t)$
 - 10: **end if**
 - 11: Sub-delegate $s_i(t)$ effort units of tasks to i ’s network of trusted agents subject to Constraints (12) and (13)
 - 12: Update $q_i(t+1)$ according to Eq. (1)
 - 13: Update $Q_i(t+1)$ according to Eq. (4)
-

Subject to:

$$\lambda_i(t) \leq A_i(t) \quad (17)$$

where $A_i(t)$ is the amount of new workload other agents attempt to delegate/sub-delegate to agent i at time step t .

In order to maximize Eq. (16), whenever $\rho_i p_i(t)r_i(t) - q_i(t) > 0$, i should accept as much new workload as possible. In this paper, we adopt a conservative strategy. Thus, when $\rho_i p_i(t)r_i(t) - q_i(t) > 0$, i accepts up to μ_i^{max} amount of new workload into its pending task queue (i.e. $\lambda_i(t) = \min[A_i(t), \mu_i^{max}]$). When $\rho_i p_i(t)r_i(t) - q_i(t) \leq 0$, i should not accept new workload.

This part of RTS corresponds to the following task acceptance policy: “the more *eager to work* an agent i is (indicated by large ρ_i values), the lighter its *current workload*, and the higher the expected *reward* for completing new tasks, the more new tasks should agent i accept”. The RTS approach is illustrated in Algorithm 1.

Analysis

In this section, we analyze the performance bounds of RTS. Let $\lambda_i^*(t)$ and $s_i^*(t)$ be the optimal values for the decision variables $\lambda_i(t)$ and $s_i(t)$, respectively. Eq. (15) becomes:

$$\begin{aligned} \rho_i \mathbb{E}\{\xi_i(t)|X_i(t)\} - \Delta(X_i(t)) \leq & \\ \rho_i \mathbb{E}\{\lambda_i^*(t)p_i(t)r_i(t) - \varphi_i(t)s_i^*(t)|X_i(t)\} & \\ - q_i(t)\mathbb{E}\{\lambda_i^*(t) - \mu_i(t) - s_i^*(t)|X_i(t)\} & \\ - C_i - Q_i(t)\mathbb{E}\{\bar{\lambda}_i - \mu_i(t) - s_i^*(t)|X_i(t)\} & \end{aligned} \quad (18)$$

As $\mathbb{E}\{\lambda_i^*(t)p_i(t)r_i(t) - \varphi_i(t)s_i^*(t)|X_i(t)\}$ represents the expected profit i receives at time step t following the optimal policy, we use $\xi_i^*(t)$ to denote this component of Eq. (18). $\mathbb{E}\{\lambda_i^*(t)p_i(t)r_i(t)\}$ denotes the optimal total effort of tasks accepted by i at time step t . Since the optimal policy results in queue stability (i.e., the total effort of accepted tasks is less than or equal to i ’s task processing capacity and sub-delegation capacity), $\mathbb{E}\{\lambda_i^*(t) - \mu_i(t) - s_i^*(t)|X_i(t)\} \leq 0$. Similarly, $\mathbb{E}\{\bar{\lambda}_i - \mu_i(t) - s_i^*(t)|X_i(t)\} \leq 0$. By inserting these results into Eq. (18), we have:

$$\Delta(X_i(t)) - \rho_i \mathbb{E}\{\xi_i(t)|X_i(t)\} \leq C_i - \rho_i \xi_i^*(t). \quad (19)$$

Summing both sides of this inequality over all i and $t \in \{0, 1, \dots, T-1\}$ and taking the time average value yields:

$$\begin{aligned} & \frac{1}{T} \sum_{t=0}^{T-1} \sum_i \mathbb{E}\{L(X_i(t+1)) - L(X_i(t)) | X_i(t)\} \\ & - \frac{\rho}{T} \sum_{t=0}^{T-1} \sum_i \mathbb{E}\{\xi_i(t) | X_i(t)\} \leq C - \frac{\rho}{T} \sum_{t=0}^{T-1} \sum_i \xi_i^*(t) \end{aligned} \quad (20)$$

where $C = \frac{1}{T} \sum_{t=0}^{T-1} \sum_i C_i$, and $\rho = \frac{1}{T} \sum_{t=0}^{T-1} \sum_i \rho_i$. ρ can be interpreted as the time averaged collective willingness of agents in a given MAS to work on tasks. For any given queue state $X_i(t)$, there exists an optimal decision policy $\langle \lambda_i^*(t), s_i^*(t) \rangle$, the conditional expectations in Eq. (20) are equivalent to unconditional expectations. Thus:

$$\begin{aligned} & \frac{1}{T} \sum_i [\mathbb{E}\{L(X_i(T))\} - \mathbb{E}\{L(X_i(0))\}] \\ & - \frac{\rho}{T} \sum_{t=0}^{T-1} \sum_i \mathbb{E}\{\xi_i(t) | X_i(t)\} \leq C - \frac{\rho}{T} \sum_{t=0}^{T-1} \sum_i \xi_i^*(t). \end{aligned} \quad (21)$$

Since $\mathbb{E}\{L(X_i(T))\} \geq 0$ and $\mathbb{E}\{L(X_i(0))\} = 0$, we have:

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_i \mathbb{E}\{\xi_i(t) | X_i(t)\} \geq \frac{1}{T} \sum_{t=0}^{T-1} \sum_i \xi_i^*(t) - \frac{C}{\rho}. \quad (22)$$

Thus, we prove that, by following RTS, the social welfare for a given MAS is within $O(\frac{1}{\rho})$ of its optimal social welfare.

We now analyze the effect of following RTS on agents' long term pending task queues. In the beginning ($t = 0$) when no tasks have been delegated to any agent, $q_i(0) = 0, \forall i$. For any given i , at some point in time t , its pending workload satisfies $q_i(t) \leq \rho_i \varphi_i^{max}$. Then, at $t + 1$, by following the proposed approach, $q_i(t + 1) \leq \rho_i \varphi_i^{max} + \mu_i^{max}$ (the pending workload can increase at most by μ_i^{max} following Line 2 in Algorithm 1).

In case the pending workload at i becomes $q_i(t) \geq \rho_i \varphi_i^{max}$, we have:

$$q_i(t) + Q_i(t) \geq q_i(t) \geq \rho_i \varphi_i^{max} \geq \rho_i \varphi_i(t). \quad (23)$$

This will trigger RTS to sub-delegate all tasks that cannot be served during time step t to other agents according to Eq. (14). Therefore, at time step $t + 1$, the pending workload becomes $q_i(t + 1) = \lambda_i(t) \leq \mu_i^{max} \leq \rho_i \varphi_i^{max} + \mu_i^{max}$. This proves that by following RTS, the agents' pending workload is bounded by $O(\rho)$.

Experimental Evaluation

After analyzing the theoretical performance bounds of RTS, we now turn to more practical aspects and examine its performance in realistic settings. For this purpose, we compare the performance of RTS with four state-of-the-art approaches in simulations based on an MAS trust network from the *Epinions* trust network dataset (Leskovec, Huttenlocher, and Kleinberg 2010). The dataset allows us to construct realistic

scenarios to compare the performance of RTS against existing approaches. The simulations allow us to better understand the behavior of RTS by varying the parameter settings to create different operating environments. In this section, we discuss our experiment design and results.

Experiment Design

Epinions is an online product rating website. The dataset contains 131,828 vertices each representing a user. Among these vertices, there are 717,667 directed edges with a weight of “+1” representing trust relationships, and 123,705 directed edges with a weight of “-1” representing distrust relationships. Based on this dataset, we generate a simulated multi-agent trust network. For any agent i in the simulation, its n_i^{sub} consists of agents connected with i through a directed “+1” edge originating from i . An agent i can only delegate/sub-delegate tasks to other agents in n_i^{sub} . Each agent has a trustworthiness value $h_i \in [0, 1]$ that affects its probability of completing a task with satisfactory quality. This value is calculated based on the number of other agents trusting and distrusting this agent as reflected in the dataset using the Beta Reputation Model in (Jøsang and Ismail 2002), and is treated as the ground truth about i 's reliability. The μ_i^{max} value of each agent is set in such a way that it is directly proportional to its h_i value.

In the simulations, we assume binary outcomes for tasks (i.e., a task is considered *successful* if the result is correct and produced before its deadline; Otherwise, it is considered *unsuccessful*). We create five duplicate multi-agent trust networks to compare the performance of five different approaches. They are:

1. The Equality-based Approach (EA): tasks are uniformly distributed among trustee agents in each agent i 's n_i^{sub} without regard to their reputation.
2. The Reputation-based Approach (RA): each truster agent adopts the task delegation approach in (Vogiatzis, MacGillivray, and Chli 2010) where the probability of a trustee agent j being selected by a truster agent i is directly related to its reputation standing among all trustee agents in n_i^{sub} .
3. The Global Considerations (GC) Approach: an agent i adjusts the probability for tasks to be delegated to each trustee agent in n_i^{sub} following (Grubshtein et al. 2010).
4. The DRAFT Approach: trustee agents make task request acceptance decisions following (Yu et al. 2013a).
5. The RTS Approach: trustee agents follow RTS.

Truster agents in Approaches 4 and 5 adopt RA.

In the experiments, we vary the overall workload level in the MAS to create different operating environments. As the workload is relative to the aggregate task processing capacity of the agent population, we define a formula to compute the maximum throughput θ of an MAS per time step as $\theta = \sum_i h_i \mu_i^{max}$. At each time step, 20% of the agents are selected at random to act as truster agents and delegate tasks to others. The workload on a given MAS is measured by a metric called the *Load Factor* (LF) which is computed

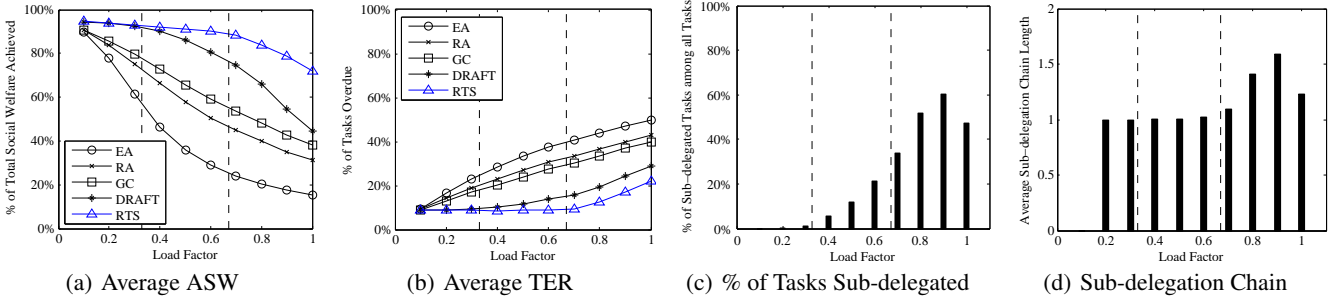


Figure 2: Experiment Results.

as $LF(t) = \frac{w_{req}}{\theta}$, where w_{req} is the amount of new workload generated by truster agents during each time step. In the experiments, we vary the $LF(t)$ value from 0.1 to 1.0 to simulate different workload conditions. Under each $LF(t)$ setting, the simulation is run for $T = 1,000$ time steps. In all experiments, trustee agents process tasks at an average rate of $0.9\mu_i^{max}$ with a standard deviation of $0.1\mu_i^{max}$ following i.i.d. On average, tasks need to be completed within 5 time steps after it is first delegated to a trustee agent.

In the experiments, we measure the performance of each approach using the following metrics:

1. *Achieved social welfare (ASW)*: this metric is computed as the ratio between the sum of the rewards of successfully completed tasks and the sum of the rewards of all the tasks proposed.
2. *Task expiry rate (TER)*: this metric is computed as the ratio between the number of tasks that are not completed before the deadlines over the all the tasks proposed.

Results and Discussions

Figure 2 contains sub-figures showing the performance of the approaches according to the evaluation metrics. Each data point in these figures represents the average value of the selected metric taken over 10 runs of a simulation under a given LF setting. We divide the overall workload condition in the experiments into three groups: 1) *Low Workload* ($0 \leq LF(t) \leq \frac{1}{3}$), 2) *Medium Workload* ($\frac{1}{3} < LF(t) \leq \frac{2}{3}$), and 3) *High Workload* ($\frac{2}{3} < LF(t) \leq 1$). As the results are based on simulations, the trend and relative performance are more important than the exact numerical values.

In Figure 2(a), it can be observed that RTS significantly outperforms existing approaches in terms of ASW. DRAFT, which is similar to RTS in principle but without the task sub-delegation function, performs the best among the existing approaches. GC can be regarded as a variant of RA with the capability of adjusting trustee agents' probabilities of receiving new task requests based on their reputation and current workload. It performs slightly better than RA. EA which does not make use of trustee agents' reputation information, has the worst performance.

The TERs of these approaches are shown in Figure 2(b). Under low workload conditions, both RTS and DRAFT are able to achieve significantly lower TERs than the other approaches. The TER of DRAFT starts to rise under medium workload conditions, while RTS is able to maintain a low

TER. As the workload increases further, the TER of RTS starts to rise, but remains lower than the other approaches.

The reason for the performance of RTS is due to its ability to sub-delegate tasks in the multi-agent trust network. As shown in Figure 2(c), the proportion of all tasks that have been sub-delegated at least once increases with increasing LF. The proportion is the highest (over 20%) under high workload conditions. Similarly in Figure 2(d), the average *sub-delegation chain* length (the average number of times a task is sub-delegated before it is completed or expires) are the highest under high workload conditions. However, as LF approaches 1.0, the trends in Figures 2(c) and 2(d) reverse. This is because as the workload levels approach the full capacity of the MAS, fewer trustee agents have spare capacity to accommodate additional workload other agents attempt to sub-delegate to them. Thus, RTS achieves slightly better performance compared to the best performing existing approach - DRAFT - under low and medium workload conditions, but significantly outperforms it by more than 30% under high workload conditions.

Conclusions and Future Work

We presented a reputation-aware task sub-delegation approach - RTS - to help resource constrained workers in crowdsourcing systems that can be modeled as multi-agent trust networks make efficient task sub-delegation decisions. To the best of our knowledge, RTS is the first approach helping workers determine when, to whom, and the amount of workload to sub-delegate. RTS is a distributed approach that seeks to minimize individual worker's cost associated with task sub-delegation while maximizing its profit, thereby efficiently utilizing the productivity of a crowd. It provides provable performance bounds in terms of social welfare and the growth of pending workload and can be executed in real time (in milliseconds). The complexity of RTS is also flexible as it can be adjusted to fit the scope of real-world requirements. Simulations based on a real trust network dataset demonstrate that RTS is significantly more efficient than existing approaches that do not sub-delegate.

As agents in many MASs are able to adjust the price they charge for their services in response to changing demands (e.g., e-commerce), crowdsourcing systems may facilitate this flexibility in the future. We will incorporate price adjustment decision support into RTS. In addition, we will also consider more sophisticated task properties such as task priority and inter-dependency.

Acknowledgements

This research is supported, in part, by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office (Grant No.: MDA/IDM/2012/8/8-2 VOL 01 and NRF-CRP8-2011-05). Qiang Yang is partly supported by the China National 973 project 2014CB340304 and Hong Kong RGC Projects 621013, 620812, and 621211.

References

- Burnett, C., and Oren, N. 2012. Sub-delegation and trust. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*, 1359–1360.
- Burnett, C.; Norman, T. J.; and Sycara, K. 2011. Trust decision-making in multi-agent systems. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence (IJCAI'11)*, 115–120.
- Grubshtein, A.; Gal-Oz, N.; Grinshpoun, T.; Meisels, A.; and Zivan, R. 2010. Manipulating recommendation lists by global considerations. In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence (ICAART'10)*, 135–142.
- Heymann, P., and Garcia-Molina, H. 2011. Turkalytics: analytics for human computation. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*, 477–486.
- Jøsang, A., and Ismail, R. 2002. The beta reputation system. In *Proceedings of 15th Bled Electronic Commerce Conference*, 41–55.
- Jøsang, A.; Ismail, R.; and Boyd, C. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems (DSS)* 43(2):618–644.
- Leskovec, J.; Huttenlocher, D.; and Kleinberg, J. 2010. Signed networks in social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*, 1361–1370.
- Li, B.; Yu, H.; Shen, Z.; and Miao, C. 2009. Evolutionary organizational search. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)*, 1329–1330.
- Liu, G.; Wang, Y.; and Orgun, M. A. 2011. Trust transitivity in complex social networks. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI-11)*, 1222–1229.
- Man, Z.; Lee, K.; Wang, D.; Cao, Z.; and Miao, C. 2011. A new robust training algorithm for a class of single-hidden layer feedforward neural networks. *Neurocomputing* 74(16):2491–2501.
- Massa, P., and Avesani, P. 2005. Controversial users demand local trust metrics: an experimental study on epinions.com community. In *Proceedings of the 25th American Association for Artificial Intelligence Conference (AAAI-05)*, 121–126.
- Miao, C.; Goh, A.; Miao, Y.; and Yang, Z. 2001. A dynamic inference model for intelligent agents. *International Journal of Software Engineering and Knowledge Engineering* 11(05):509–528.
- Miao, C.; Yang, Q.; Fang, H.; and Goh, A. 2002. Fuzzy cognitive agents for personalized recommendation. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE'02)*, 362–371.
- Neely, M. J. 2010. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers.
- Pan, L.; Meng, X.; Shen, Z.; and Yu, H. 2009. A reputation pattern for service oriented computing. In *Proceedings of the 7th International Conference on Information, Communications and Signal Processing (ICICS'09)*.
- Pickard, G.; Pan, W.; Rahwan, I.; Cebrian, M.; Crane, R.; Madan, A.; and Pentland, A. 2011. Time-critical social mobilization. *Science* 334(6055):509–512.
- Song, H.; Miao, C.; Shen, Z.; Miao, Y.; and Lee, B.-S. 2009. A fuzzy neural network with fuzzy impact grades. *Neurocomputing* 72(13):3098–3122.
- Vogiatzis, G.; MacGillivray, I.; and Chli, M. 2010. A probabilistic model for trust and reputation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'10)*, 225–232.
- Yu, H.; Shen, Z.; Miao, C.; Leung, C.; and Niyato, D. 2010. A survey of trust and reputation management systems in wireless communications. *Proceedings of the IEEE* 98(10):1755–1772.
- Yu, H.; Liu, S.; Kot, A. C.; Miao, C.; and Leung, C. 2011. Dynamic witness selection for trustworthy distributed cooperative sensing in cognitive radio networks. In *Proceedings of the 13th IEEE International Conference on Communication Technology (ICCT'11)*, 1–6.
- Yu, H.; Shen, Z.; Miao, C.; and An, B. 2012. Challenges and opportunities for trust management in crowdsourcing. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'12)*, 486–493.
- Yu, H.; Miao, C.; An, B.; Leung, C.; and Lesser, V. R. 2013a. A reputation management model for resource constrained trustee agents. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 418–424.
- Yu, H.; Shen, Z.; Leung, C.; Miao, C.; and Lesser, V. R. 2013b. A survey of multi-agent trust management systems. *IEEE Access* 1(1):35–50.
- Yu, H.; Shen, Z.; Miao, C.; and An, B. 2013c. 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)*, 1315–1316.
- Yu, H.; Miao, C.; An, B.; Shen, Z.; and Leung, C. 2014. Reputation-aware task allocation for human trustees. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*, 357–364.