CrowdTrust: A Context-Aware Trust Model for Worker Selection in Crowdsourcing Environments

Bin Ye
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
bin.ye@students.mq.edu.au

Yan Wang
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
yan.wang@students.mq.edu.au

Ling Liu
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332, USA
ling.liu@cc.gatech.edu

***Abstract*— On a crowdsourcing platform consisting of task publishers and workers, it is critical for a task publisher to select trustworthy workers to solve human intelligence tasks (HITs). Currently, the prevalent trust evaluation mechanism employs the overall approval rate of HITs, with which dishonest workers can easily succeed in pursuing the maximal profit by quickly giving plausible answers or counterfeiting HITs approval rates.**

In crowdsourcing environments, a worker's trustworthiness varies in contexts, i.e. it varies in different types of tasks and different reward amounts of tasks. Thus, we propose two classifications based on task types and task reward amount respectively. On the basis of the classifications, we propose a trust evaluation model, which consists of two types of context-aware trust: task type based trust (TaTrust) and reward amount based trust (RaTrust). Then, we model trustworthy worker selection as a multi-objective combinatorial optimization problem, which is NP-hard. For solving this challenging problem, we propose an evolutionary algorithm MOWS_GA based on NSGA-II. The results of experiments illustrate that our proposed trust evaluation model can effectively differentiate honest workers and dishonest workers when both of them have high overall HITs approval rates.

***Keywords*- Crowdsourcing; Contextual Trust; Worker Selection; Combinatorial Optimization;**

I. INTRODUCTION

Crowdsourcing provides an economical platform to take full advantage of human wisdom. The term crowdsourcing is first coined by Jeff Howe [1], which is defined as the act of utilizing a scalable number of undefined workers to tackle problems in the form of an open call. Some well-known crowdsourcing platforms, like Wikipedia[1], FreeLancer[2] and Amazon Mechanical Turk[3], have proved that the wisdom of crowds [2] possesses tremendous potential in addressing complex problems. Wikipedia is a remarkable encyclopedia which is continually improved by participants from around the world. FreeLancer is a well-known crowdsouricng platform in Australia. By 2015, over 15,283,124 employers have been hired to do work at FreeLancer in various areas, such as software development, data entry, and article writing [3]. Amazon Turk is a more comprehensive crowdsourcing platform, which engages a diverse, on-demand and scalable workforce to tackle ten thousands of human intelligence tasks (HITs). With the sprawl of crowdsourcing, the issues of trust emerge and become prominent. Firstly, some workers tend to cheat when they can easily get the permission for participating in tasks [4]. In our paper, we name this behaviour as *Distorted Pursuit*. These workers quickly give plausible answers to pursue the maximal profit rather than conscientiously working on the task [5]. They can easily succeed in open-ended HITs, such as comparing two images and completing a survey. The other behaviour is more prevalent in crowdsourcing, known as *Rank Boosting* [6], where dishonest workers boost their overall trust levels by participating in easy tasks or in fake tasks published by themselves.

A variety of trust evaluation approaches for evaluating potential participants' trustworthiness have been proposed in e-commerce, service-oriented applications and social network environments respectively [7][8][9][10]. Compared to these approaches, trust evaluation in crowdsourcing is more complex due to three new characteristics [5][11][12]: (1) unknown workers, (2) weak interactions, and (3) the diversity of HITs. Because ten thousands of unknown workers may participate in the same task, it is not practical for a task publisher to comment all the workers' trustworthiness. In addition, there is rather weak or no social information about interactions between workers and task publishers on crowdsourcing platforms [12]. Thus, the methods for evaluating social network trust are not applicable in crowdsourcing environments. Moreover, a worker's performance varies when participating in different types of HITs and differently rewarded HITs.

To date, crowdsourcing platforms, like FreeLancer and Amazon Turk, use historical HITs records to evaluate workers' trust level. Amazon Turk adopts the *overall approval rate* of HITs to select workers. The *overall approval rate* of HITs is the percentage of the accepted answers in all the submitted answers. Though *overall approval rate* is intuitive for indicating a worker's trustworthiness, it can not determine the priority of workers in an upcoming HIT if they possess the same overall approval rate. For example,

[1] en.wikipedia.org
[2] www.freelancer.com.au
[3] www.mturk.com

worker A who is a writer and worker B who is an image translating expert have the same approval rate at Turk. When they apply to participate in a task of image translation, worker B should be more trustworthy. However, the overall approval rate based work selection mechanism can not differentiate worker A and worker B in such a context. In addition, *Distorted Pursuit* and *Rank Boosting* problems can more easily happen in overall approval rate based trust evaluation systems. Taking a simple case as an example, worker C, who pursues the maximal profit by completing HITs quickly without much effort, can easily obtain a high overall approval rate by performing *Rank Boosting* frauds. Thus, a new trust evaluation approach is highly in demand for crowdsourcing platforms.

In the literature, several qualitative trust management approaches have been discussed to detect dishonest workers in [5][13][14]. However, most of these approaches are not quantitive. In addition, other existing trust models proposed in recent studies [12][15][16] do not consider workers' trustworthiness in different contexts, such as task type and reward amount. Therefore, existing approaches cannot effectively evaluate workers' trustworthiness.

In this paper, we aim to solve the trustworthy worker selection problem based on our proposed *CrowdTrust*, which is a context-aware trust model in crowdsourcing. The main contributions are summarised as follows.

(1) We propose a two-dimensional crowdsourcing trust model, where a worker possesses two context-aware trust values: *TaTrust* and *RaTrust*. According to the components of HITs: input, processing and output, we first propose a novel classification for crowdsourcing tasks. Then we present an approach to calculating task type based trust: *TaTrust*. From the perspective of task reward amount, we propose another classification for HITs, which is used to calculate reward amount based trust *RaTrust (see Section III)*.

(2) Conventional trust evaluation methods [7][8] sum multiple trust values into one value for ranking. However, the weights for summation may contain subjective bias. Thus, we model the trustworthy worker selection into a multi-objective combinatorial optimization problem, which is NP-hard. The objective vector consists of *TaTrust* and *RaTrust*. Then, we propose a Multi-Objective Worker Selection Algorithm MOWS_GA, based on NSGA-II [17], to find the Pareto front [18] in which the combination of trustworthy workers can be determined *(see Section IV)*.

(3) We have conducted simulations on 1000 randomly generated workers to evaluate the effectiveness of the *CrowdTrust*. The results show that our proposed *CrowdTrust* can effectively select workers who are more trustworthy than workers selected by overall approval rate based selection. Furthermore, *CrowdTrust* can effectively differentiate dishonest workers and trustworthy workers when both of them have high HIT approval rates *(see Section V)*.

The paper is organized as follows. In Section II, we give an overview of related research work in crowdsourcing. Section III formalizes the two-dimensional context-aware trust model based on our novel classifying methods for crowdsourcing tasks. Section IV presents our proposed evolutionary algorithm for worker selection. Section V introduces the experimental results and analysis. Finally, we conclude this paper in Section VI.

II. Related Work

With the fast development of crowdsourcing, selecting trustworthy workers has become a critical issue [19]. In particular, identifying untrustworthy workers has drawn much attention from both academia and industry. However, there has been little research work reported in the literature that addresses this issue from the perspective of context awareness.

A. Trust Evaluation Based on Contextual Information

Trust evaluation has been studied in many online environments such as e-commerce, social networks and cloud computing [10][20][21][22]. Through identifying possible hierarchical structures of multi-agent systems, Samek et al. proposed a context-aware trust model to select cooperators [23]. From the perspective of multi-dimensional trust, a trust model is designed to provide references to buyers based on three transaction dimensions (i.e. timeless, quality and cost) in [24]. However, this work does not solve the issue of evaluating a seller's trust level. Compared with conventional practical environments, the trust evaluation problem in crowdsourcing systems is more complex, because the contexts are different from those in conventional environments.

Though context-aware trust evaluation has been proved to be effective in conventional online systems [9][10][22][25], no such work in crowdsourcing has been reported in the literature.

B. Trust Evaluation in Crowdsourcing

In crowdsourcing environments, some qualitative approaches for selecting trustworthy workers have been proposed. In [13], Doan et al. present that trust management schemes, like blocking workers and manual monitoring, can prevent untrustworthy behaviours. However, the drawback is that there is no criteria for identifying untrustworthy behaviours. In 2008, Kittur et al. [14] demonstrate that extra resource are consumed by dealing with untrustworthy workers' responses. Though their experimental results illustrate that setting verification questions can improve the quality of answers, they do not present the method to exactly determine the difficulty and frequency of questions. In 2010, Chen et al. [26] propose a consistent detection approach for protecting systematic cheating behaviours on their crowdsourcing platform. However, their approach is just effective in processing binary-choice problems. In 2011, Hirth et al. [16] propose two mechanisms (MD and CG) to detect cheating behaviours when both verification questions and manual re-checking are ineffective. However, they assume that each worker has the same probability to correctly evaluate answers, which cannot reflect reality in crowdsourcing

environments. In 2012, from the perspective of maximizing social welfare, Yu et al [12] extend three existing trust management models, Beta Reputation System (BRS) [15], knowledge degree model [27] and sequential trust model [28], to adapt to crowdsourcing environments. However, they ignore that a worker's trust level should vary when facing different HITs. In addition, they do not propose any method to calculate the knowledge degree based trust value when there is less available social information about interactions in crowdsourcing environments.

Most of existing works focus on detecting cheating behaviours after workers have submitted answers. These works ignore the fact that the selection of trustworthy workers can effectively improve the quality of answers. In addition, they do not discuss the changes of worker' trustworthiness in different contexts.

To sum up, conventional context-aware trust models can not be directly applied to crowdsourcing environments due to their new characteristics introduced in Section I. There are many limitations by using qualitative approaches for evaluating workers' trustworthiness in crowdsourcing environments.

III. Two-Dimensional Context-Aware Trust Evaluation

On a crowdsourcing platform, a worker may have different trust levels in different contexts. These contexts include the types of tasks and the reward amounts of tasks. Generally speaking, workers perform satisfactorily in their familiar types of HITs. When facing difficult HITs, workers may perform unsatisfactorily. Moreover, the reward amount of a HIT represents the difficulty level of the HIT to some extent [12]. Thus, we propose two HIT classifications: task type based classification and task reward amount based classification. Based on the two classifications, we calculate two types of context-aware trust.

A. Task Type based Trust Evaluation

1) Task Type based Classification of Human Intelligence Tasks: A type of HITs can be decomposed into three dimensions: input, processing and output. For example, there is a task that finding contact information of a toll manufacturer according to an given example. This task consists of coordinates in three dimensions: *input* (a text example), *processing* (finding contract information according to the example), and *output* (text messages).

In Fig. 1, the three HIT dimensions are HIT Input, HIT Processing and HIT Output. In the dimension of HIT Input, there are 5 types: *figural*, *symbolic*, *semantic*, *audio* and *video*, which we summarize from the tasks at Amazon Turk. According to the structure of Guilford's SI model [29], we conclude 5 basic types of HIT Processing, i.e. *cognition*, *memory*, *divergent production*, *convergent production* and *evaluation*, while 6 types of HIT Output are defined: *units*, *classes*, *relations*, *system*, *transformations* and *implications*. Based on the three dimensions, an intelligence space, consisting of 150 (5*5*6) cubes, is established as the container for classifying HITs, which is depicted in Fig. 1.

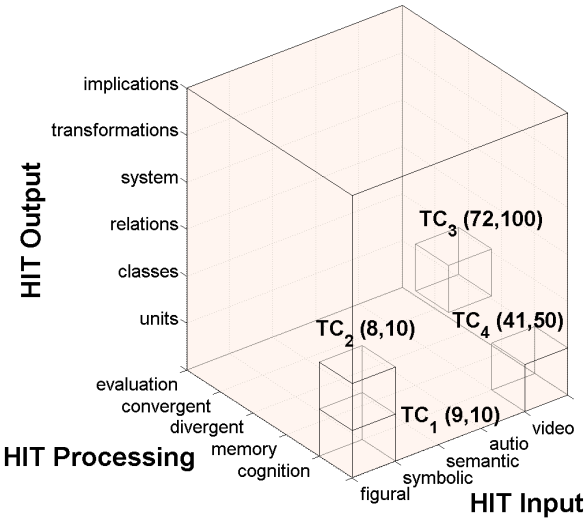


Figure 1. An Intelligence Space for Human Intelligence Tasks Classification

In the three-dimensional intelligence space, each cube represents a type of human intelligence requirements for workers. Thus, HITs with the same intelligence requirement can be classified into the same cube. For example, a task that requires workers to find product information according to a given example is classified into the same cube $c = \{semantic, cognition, units\}$ as the task (finding contact information), because they both have the same input (semantic examples), the processing (cognition for examples), and the output (information units).

In the 3D intelligence space, a worker's historical HIT records are stored in the corresponding cubes. In each cube, ha represents the approval number of HITs and hs represents the submitted number of HITs. We name such a cube with ha and hs as the *Trust Cube* (TC), with which the approval rate of HITs hr in the cube can be calculated as $hr = ha/hs$.

2) Task Type based Trust: Suppose there is a worker with historical historical records in TC_1, TC_2, TC_3 and TC_4. When the worker applies for a HIT, the value of the worker' task type based $TaTrust$ is differently influenced by records in TC_1, TC_2, TC_3 and TC_4.

We define the influence factor inf to represent the different degrees of influence. For example, $inf_{(TC_1,TC_2)}$ represents the degree that the approval rate in TC_1 influences the value of $TaTrust$ when the upcoming HIT belongs to TC_2. The values of influence factor inf range from 0 to 1. The value 0 represents a worker's approval rate in TC_j has no influence on the worker' trustworthiness, and 1 represents the influence is the max. We formulise the task type based trust $TaTrust$ as follows:

$$TaTrust = \sum_{k=0}^{3} \left(\frac{\sum_{i=1}^{n} \sqrt[(4-k)]{ha_{ki}}\, inf_{ki}}{\sum_{i=1}^{n} hs_{ki}} \right), \quad (1)$$

where n is the number of TCs in which a worker possesses

historical records. And k is the number of the same coordinates between two TCs. For example, when the type of the upcoming HITs belongs to TC_1, k of TC_2 is set to 2. Because, the coordinates of $TC2$ and $TC1$ are same in 2 dimensions: HIT Input and HIT Processing.

The influence factor inf is formulized as a function of k and ha by using the sigmoid function in Eq. (2).

$$inf_{ki} = \frac{1}{1 + exp(-g_i(ha, ha_{ki}, k))}, \tag{2}$$

where, $g(ha, ha_{ki}, k)$ is regarded as the independent variable of inf, and $g(ha, ha_{ki}, k)$ is a monotonically increasing function of k and ha.

Influence factor inf is formulised based on three characteristics. First, the marginal influence of HIT records belonging to one TC should be diminishing on a real crowdsourcing platform. Thus the gradient changes should be narrowing when the value of inf_i approaches to the border. We use the sigmoid function to model this characteristic in Eq. (2).

In addition, the influence factor inf between two TCs is determined by k which represents the number of the same coordinates between TCs. For example, if an upcoming HIT belongs to $TC1$ in Fig. 1, the influence factor of each approval HIT in TC_2 ($k = 2$) should be larger than each approval HIT in TC_4 ($k = 1$), i.e. $inf_{(TC_2,TC_1)} > inf_{(TC_4,TC_1)}$. Because, the coordinates of TC_2 and TC_1 are same in 2 dimensions: HIT Input and HIT Processing, while the coordinates of TC_4 and TC_1 are same in one dimension: HITs processing.

Moreover, the differences, existing in the numbers of approval answers in different TCs, may affect the value of the influence factor inf. Suppose worker A's approval rate in $TC2$ is 80% ($ha_A = 8$ and $hs_A = 10$) and worker B's approval rate in $TC2$ is also 80% ($ha_B = 80$ and $hs_B = 100$), worker B is more trustworthy than worker A because worker B completes more HITs with the same approval rate.

We assume the influence generated by the number of approved HITs exponentially drops with the change of k. Based on the independent variables k and ha, the independent variable $g(ha, ha_{ki}, k)$ is defined in Eq. (3).

$$g_i(ha, ha_{ki}, k) = \begin{cases} \frac{\sqrt[(4-k)]{ha_{ki}} - ha}{MIN(ha, \sqrt[(4-k)]{ha_{ki}})}, & ha \neq 0 \\ \sqrt[(4-k)]{ha_{ki}}, & ha = 0 \end{cases} \tag{3}$$

where $k \epsilon \{0, 1, 2, 3\}$ depends on the same coordinates among TCs. According to Eqs. (1) (2) and (3), the task type based trust $TaTrust$ is defined in Eq. (4),

$$TaTrust = \sum_{k=0}^{3} \left(\frac{\sum_{i=1}^{n} \sqrt[(4-k)]{ha_{ki}} \frac{1}{1-exp(-g_i(ha, ha_{ki}, k))}}{\sum_{i=1}^{n} hs_{ki}} \right). \tag{4}$$

All the equations are proposed to calculate a relative trust value among all workers. Thus we do not set the specific parameters for the defined equations.

B. Reward Amount based Trust Evaluation

A worker's trustworthiness may vary with the changes in difficulty levels of the upcoming HITs. However, there is no indicator to directly quantify difficulty levels for HITs in crowdsourcing environments. HITs belonging to the same TC may have different difficulty levels. Reward amounts can indirectly reflect the difficulty levels of HITs. From this perspective, we propose another HIT classification based on the reward amounts of HITs.

In e-commerce, the transaction amount (price) has been proved to be a vital attribute in evaluating a seller's contextual trust level [8][22]. For example, a seller, who usually sells expensive goods with good reputation, is regarded to be trustworthy in an upcoming transaction with a lower price. In our trust model, we go further to consider that all HIT records with different reward amounts jointly influence a worker's final trustworthiness.

In crowdsourcing, a worker, who performs well in a range of reward amounts, is likely to be trustworthy in the HITs belonging to the same range. However, once the reward amount of an upcoming HIT is much higher or lower than the reward amount of tasks that the worker used to participate in, the worker's performance may change with a high probability. Thus, we calculate reward amount based trust *RaTrust* according to a worker's historical approval records in HITs with different reward amounts.

If the reward amount of an upcoming HIT is r', then those HITs rewarded between $\alpha r'$ and $\beta r'$ are classified into one type, where α and β are constants. Through investigating the HITs at Amazon Turk, we get the ratio among maximum reward amount and minimum reward amount. The ratio is around 1000. Thus, the ranges for p are set as the times of 10. We use the ratio p to classify HITs, which is calculated in Eq. (5),

$$p = \begin{cases} 1, \ if \ 0 < \frac{max(r', r_i)}{min(r', r_i)} < 1 \\ 2, \ if \ 1 < \frac{max(r', r_i)}{min(r', r_i)} < 10 \\ 3, \ if \ 10^1 < \frac{max(r', r_i)}{min(r', r_i)} < 10^2 \\ ..., \ ... \\ h, \ if \ 10^{(h-1)} < \frac{max(r', r_i)}{min(r', r_i)} < 10^h \end{cases}, \tag{5}$$

where r_i is the reward amount of a historical HIT record.

Then, we model a HIT record as $H_i = \{\widetilde{ha}_i, \widetilde{hs}_i, r_i, r'\}$, in which $\widetilde{ha}_i$ is the approval number of HITs and $\widetilde{hs}_i$ is the number of submitted HITs. $RaTrust$ represents the trustworthiness of a worker according to the reward amount of the upcoming HIT.

Similar to the calculation method for $TaTurst$, the calculation of $RaTrust$ is defined in Eq. (6):

$$RaTrust = \sum_{p=1}^{h} \left(\sum_{i=1}^{n} \frac{\sqrt[p]{\widetilde{ha}_{pi}} L_{pi}}{\widetilde{hs}_{pi}} \right), \tag{6}$$

where $0 < L < 1$ is determined by the ratio p and $\widetilde{ha}_{pi}$.

First, approval HITs, which have low values of p, influence the *RaTrust* more than those HITs with high values of

p. In addition, the influence of each approval HIT increases when the total number of approval HITs increases. Besides, the marginal influence of records diminishes when the value of L_{pi} approaches the border. Thus, L_{pi} is defined in Eq. (7):

$$L_{pi} = \frac{1}{1 + exp(-(z_i(\widetilde{ha}, \widetilde{ha}_{pi}, p)))}, \tag{7}$$

where, $z_i(\widetilde{ha}, \widetilde{ha}_{pi}, p)$ is regarded as the independent variable of L_{pi}. And $z_i(\widetilde{ha}, \widetilde{ha}_{pi}, p)$ is defined as a monotonically increasing function of p and $\widetilde{ha}$ in Eq. (8).

$$z_i(\widetilde{ha}, \widetilde{ha}_{pi}, p) = \begin{cases} \frac{\sqrt[p]{\widetilde{ha}_{pi}} - \widetilde{ha}}{MIN(\widetilde{ha},\ \sqrt[p]{\widetilde{ha}_{pi}})}, & \widetilde{ha} \neq 0 \\ \sqrt[p]{\widetilde{ha}_{pi}}, & \widetilde{ha} = 0 \end{cases} \tag{8}$$

where p_i is calculated in Eq. (5).

IV. A Worker Selection Algorithm based on Our Two Dimensional Trust Evaluation Model

In crowdsourcing, a task publisher needs to select multiple workers for solving the published HITs. An effective trustworthy worker selection method is vital for preventing the untrustworthy workers from participating in HITs and for selecting more trustworthy workers for solving the HITs. In existing trust models [7][8][9][22], a normalized trust value is calculated based on preset weights of all sub-attributes. The advantage is that different users' trustworthiness can be directly compared. However, it is hard to eliminate the subjective bias caused by weights.

In our proposed method, we do not set weights for task type based trust $TaTrust$ and reward amount based trust $RaTrust$. Instead, we model the trustworthy worker selection problem as a multi-objective combinatorial optimization problem without subjective weights. There are two objectives in our model, i.e. the average $TaTrust$ value and the average $RaTrust$ of the worker combination. In this section, we propose a modified evolutionary algorithm MOWS_GA based on NSGA_II [16] to finding out the efficient worker combination. A worker combination is efficient when none of the objectives can be improved in value without degrading some of the other objective values. Thus, there is no subjective bias. Our algorithm is base on NSGA_II, because NSGA_II is an efficient algorithm for solving multi-objective optimization problems.

A. Modelling Multi-Objective Worker Selection Problem

A HIT requirement vector $HR = (wn, aw, ar, TC, R)$ is generated when a HIT is published on a crowdsourcing platform. The wn represents the fixed number of workers required by a task publisher for solving the HITs. The aw is the number of the current available workers who match the basic requirement ar (i.e. overall approval rate). TC and R represent the trust cube and the reward amount respectively, which are used to calculate context-aware trust $TaTrust$ and $RaTrust$. Then, we formulise trustworthy worker selection problem into a multi-objective combinatorial optimization in Eq. (9),

$$\begin{aligned} f_{(X)} &= minimize(\frac{wn}{\sum_{i=1}^{wn} TaTrust_{(i)}}, \frac{wn}{\sum_{i=1}^{wn} RaTrust_{(i)}}) \\ s.t \quad & X \in D \end{aligned}, \tag{9}$$

where $0 < wn \ll aw$, and X represents a solution. In a feasible solution $X = \{x_1, x_2, ...x_i, ..., x_{aw}\}$, the value of x_i is 0 or 1. $x_i = 0$ represents $worker_i$ is not selected. Conversely, $x_i = 1$ means $worker_i$ is selected in the current feasible solution. Each solution X has a corresponding image point Tr in the objective space, which consists of the average values of two context-aware trust: $\overline{Tr}_1$ and $\overline{Tr}_2$. Thus, Tr' is the corresponding image point of X'. If $\exists i, \overline{Tr}_i > \overline{Tr'}_i$ and $\forall i, \overline{Tr}_i \geq \overline{Tr'}_i$, point Tr dominates Tr'. Corresponding, solution X dominates X'. If no X dominating X' can be found in D, X' is called efficient solution. The image of X' is called non-dominated point. In general, the efficient solution is not unique. Thus, the set of all efficient solutions are named as the efficient set. The images of efficient solutions in the objective space is named *Pareto front*. Our ultimate objective is to select the trustworthy worker combinations with images falling in the *Pareto front*.

B. A Modified Multi-Objective Evolutionary Algorithm MOWS_GA

A number of evolutionary algorithms (MOEAs) have been proposed for solving the multi-objective combinatory optimization problem. NSGA-II has been proved to an effcient approach to solving multi-objective optimization problems [17]. Based on NSGA-II, we propose a modified evolutionary algorithm MOWS_GA to solve the trustworthy worker selection problem in crowdsourcing.

The processes of MOWS_GA are as follows.

Step 1: Generate initial worker combination sets PW with size N, in which PW_i should satisfy $\sum pw = wn$. PW_i represents a worker combination and wn is the number of required workers.

In NSGA-II, the initial solutions are generated randomly to keep global search ability. However, searching a random situation costs much more time than starting with a better solution set. Thus, in MOWS_GA we modified this step by increasing the possibility of selecting those workers who possess obvious good records. Firstly, a worker set is generated after sorting all workers according to $\overline{T}$. The $\overline{T}$ is the sum of task type based trust *TaTrust* and reward amount based trust *RaTrust*, i.e. $\overline{T} = TaTrust + RaTrust$. We preset an initial selection possibility $p_i = \frac{\overline{T}_i}{\overline{T}_{max}}$ for each worker.

Step 2: For each worker combination WC in PW_i, its fitness fit (non-domination level) and density-estimation metric d are calculated by adopting the same methods proposed in NSGA-II.

In the first stage, all worker combinations that belong to the first non-dominated front are identified by comparing the trust values of all the *N* worker combinations. Then, after striping out the first non-dominated front, the second front

can be identified in the similar way. All WCs are divided into corresponding non-dominated fronts by repeating this procedure. The level of a non-dominated front is regarded as the fit for each combination, e.g., 1 represents the first non-dominated front. However, the priority of WCs in the same level can not be determined by simply relying on the fit. Then, we calculate the density-estimation metric d, which is determined by the distance of each objective among current worker combination and the nearest combinations in Eq. (10),

$$d = \frac{|TaTrust^{+} - TaTrust^{-}| + |RaTrust^{+} - RaTrust^{-}|}{TaTrust^{max} - TaTrust^{min} + RaTrust^{max} - RaTrust^{min}}. \quad (10)$$

Step 3: Select a worker combination set SW_k^i (size $n = N/2$) from WC by using usual binary tournament strategy. Crossover and mutation operators are executed to generate offspring population QW_k^i (size p). ζ represents the possibility to execute crossover in QW_k^i. In MOWS_GA, the crossover operator is modified to satisfy the constraint that the number of selected workers is fixed. In our MOWS_GA, we modify the mutation operator to be an adaptive variable $\sigma = \sigma\gamma$, which is calculated in Eq. (11),

$$\gamma_i = \begin{cases} min(\frac{\sum_{i=1}^{n} \frac{\overline{T}_{i(j-1)} - \overline{T}_{i(j-2)}}{\overline{T}_{ij} - \overline{T}_{i(j-1)}}}{n}, 1), \ if \ j \geq 2 \\ 1, \ if \ j < 2 \end{cases} . \quad (11)$$

The γ represents the increasing ratio of trust values between two evolutions. If the increase of ratio is bigger than the last time, which means the mutation promotes the increase of the trust values. Thus, we use γ to decline the value of σ to avoid time consumption in searching other directions. We set 0.2 as the maximum value of σ, because frequent mutations influence the convergence of the algorithm.

Step 4: Use elitism to select a worker combination set (size N) from $PW^i \bigcup QW^i$. The combinations in the set are stored in $OW^{(i+1)}$. Adopt the same strategy to cross and mutate $OW^{(i+1)}$ to generate the next generation $PW^{(i+1)}$.

Step 5: Check whether the termination condition is satisfied. Once the number of iterations reaches the preset maximal value or no new dominated solution appears during 10 continuous iterations, the interaction is terminated. Otherwise, go to **Step 2**.

The complexity of non-dominated sorting is $O(M(2N)^2)$, in which M is the number of optimal objectives and N is the number of worker combinations. Thus, the overall complexity of MOWS_GA is $O(TN^2)$, where T is the iteration times. Though the complexity of MOWS_GA is the same as that of NSGA-II, we have modified the initialization and adaptive mutation operator to improve the efficiency of the evolutionary algorithm.

V. Experiments and Analysis

In this section, we evaluate our proposed *CrowdTrust* in a scenario where a task publisher needs to select 100 workers from 1000 workers. The 1000 workers' overall HIT approval rates are preset to satisfy the requirement of the task publisher. The experiment settings are presented in Section V-A. The results and the analysis are introduced in Section V-B.

A. Experiment Setting

As there is no available worker dataset in crowdsourcing environments, we generate synthetic data for simulations. In the synthetic dataset, some workers are preset to have *Distorted Pursuit* and *Rank Boosting* behaviours. The simulations are conducted for selecting trustworthy worker combinations from the dataset.

Table I
Constraints for 1000 Workers

Behaviour	Percentage	Constraints
Rank Boosting	20%	$k = rand$(0,1),$p = rand$(3,4) or $k = rand$(0,1,2,), $p = rand$(3,4)
Distorted Pursuit	15% (dishonest)	$k = rand$(0,1,2), $p = rand$(2,3,4)
	5% (marginal)	$k = rand$(1,2,3), $p = rand$(1,2,3)
Honest	60%	$k = rand$(2,3) $p = rand$(1,2)

In the synthetic dataset, workers' historical HIT records are randomly generated with a series of constraints listed in Table I. We generate the records of 1000 workers in different TCs and reward amount ranges as follows.

First, the numbers of workers' submitted HITs are randomly generated based on the normal distribution. Then, each worker's overall approval rate is randomly generated in the range of 90% and 95%, because workers, with an overall approval rate of 90% or above, are always permitted to participate in HITs on crowdsourcing platforms, e.g., Amazon Turk.

According to the overall HITs records, 200 dishonest workers (i.e. 20% of 1000 workers) are generated with *Rank Boosting* behaviours. In addition, 150 dishonest workers (15%) with *Distorted Pursuit* behaviours are generated. Considering workers with *Distorted Pursuit* behaviours may apply for HITs that they are good at, we generate 50 marginal workers (5%) with *Distorted Pursuit* behaviours to some extent. These marginal workers possess a certain number of honest records in the HITs similar to the upcoming one though they perform *Distorted Pursuit* frauds in other HITs.

Furthermore, we generate 600 honest workers (60% of 1000 workers). Honest workers' records are generated in those tasks, which have similar types and reward amounts to the upcoming HIT.

The parameters for MOWS_GA are listed in Table II.

Table II
Parameters for MOWS_GA

Objective Variable	Decision Variable	Population Size (N)
2	1000	20
Max Iterations (T)	**Crossover probability (ζ)**	**Mutation probability (σ)**
500 / 1000	0.9	0-0.2

In our simulations, there are 2 objective variables: *TaTrust* and *RaTurst* and 1000 decision variables. In order to increase

the capability of global search in MOWS_GA, the value of crossover probability ζ is set to 0.9. Mutation probability σ is modified to be an adaptive one in the range of 0-0.2, which can avoid time consumption caused by excessive mutations.

Considering the efficiency of MOWS_GA, the population size of initial worker combinations is set to a relative small size 20, and the max times of iterations is set to 500 and 1000 respectively.

B. Performance Comparison in Trustworthy Worker Selection

In the literature, no context-aware solution has been reported for selecting trustworthy workers in crowdsourcing environments. Thus, we compare the performance difference between the overall approval rate based selection *ARS* and our proposed *CrowdTrust*. On crowdsourcing platforms, e.g., Amazon Turk, if workers' overall approval rates satisfy the preset requirement (in general 90%), they are selected on the first-come-first-serve basis. Because all workers in our synthetic dataset are randomly generated with an overall approval rate above 90%, we first use *ARS* to randomly select 20 worker combinations. Each combination has 100 workers. Then, we calculate $TaTrust$ and $RaTrust$ for each worker combination, which are compared with the results selected by *CrowdTrust*.

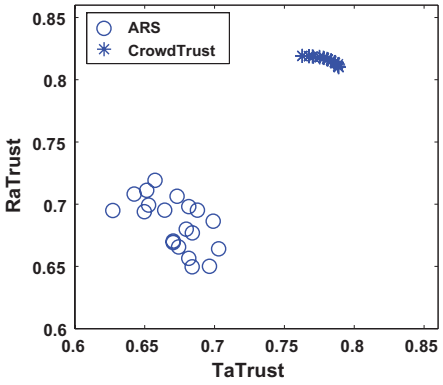
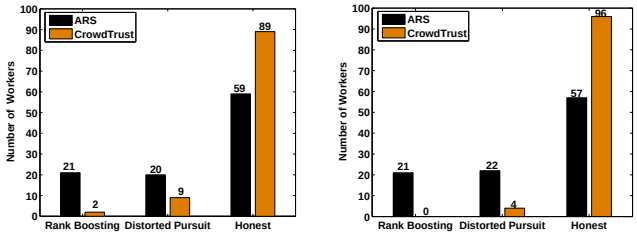


Figure 2. The Comparison in $TaTrust$ and $RaTrust$

Result 1. Fig. 2 plots the trust values of worker combinations selected by *ARS* and *CrowdTrust* respectively. From Fig. 2, we can observe that the best $TaTrust$ and $RaTrust$ values in *ARS* are 0.71 and 0.73 respectively. By contract, the best $TaTrust$ and $RaTrust$ values in the worker combinations selected by *CrowdTrust* are 0.805 and 0.81 respectively, which are 13.4% and 10.9% higher than the ones delivered by *ARS*. Thus, our proposed *CrowdTrust* can select worker combinations with on average 10% higher context-aware trust values than the trust values of worker combinations selected by *ARS*.



(a) Selection Result (500 Iterations) (b) Selection Result (1000 Iterations)

Figure 3. The Comparison in Trustworthy Worker Selection

Result 2. Fig. 3 plots the average numbers of untrustworthy workers and honest workers in the worker combinations selected by *ARS* and *CrowdTrust* respectively. From Fig. 3, we can see that compared to *ARS*, *CrowdTrust* selects fewer workers with untrustworthy behaviours and more workers with honest behaviours.

In Fig. 3(a), after 500 iterations, 21 workers with *Rank Boosting* behaviours and 20 workers with *Distorted Pursuit* behaviours, are selected by *ARS*. This is close to 40% - the percentage of the workers with frauds in the dataset. By contract, the total number of workers with these behaviours selected by *CrowdTrust* is 11 only, which is 73.2% less than the one selected by *ARS*. In addition, 89 honest workers are selected by *CrowdTrust*, which is 50.9% more than the number delivered by *ARS*.

Fig. 3(b) plots the results of 1000 iterations. From Fig. 3(b), we can see that the numbers of workers who perform *Rank Boosting* or *Distorted Pursuit* frauds are 21 and 22 respectively in *ARS*. By contrast, the numbers are 0 and 4 respectively in *CrowdTrust*. Through observing the trust values of the 4 selected workers with *Distorted Pursuit* behaviours, we find that the 4 workers are all marginal workers, who possess a certain number of honest records in the HITs similar to the upcoming one. In addition, 96 honest workers are selected by *CrowdTrust*, which is 68.4% more than 57 honest workers selected by *ARS*.

From the above results, we can conclude that our proposed *CrowdTrust* can select worker combinations with higher $TaTrust$ and $RaTrust$ values than *ARS* and effectively identify workers with untrustworthy behaviours.

VI. Conclusions

In crowdsourcing environments, trust issue has been taken as a pressing task in the literature [11][13][19]. In this paper, we have proposed two classifications for Human Intelligence Tasks (HITs). Based on the classifications, a context-aware trust model *CrowdTrust*, for calculating the task type based trust $TaTrust$ and task reward amount based trust $RaTrust$, has been proposed. To the best of our knowledge, this is the first solution in the literature to evaluating workers' trust from the perspective of context awareness. For solving the trustworthy worker combinations selection problem with two context-aware trust objectives, which is NP-Hard, we have proposed a modified evolutionary algorithm MOWS_GA based on NSGA-II. The

results of experiments conducted on a synthetic dataset have demonstrated that *CrowdTrust* can effectively identify dishonest workers. However, our proposed approach may be vulnerable to attack when workers counterfeit fake records and apply for tasks in a specific type of tasks. In future work, we plan to extend our proposed trust model to detect some attacks, e.g., *Sybil Attack*, towards a robust trust evaluation model in crowdsourcing environments.

REFERENCES

[1] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[2] J. Surowiecki, *The wisdom of crowds*. Random House LLC, 2005.

[3] "About freelancer," http://www.freelancer.com/about, 2015.

[4] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgment and Decision making*, vol. 5, no. 5, pp. 411–419, 2010.

[5] C. Eickhoff and A. de Vries, "How crowdsourcable is your task," in *Proceedings of the workshop on crowdsourcing for search and data mining (CSDM) at the fourth ACM international conference on web search and data mining (WSDM)*, 2011, pp. 11–14.

[6] P. Ipeirotis, "Be a top mechanical turk worker: You need $5 and 5 minutes," *Blog: Behind Enemy Lines*, 2010.

[7] H. Zhang, Y. Wang, and X. Zhang, "Transaction similarity-based contextual trust evaluation in e-commerce and e-service environments," in *Web Services (ICWS), 2011 IEEE International Conference on*. IEEE, 2011, pp. 500–507.

[8] Y. Wang and E.-P. Lim, "The evaluation of situational transaction trust in e-service environments," in *e-Business Engineering, 2008. ICEBE'08. IEEE International Conference on*. IEEE, 2008, pp. 265–272.

[9] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan, "Trust management towards service-oriented applications," *Service Oriented Computing and Applications*, vol. 3, no. 2, pp. 129–146, 2009.

[10] G. Liu, Y. Wang, and M. A. Orgun, "Social context-aware trust network discovery in complex contextual social networks." in *AAAI*, vol. 12, 2012, pp. 101–107.

[11] J. Howe, *Crowdsourcing: How the power of the crowd is driving the future of business*. Random House, 2008.

[12] H. Yu, Z. Shen, C. Miao, and B. An, "Challenges and opportunities for trust management in crowdsourcing," in *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*. IEEE Computer Society, 2012, pp. 486–493.

[13] 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.

[14] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2008, pp. 453–456.

[15] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, 2002, pp. 41–55.

[16] M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*. IEEE, 2011, pp. 316–321.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[18] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.

[19] C. Eickhoff and A. P. de Vries, "Increasing cheat robustness of crowdsourcing tasks," *Information retrieval*, vol. 16, no. 2, pp. 121–137, 2013.

[20] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[21] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments." *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[22] H. Zhang, Y. Wang, and X. Zhang, "A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments." in *SOCA*, 2012, pp. 1–8.

[23] J. Samek and F. Zboril, "Hierarchical model of trust in contexts," in *Networked Digital Technologies*. Springer, 2010, pp. 356–365.

[24] A. Caballero, J. A. Botía, and A. Gómez-Skarmeta, "On the behaviour of the trsim model for trust and reputation," in *Multiagent System Technologies*. Springer, 2007, pp. 182–193.

[25] H. Zhang, Y. Wang, and X. Zhang, "Efficient contextual transaction trust computation in e-commerce environments," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012, pp. 318–325.

[26] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of euphoria: a crowdsourcing platform for qoe assessment," *Network, IEEE*, vol. 24, no. 2, pp. 28–35, 2010.

[27] V. Muñoz, J. Murillo, B. López, and D. Busquets, "Strategies for exploiting trust models in competitive multi-agent systems," in *Multiagent System Technologies*. Springer, 2009, pp. 79–90.

[28] M. Hoogendoorn, S. W. Jaffry, and J. Treur, "Exploration and exploitation in adaptive trust-based decision making in dynamic environments," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 2. IEEE, 2010, pp. 256–260.

[29] J. P. Guilford, "The nature of human intelligence." 1967.