Nobody of the Crowd: An Empirical Evaluation on Worker Clustering in Topcoder

Razieh Saremi
rsaremi@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

Hamid Shamszare
hshamsza@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

Marzieh Lotfalian Saremi
m.lotfa@encs.concordia.ca
Concordia University
Montreal, QC, Canada

Ye Yang
yyang4@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey

ABSTRACT

Context: Software crowdsourcing platforms typically employ extrinsic rewards such as rating or ranking systems to motivate workers. Such rating systems are noisy and only provide limited knowledge about worker's preference and performance.

Goal: The objective of this study is to empirically investigate patterns and effects of worker behaviour in software crowdsourcing platform in order to improve the success and efficiency of software crowdsourcing.

Method: First, we create the bipartite network of active workers based on common registration for tasks. Then, we use Clauset-Newman-Moore graph clustering algorithm to identify developer clusters in the network. Finally, we conduct empirical evaluation to measure and analyze workers behaviour per identified cluster in the platform by workers' ranking. More specifically, workers behaviour is analyzed based on worker reliability, worker trustworthiness, and worker success as measures for workers' performance, worker efficiency, and worker elasticity to represent workers' preferences, and worker contest, worker confidence, and worker deceitfulness to understand workers' strategies. The empirical study is conducted on more than one year's real-world data from topcoder, one of the leading software crowdsourcing platforms.

Results: We identify four clusters of active workers: mixed ranked, high ranked, mid ranked, and low ranked. Based on statistical analysis, this study can only support that the low ranked group associates with the highest reliable workers with average reliability of 25%, while mixed ranked group contains the most trustworthy workers with average trustworthiness of 16%.

Conclusions: These findings are helpful for task requesters to understand preferences and relations among unknown resources in the platform and plan for task success in a more effective and efficient manner in software crowdsourcing platform.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

• **Software and its engineering → Software development process management**; • **Information systems → Data analytics**.

KEYWORDS

software crowdsourcing, worker network, worker performance, worker preference, worker success, competition strategy

ACM Reference Format:
Razieh Saremi, Hamid Shamszare, Marzieh Lotfalian Saremi, and Ye Yang. 2021. Nobody of the Crowd: An Empirical Evaluation on Worker Clustering in Topcoder. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Crowdsourced software development (CSD) has gained increased popularity in recent years, however, there are many risks associated with it. Major risks are associated with the uncertainties on both the number of registrants and quality of the received submissions from the unknown workers [37] [34] [10]. This raises the challenge on how to engage unknown workers and trust on their work [35]. To encourage contribution and engagement of crowdworkers, crowdsourcing platforms employ the use of extrinsic rewards such as rating [6]. To that end, crowdsourcing platforms employ different reputation systems to manage crowd rating based on their participation history. For example, HITs rate is used in Amazon Mechanical Turk [10], and numeric developer rating based on Elo rating algorithm is used in TopCoder [30] . However, such rating systems are noisy since the reliability or reputations of the workers are often unknown [17], and only provide limited knowledge about worker's preference and performance. Also, according to competitive exclusion principle[15][25], rating systems can provide advantages for a group of workers over others in competition. This advantage can result in frequent win of higher ranked workers in competitions.

In general, competing for success over shared tasks among different ranked workers creates a complex network of workers. To understand the dynamic among workers in the network, it is essential to identify the existing clusters in this network and investigate the behaviour of crowd workers per cluster. To that end, we need to develop a better understanding of the sensitivity in worker preference, performance, and engagement [37] [19][32][20] per cluster.

To the best of our knowledge, there is no investigation on impact of workers behaviour among different cluster of crowd workers on workers performance and success.

The objective of this study is to empirically investigate patterns and impact of crowd workers performance and preferences in software crowdsourcing platform in order to improve the success and efficiency of software crowdsourcing. In this study, we first present a motivational example to shed light on the identification of four identified cluster of workers in software crowdsourcing platform. Then, we propose an empirical evaluation framework, and develop an approach to characterizing and analyzing workers behaviour in each identified cluster of workers. More specifically, this includes grouping similar ranked workers, and then studying the worker preference, performance and strategy per cluster. The empirical study is conducted on more than one year's real-world data from topcoder[1], the leading software crowdsourcing platform with an online community of over 1.5M workers and 55k mini-tasks. The evaluation results show that: 1) there are four active clusters of workers in the pool of workers: mixed ranked, high ranked, mid ranked, and low ranked; 2) lower ranked group provide the highest level of reliable workers to make a submission; 3) mixed ranked group respect their expertise to compete fairly on a task; 4) mixed ranked group apply strategies to assure their success.

The remainder of this paper is structured as follows. Section II introduces a motivating example that inspires this study. Section III presents background and related work. Section IV outlines our research design. Section V presents the empirical results. Section VI discusses the key findings of our study. Section VII presents the conclusion and outlines a number of directions for future work.

2 MOTIVATING EXAMPLE

Figure 1 depicts a motivating example with task selection, task status, workers expertise ranking and competition information among six workers and five tasks. The information on top of each workers are workers ID and workers expertise rank based on topcoder five-level rating belt definition[2] and number of task registration and submission (WR,WS). The number under the tasks represents task ID, task status (C = complete and F = failed), and the number of workers registration and submissions (R,S). As it is shown in figure 1, three of workers are high ranked (i.e Yellow, and Blue), one is mid ranked (i.e Green) and the other two are lower ranked from Gray belt. Worker VI is registering for two tasks and won both, while worker III won one of the three tasks s/he registered. Workers IV and V became the runner up for one of the tasks they registered.

The information provided from choice of task per worker enable us to create the network of workers and study their behaviour. Figure 2 illustrates the network of workers based on the common choice in registering for tasks. According to the network of workers, worker VI is a high ranked worker (i.e Blue) and won both tasks s/he registered for while competing against lower ranked workers(Green and Gray workers). Interestingly other successful workers (i.e worker III, worker IV, and worker V) had a diverse competitors in terms of experience level when they won a tasks or became runner up. It is possible that worker VI plan strategies based on "Hawk-Dove" game [26], in which less aggressive competitors will yield to aggressive competitors. In crowdsourced environments, workers historical score rate would directly affect on the final competition [1], and can be used to identify easier competition [24]. This observation motivated us to investigate patterns and effect of workers choices in task competitions. We believe the result of such study enabling task requests to understand workers preferences and relations in the platform and plan for task success in a more efficient manner.

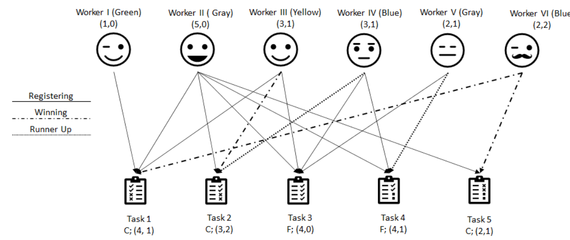

Figure 1: Overview of Motivating Example

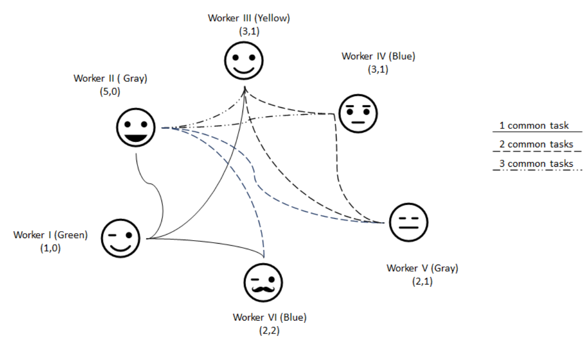

Figure 2: Network of Workers of the Motivating Example

3 BACKGROUND AND RELATED WORK

3.1 Workers Cluster in Crowdsourcing

To encourage contribution and engagement, many communities including crowdsourcing adopt the use of extrinsic rewards such as ranking as game elements for workers to compete in a non-gaming context [6]. Extrinsic rewards can increase the overall workers' engagement and commitment[3][4], motivation [33][31][5] and collaboration [14], since they address a type of social need for some community members[27]. In a CSD platform, a competitive environment not only influences the decisions of workers regarding which tasks to register and submit but also how they react to their peers. Therefore, many challenges in crowdsourcing such as worker motivation [33][31][5], collaboration [18], creativity [38][20]and performance and trust [22][35] can be tackled from online communities aspect [39].

The principle of online communities have been used to improve the quality of performance of crowd workers[9]. Also it is reported that crowd workers take collective action to improve their own

[1]https://www.topcoder.com/
[2]https://www.topcoder.com/member-onboarding/understanding-yourtopcoder-rating/

working situation[28]. Moreover, defining cluster of workers in the pool of crowd workers is beneficial for both platform and task owner in different ways such as workers loyalty to the platform, collaboration among workers and trust [39].

3.2 Workers Performance in Crowdsourcing

Software workers’ arrival to the platform and the pattern of taking tasks to completion are factors that shape the worker dynamic in a crowdsourcing platform, however, the reliability in returning the qualified tasks creates the dynamic of the platform. Generally, not only would the award associated with the task influence the workers’ interests in competitions[36], the number of registrants for the task, the number of submissions by individual workers, and certainly the workers’ historical score rate would directly affect their final performance [21][30]. For newcomers or beginners, there is a time window required to improve and to develop into an active worker [11]. Therefore, it is typical that the workers need to communicate with the task owner in order to better understand the problems to be solved [20]. Existing studies show that over time, registrants gain more experience, exhibit better performance, and consequently gain higher scores [11] [2] [20]. Still, there are workers who manage not only to win but also to raise their submission-to-win ratio [8]. This motivate workers to develop behavioral strategies in topcoder [1] [2]. Moreover, the ranking mechanism used by topcoder contributes to the efficiency of online competition and provides more freedom of choice for higher rate workers in terms of controlling competition level [1].

3.3 Competition Strategies in Crowdsourcing

Crowdsourcing a project inherently involves a concern for how reliable and trustworthy the unknown crowd workers are [35]. It is reported that unreliable workers are not very interested in taking novel tasks that require creativity and abstract thinking [10] . Due to the diversity of workers with different individual skill levels, it is not practical for the requester to evaluate all the workers’ trustworthiness [35], nor is there a clear record of workers’ interaction in the pool of workers [10]. This fact creates a trust network among the worker community itself; which is a result of workers’ rating, skill set and history of winning a task. Such networks creates opportunities for workers to apply different strategies and assure their success and increasing their rank in the system. One primary example is rank-boosting [16] in Amazon Mechanical Turk, where workers mostly register for easy tasks or fake tasks that they themselves are uploading in order to increase their rating, or distorted pursuit [35], in which workers quickly submit a possibly correct answer in order increase their benefits instead of working on the task and submitting acceptable answer. Another example is detecting cheap talk phenomena [12][1] in topcoder. In CSD, higher rated workers have more freedom of choice in comparison with lower rated workers and can successfully affect the registration of lower rated workers. To assure a softer and easier competition level, higher rated workers register early for some specific projects while lower rated workers must wait for higher rated workers to make their choice.

4 RESEARCH DESIGN

Crowd workers are exchanging information and developing professional or social contacts. Such interaction leads to creation of a *complex network*, in which mutual motivation factors and personal preferences in taking a task among workers creates different *clusters*. Workers in each cluster are densely connected to each other and loosely connected to the workers in the other clusters in the network of workers [23].
In this study, to investigate dynamic behaviour patterns of crowd workers’ in terms of performance, preference, and finally strategies in taking and submitting tasks based on different workers’ clusters, a preliminary analysis is conducted using data from topcoder platform.

4.1 Empirical Evaluation Framework

To develop a better understanding of the workers’ dynamics in different clusters of workers in task supply and execution, we design four evaluation study to investigate the differences in the preferences, performances, and strategies of workers in each identified cluster; and effect of each cluster on delivering successful crowdsourcing tasks is studied.

The proposed evaluation framework is illustrated in figure 3. Next, we explain the analysis steps and introduce a few working definitions used in this framework.

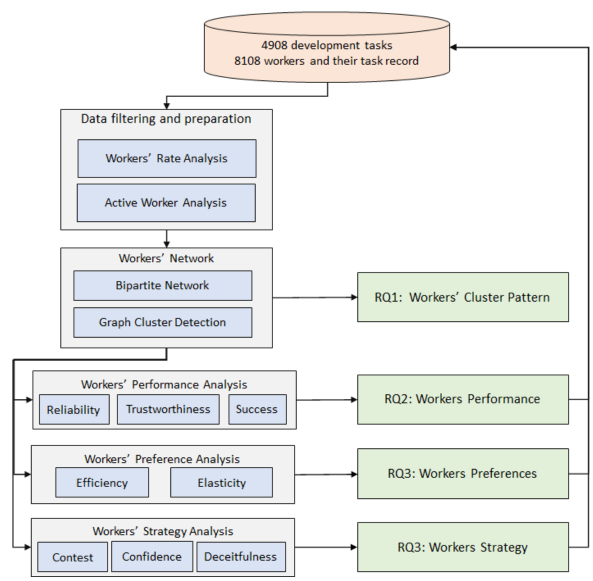

Figure 3: Main Flow of the Proposed Empirical Evaluation Framework and Relationship to Research Questions

4.1.1 *Workers’ Network.*
We created a bipartite network of active workers based on the number of common registered tasks among workers and detected the existed clusters in the network. To detect the existed clusters, we applied Clauset-Newman-Moore greedy modularity maximization algorithm [7], one of the most used graph clustering methods. This algorithm is hierarchical agglomeration therefor it supports graph classes and does not consider

edge weights. The algorithm begins with each node in its own cluster and joins the pair of clusters that makes the most increase in modularity until no such pair exists. As the result of this analysis, we identified four clusters of workers in the network of active workers. This analysis helps us to reveal the hidden relation among the workers in the network.

Each cluster is a subset of the whole network of active workers [13]. Therefore, each identified cluster should contain different type of workers in terms of expertise, and ranks. We followed topcoder rating belt, introduced in part 4.3, and grouped workers per cluster to 5 different belts of Gray, Green, Blue, Yellow, and Red. This helps us to investigate workers behaviour per identified cluster of workers. Since, in the entire dataset, we only had eight workers from Red belt (i.e 0.16% of workers), this group of workers will be ignored in further analysis.

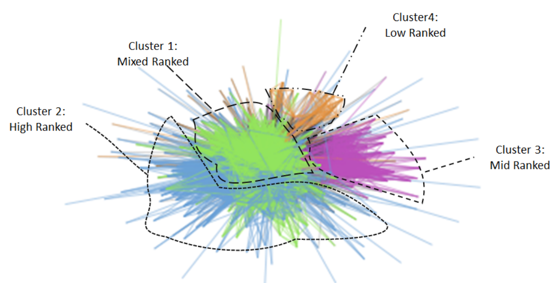

Figure 4: Identified Clusters in the Network of Active Workers

Identified Clusters: Figure 4 shows the four identified clusters in the dataset. Based on the diversity of workers' experience rank, table 2, per identified cluster, the four cluster are named as:

(1) *Mixed Ranked,* contains the largest group of workers with a diverse expertise rate.
(2) *High Ranked,* represents the second largest group of workers, also it contains the largest group of higher experienced ranked workers(Yellow and Blue).
(3) *Mid Ranked,* as the third largest cluster represents the highest number of average experienced ranked workers (Green and Blue).
(4) *Low Ranked,* presents the smallest cluster which also created with mostly lower experienced ranked workers (Gray).

Then, for each cluster, we measured 4 different network analysis metrics, namely, "Number of Common Neighbors", "Worker Rank", "Closeness Centrality", and "Betweenness Centrality". The definition of these metrics are as follow:

- *Number of Common Neighbors (#CN)*: Measures the number of common workers between any two workers in the sub-network;
- *Worker Rank (WR)*: Counts the number and quality of links to a worker to determine a rough estimate of how important the worker is;
- *Closeness Centrality (CC)*: Measures the average shortest distance from each worker in the cluster to the other workers;
- *Betweenness Centrality (BC)*: Measures the extent to which a worker lies on paths between other workers;

4.1.2 Worker Performance. In this study, worker performance is defined as a tuple of worker reliability (RL_i), worker trustworthiness (TL_i) and worker success (SL_i). Worker reliability is the probability of receiving a submission by workers in the same cluster of workers, while worker trustworthiness is the probability of receiving a valid submission by worker in the same cluster; worker success represents the probability of a worker making a valid submission and win in the same cluster.

Note that each worker is a tuple of the number of registrations (R_k), number of submissions (S_K), and the number of valid submissions ($VS_{k.}$).

$$WorkerPerformance_i = (RL_i, TL_i, SL_i)$$

Worker reliability, worker trustworthiness, and worker success are defined as below:

Def. 1: Worker Reliability Level, RL_i, Representing the ratio of the worker i registrations, S_i, to the total registrations by his/her cluster, CR_k.

$$RL_i = \frac{\sum_{i=0}^{n} S_i}{\sum_{k=0}^{n} CR_k} \tag{1}$$

Tasks owners trust on receiving a qualified submission after receiving a registration by a worker. Such level of trust is quantified as Worker Trustworthiness Level.

Def. 2: Worker Trustworthiness Level, TL_i, measures average valid submission, VS_i, for worker i to register for tasks to compete with the same Cluster workers, CR_{ki}.

$$TL_i = \frac{\sum_{i=0}^{n} VS_i}{\sum_{k=0}^{n} CR_k} \tag{2}$$

While receiving a valid submission increases workers trustworthiness, it does not guarantee the workers' success. There are different reasons to not win a task even with a valid submission. Therefore, a worker's success level, in this research, is defined as a function of the worker's frequency of task winning and the other cluster members'.

Def. 3: Worker Success Level, SL_i, measures average win W_i, for worker i to register for tasks to compete with the same cluster workers, CyR_k.

$$SL_i = \frac{\sum_{i=0}^{n} W_i}{\sum_{k=0}^{n} CyR_k} \tag{3}$$

4.1.3 Worker Preference. Worker preference for worker i is defined as a tuple of the worker's efficiency in working on a task that requires worker's skill set (EF_i) and workers' elasticity involvement with the other workers from the same cluster in a taken task (EL_i).

$$WorkerPreference_i = (EF_i, EL_i)$$

Worker efficiency, and worker elasticity are defined as follow:

To understand workers' efficiency in performing a task, first we need to understand workers' proficiency in the requirement technology to perform a task.

Def. 4: Worker i Proficiency Level in the technology j, $PL_{i,j}$, is defined as worker i's registration frequency,$WTech_{i,j}$, in tasks with technology j as a requirement compared to the total registration frequency by the all the workers in the same cluster for those tasks .

$$PL_{i,j} = \frac{WTech_{i,j}}{\sum_{k=1}^{n} WTech_{k,j}} \tag{4}$$

Def. 5: Worker Efficiency Level, EF_i, measures the average proficiency of workers $PL_{i,j}$ who register for tasks competing with the same worker cluster, CyT_k.

$$EF_i = \frac{\sum_{i=0}^{n} PL_{i,j}}{\sum_{i=0}^{n} CyT_k} \tag{5}$$

In project k, the maximum number of registrants for the project tasks from the all clusters, TC_k, and cluster l, $RC_{k,l}$, is represented as:

Def. 6: Cluster l Elasticity, EL_l, is the ratio of the $RC_{k,l}$, to TC_k

$$EL_i = \frac{\sum_{i=0}^{n} RC_{k,i}}{\sum_{i=0}^{n} TC_k} \tag{6}$$

4.1.4 Worker Strategy. Worker strategy is defined as tuple of Workers' contest which is the ratio of number of registrants with lower rank to total number of registrants on a task (CT_i), workers' confidence in making a submission for a task (CL_i) and workers' deceitfulness which is the probability of a worker register for a task an the task starves (DL_i) .

$$WorkerStrategy_i = (CT_i, CL_i, DL_i)$$

Worker contest, worker confidence, and worker deceitfulness are defined as follow:

Def. 7: Worker Contest Level, CT_i, measures the ratio of number registrants for a task from lower ranked belt LRR_j to total task competition level TC_j on the task j that worker i registered for per worker cluster.

$$CT_i = \frac{\sum_{j=0}^{m} LRR_j}{\sum_{i=0}^{n} TC_j} \tag{7}$$

Def. 8: Worker Confidence Level CL_i defines as the maximum task competition level TCL_j that worker$i$ make a submission for a task.

$$CL_i = \max(TCL_i) \tag{8}$$

$$where : S_i > 0$$

Def. 9: Worker deceitfulness Level, DL_i, measures the ratio of number of tasks that worker i registered for and starved (received zero submission), ST_i, to total number of tasks that worker i registered for R_i.

$$DL_i = \frac{\sum_{i=0}^{n} ST_i}{\sum_{i=0}^{n} R_i} \tag{9}$$

4.2 Research Questions

Four research questions are formulated as following:

- *RQ1 (Workers Cluster Patterns)*: How do workers distribute in different cluster in a competitive CSD?
  This research question aims at providing general overview of workers' distribution per identified cluster based on the members rank and expertise in the platform in the CSD platform;
- *RQ2 (Worker Performance)*: How does different workers cluster impact workers' performance?
  Understanding worker reliability, worker trustworthiness and worker success per identified worker cluster can be good measure to indicate worker consistency to perform a task;
- *RQ3 (Worker Preferences)*: How does different worker cluster pattern impact workers' preferences in taking a task?
  The degree of worker efficiency, and worker elasticity per identified cluster represent workers' choice to get involved and compete on tasks.
- *RQ4 (Worker Strategy)*: How do workers from different identified cluster guarantee their winning in a competition?
  The degree of worker contest, worker confidence, and worker deceitfulness per identified cluster represent workers' strategy to compete on tasks.

4.3 Dataset

The dataset from topcoder contains 403 individual projects including 4,907 component development tasks (ended up with 4,770 after removing tasks with incomplete information) and 8,108 workers from January 2014 to February 2015 (14 months). Tasks are uploaded as competitions in the platform, where crowd software workers would register and complete the challenges. When the workers submit the final files, it will be reviewed by experts to check the results and grant the scores. This flow creates dynamic attributes of workers that influences successful task delivery in the platform.

4.3.1 Data Preparation. The introduced dataset contains tasks attributes such as required technology, platform, task description, task status, monetary prize, days to submit, registration date, submission date, and workers attributes such as registration date, submission date, valid submission, winning placement, winning status, rating score, and winning score. In this step, workers' performance, preferences and strategies metrics are not included.
We used available data and create attributes such as workers skillsets (WTech), which is proxy by the number of technologies (#Tech) required to perform taken tasks by the worker. We create binary variables for each technology (Tech) required in each task, where:

$$WTech(x,s) = \begin{cases} 1 & \text{worker x is an expert in technology s} \\ 0 & \text{otherwise} \end{cases}$$

We create a dataset of worker attributes for each identified cluster. The summary of metrics definition used in the analysis are presented in Table 1.

Workers' Rate: TopCoder adopts a numeric worker rating system based on Elo rating algorithm Elo rating is a method for calculating the relative skill levels of players in competitor-versus-competitor games [30]. Based on this numeric rating and a five-level rating scheme, TopCoder divides worker community into 5 groups. The 5 worker groups are defined into 5 belts of Red, Yellow, Blue, Green and Gray, which corresponds to the highest skillful workers to the lowest ones[30]. The numeric ratings are with respect to three different task categories including algorithm, marathon matches and development, following sophisticated calculation algorithm.

Table 1: Summary of Metrics Definition

Type	Metrics	Definition
Workers Attributes	# Registration (R)	Number of registrants that are willing to compete on total number of tasks in specific period of time.
	# Submissions ($S_{i,t,k}$)	A binary variable correspond to registration t of worker i in cluster k, equal to 1 for a submission and zero for lack of submission by its submission deadline in specific period of time.
	# Valid Submissions ($VS_{i,t,k}$)	a binary variable correspond to registration t of worker i in cluster k, equal to 1 for a submission and zero for lack of a submission by task's submission deadline and passed the peer review and labeled as either completed or active.
	Win ($W_{i,t,k}$)	A binary variable correspond to registration t of worker i in cluster k, equal to 1 for a submission and zero for lack of a submission that pass the peer review and labeled as active submission.
Workers-Tasks Attributes	Task Status	Completed or failed task.
	Task Competition Level (TC)	Total number of workers to register and are willing to compete on a task.
	# Starved Tasks (ST)	Number of Tasks that receives zero submission by its submission deadline and failed.
	Technologies (Tech)	Number of technologies used in task.
	Platform (PLT)	Number of platforms used in task.
	Worker expertise (WTech)	Number of technologies and platforms used in tasks that worker compete on.

Table 2: Summary of Different Workers' Belt [29]

Workers' Belt	Rating Range(X)	Workers%	p(qualified Sub)
Gray	$X < 900$	90.02%	0.25
Green	$900 < X < 1200$	2.88%	0.45
Blue	$1200 < X < 1500$	5.39%	0.39
Yellow	$1500 < X < 2200$	1.54%	0.6
Red	$X > 2200$	0.16%	0.6

We followed topcoder rating algorithm to identify the workers belonging to different rating belt in our dataset. Table 2 summarize the distribution of workers in different rating belt in our dataset.

Active Workers: To identify active workers in the dataset, we grouped workers' attribute per month and labeled workers who had minimum one registration as active worker. This reduced the number of workers from 8180 to 2259. Then, we created the network of active workers based on frequency in common task registration.

4.4 Empirical Study Design

To empirically investigate the evaluation framework, we design analysis to answer the four research questions in section 4.2 and conduct experiments using real-world data collected over a period of more than one year, section 4.3. Figure 3 summarizes the steps associated with the study conducted. In the following subsections, details related to the workers' cluster pattern, workers' performance analysis, workers' preference analysis and workers strategies analysis are presented.

4.4.1 Decision Variables. In any type of study there are 3 types of decision variables, namely dependent, independent and control variables.

The independent variables are the ones which values are not affected by the other variables in the system. In this research the independent variables are registration (R), submissions (S), valid Submissions (VS), and win (W).

On the other hand there are dependent variables, which are extracted from either the independent variables or the other dependent variables. In this research variables defined in section 4.1, i.e equations 1 to 9 such as worker reliability level, worker trustworthiness, worker success level, and worker confidence level, are the dependent variables. The dependent variables are used to provide a tangible understanding toward workers' preferences and performances.

Also, there is a subset of independent variables named as control variables. The control variables are representative of workers' decisions making and directly impact their performances. In this research workers' registration (R) and submissions (S) are the control variable.

4.4.2 Workers' Cluster Pattern. Workers' cluster pattern helps to identify the existed cluster of workers among the active workers in our dataset. To identify existed clusters w we perform the following:

first, we analyzed the worker-worker relation by calculating the number of times both workers were competing on a same tasks. This analysis provided us with a dataset of "source", "target" and "weight", in which source was the chosen worker, target is worker who had similar or partially similar competition history with the source, and weight is frequency of both source and target registered for the same task. Using this dataset we created the bipartite network of workers.

Second, we identified existed clusters in the network of workers. For identifying the existed clusters we used Clauset-Newman-Moore greedy modularity algorithm [7] as one of the most used graph clustering methods. In this step we found four active clusters with diverse combination of workers.

Third, we analyzed each clusters by different metrics of number of common neighbors, worker rank in the network, closeness centrality and betweenness centrality. Also, we took a deeper look on the workers diversity in terms of expertise level based on topcoder definition [30].

4.4.3 Workers' Performance Analysis. We investigate the performance of different ranked workers per cluster by looking into workers' reliability, trustworthiness and success. The probability of a worker make a submission after registered for a task will be reported as worker reliability, the probability of a submission passes the peer review and labeled as valid submission provides workers trustworthiness, and the probability of a submission passed the peer review and labeled as win provides workers success. In topcoder, crowd worker's reliability of competing on the tasks is measured based on last 15 competitions workers registered and submitted. For example, if a worker submitted 14 tasks out of 15 last registered tasks, his reliability is 93% (14/15).

4.4.4 Workers' Preference Analysis. We investigate the preference of different ranked workers per cluster to get involved in a task by looking into workers' efficiency and elasticity. The probability of a worker register for a task with same technical requirements of workers expertise reports as worker efficiency, and the ratio of maximum number of registrant per task in a project per cluster by the maximum number of registrant per task in a project in the platform provides worker elasticity.

4.4.5 Workers' Strategy Analysis. We investigate the potential strategies that higher ranked workers (Blue and Yellow belt) may take per cluster to assure easier competition level and their success. To that end we looked into workers' confidence, the average task competition level that a higher ranked worker make a submission. Then we looked at the workers contest which is the ratio of number of lower ranked workers register for the task that higher ranked workers registered too. And finally, workers' deceitfulness is the probability of a higher ranked worker registered for a task and the task starved. The result of this part is relevant since it is an indication of fairness in a competition per identified cluster of workers.

5 EMPIRICAL RESULTS

5.1 Workers Cluster Pattern (RQ1)

We identified four clusters in the network of workers in our dataset. Then, we applied four most common network analysis metrics on the identified clusters to understand the dynamic of workers within them. Based on the result of these analysis, we defined clusters as:

(1) *Mixed Ranked* contains the largest group of workers with a very mix expertise rate. This cluster, on average, has 52 common workers per each pair of workers. Each worker has 0.6 probability to be close to closest neighbour and 0.2 probability to be in another worker shortest path. Also, the average worker rank is 0.07, which means there is 7% chance that a worker is competing on a task against a highly similar worker;
(2) *High Ranked* represents the second largest group of workers; it contains the largest group of higher ranked workers (15% Blue and 11% Yellow) among all four clusters. Interestingly, workers in this cluster have the average largest number of common neighbour (i.e. 88) with standard deviation of 51. The closeness centrality is 0.59 and average betweenness centrality is 0.018 with average worker rank of 0.07;
(3) *Mid Ranked* contains 305 members from the pool of active workers and represents the highest level of mid ranked workers with 48% population from Green and Blue belt. On average, each workers have 65 common neighbors with average closest centrality 0.6 and average betweenness centrality of 0.012. Also the average worker rank is 0.075;
(4) *Low Ranked* presents the smallest cluster which also created with mostly lowest ranked workers. Workers in this cluster on average have 37 common neighbors with closeness centrality of 0.58 and betweeness centrality of 0.21. Also, each worker on average receives the degree of importance of 0.06.

Table 3 summarized the details of each cluster.

Table 3: Summary of Identified Clusters

Cluster # Workers	Rate Belt		#CN	WR	CC	BC
Mixed Ranked — 906	Gray: 52% Green: 30% Blue: 11% Yellow: 7%	Mean	52	0.07	0.6	0.02
		Std	21	0.056	0.087	0.056
High Ranked — 850	Gray: 59% Green: 15% Blue: 15% Yellow: 11%	Mean	88	0.07	0.59	0.018
		Std	51	0.056	0.088	0.055
Mid Ranked — 305	Gray: 44% Green: 36% Blue: 12% Yellow: 8%	Mean	65	0.075	0.60	0.012
		Std	35	0.057	0.090	0.058
Low Ranked — 198	Gray: 72% Green: 11% Blue: 12% Yellow: 5%	Mean	37	0.060	0.58	0.021
		Std	23	0.064	0.11	0.063

Finding 1.1: High ranked cluster provides the highest level of interaction among it's member.

Finding 1.2: Workers in low ranked cluster have highest probability to register for the same task as any other pair of two workers in the same cluster.

5.2 Worker Performance (RQ2)

In order to have a better understanding of workers' performance, we studied workers reliability, trustwortiness and success per cluster. To do so, we created four different datasets based on the identified clusters of workers in the network, then we calculated each of the metrics per dataset.

Worker Reliability: According to Figure 5, while high ranked cluster correspond to an increasing trend of worker reliability as workers' belts increase from 0% in Gray to 13% in Yellow. Interestingly, mixed and mid ranked clusters seem to follow the similar pattern, in both clusters Gray and Blue workers are among the highest reliable workers. while in mid ranked cluster both Blue and Gray workers provide the same level reliability of 17%; in mixed ranked cluster, Blue workers provide slightly higher level of reliability than Gray workers (i.e 26% and 24%). Also, in the mixed ranked, Green workers are the least reliable workers with the reliability score of 13%; in the mid ranked cluster, Yellow workers bring the lowest level of reliability (i.e 2%). Moreover, the highest reliability among workers belongs to Green belt from low ranked cluster with 45% reliability. The second lowest level of reliability among the low ranked (Blue workers) is equal to the highest level of reliability that mid ranked workers can provide (i.e 17%). And, the second highest reliable workers in the low ranked cluster are Yellow belt workers with reliability of 24% and the lowest reliability belongs to Gray workers with 13%.

We ran repeated measure one-way ANOVA test on the worker reliability result from the four different worker cluster. Based on ANOVA test results, the worker reliability is significantly different across the existed cluster since the p-value is 0.025.

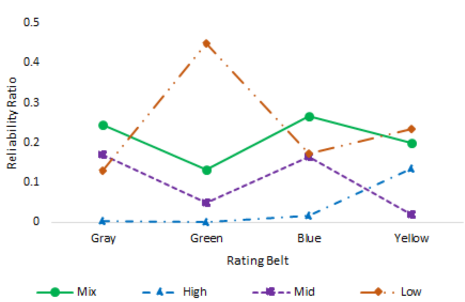

Figure 5: Average Worker Reliability per Belt per Cluster of Workers

Worker Trustworthiness: Besides attracting reliable workers to make a submission, it is important to trust on the workers' submissions. Hence, we investigate workers' trustworthiness ratio in returning valid submission. Figure 6 represents the distribution of workers' trustworthiness ratio among different workers' belt per cluster. The mixed ranked cluster provides the highest trustworthy workers among all the clusters. Green workers in this cluster are the most trustworthy workers with 27% of trust following by Gray and Yellow workers with 15% and 12% trust level, respectively; the Blue workers of mixed ranked cluster only provide 6% of trust. The high ranked cluster contains the least trustworthy workers, while Gray and Green workers in high ranked cluster provide trust level of 1% and 0%; the Blue and Yellow workers are as much trustworthy as their same ranked peers in low ranked cluster (1% and 4% respectively). Also, Gray workers in mid ranked cluster and Green belt workers of low ranked cluster provide 11% of trust. ANOVA test showed that different clusters significantly influence workers' trustworthiness (i.e. p-value is 0.022).

Worker Success: Workers' performance can be analyzed based on workers' capability of returning a valid submission and win

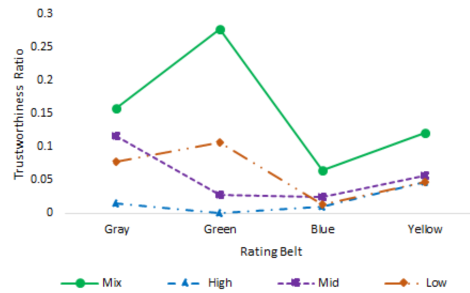

Figure 6: Average Workers' Trustworthiness per Belt per Cluster of Workers

the competition at the same time. Therefore, we analyzed workers success based on different workers' ranked belts in different clusters of workers. Figure 7 illustrates the average workers' success per cluster. As it is shown in Figure 7, mixed ranked cluster contains the highest successful workers with success level of 45%, 51% , 72% and 50% for Gray, Green, Blue and Yellow belts, respectively. The second cluster with overall highest success level is mid ranked, with highest success rate from the Gray belt with 45% followed by Yellow, Green and Blue belts for 32%, 28% and 26%, respectively. Low ranked cluster is the third place in workers success level with on average 16% lower than mid ranked cluster. The highest success level in this cluster belongs to Gray workers for 23% followed by Green and Yellow for 17% each. However, the high ranked cluster has the lowest overall success rate, it contains the most successful workers in the pool of workers. Gray belt from this cluster have 67% success. ANOVA test showed that worker reliability is significantly different among all four clusters with p-value of 0.047.

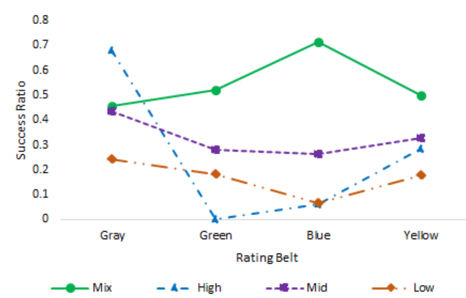

Figure 7: Average Workers' Success Per Belt per Cluster of Workers

Finding 2.1: Workers in the mix ranked cluster on average provide a higher level of trustworthiness and success, however, workers in the low ranked cluster are the most reliable ones.

5.3 Worker Preference (RQ3)

For adaptive teams to leverage CSD to increase team elasticity, it is critical to understand crowd worker's sensitivity and preference in taking tasks and rate of team elasticity for them. Therefore, we studied the workers efficiency and workers elasticity per different cluster of workers. we calculated each of the metrics per cluster dataset.

Worker Efficiency: Workers' efficiency can be analyzed based on workers' preferences for performing tasks with requirements from the works's list of expertise. We analyzed workers efficiency based on different workers ranked belts in different clusters of workers.

Figure 8 displays the average workers' efficiency per cluster. As it is shown, by increasing workers rank, in mixed ranked cluster, efficiency levels increases. While Gray workers choose to compete on tasks with on average 28% expertise efficiency, Green workers provide 36% efficiency following by 40% efficiency for both Blue and Yellow workers. On the other hand, in mid ranked cluster, by increasing workers rank, efficiency level follows a flat U-shape. Workers in Gray belt provide the highest level of efficiency on this cluster with 21%, following by 18% efficiency from Yellow workers. The lowest level of efficiency comes from Green workers with 15% efficiency and Blue workers with 16% efficiency.

The average level of efficiency for entire cluster puts high ranked cluster in the second place after mixed rank cluster with 35% efficiency. In this cluster, Gray and Blue workers have higher efficiency level than mixed ranked (i,e 38% and 52%, respectively) while Green and Yellow workers provide much less efficiency than their peers in mixed ranked cluster with 18% and 34%, respectively. In low ranked group, Gray, Blue, and Yellow workers provide efficiency levels less than 5%, however, Green worker have on average 48% efficiency. This puts Green workers from low raked cluster in the second place after Blue workers from high ranked cluster.

ANOVA test showed that worker efficiency is not significantly different among all four clusters with p-value of 0.078.

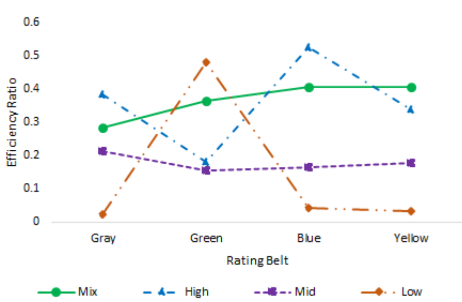

Figure 8: Average Workers' Efficiency Per Belt per Cluster of Workers

Workers Elasticity: After understanding different patterns of workers efficiency for each cluster, we investigate the workers elasticity in crowdsourced tasks per workers ranked belt per cluster. Figure 9 illustrates the distribution of workers elasticity per worker belt per cluster. Low ranked cluster provides highest level of worker elasticity with the average of 16% . In details, this cluster of workers has 21% elasticity for Blue workers, following with 17% for Green and 15% for Gray. Yellow workers in this cluster provide the minimum level of elasticity of 12%. Both mixed and high ranked clusters provide almost same level of elasticity of, on average, 14%. Interestingly, workers in both clusters follow almost the same pattern. In mixed ranked cluster, Yellow workers provide the highest level of 30% elasticity, followed by Green workers with 12% , Gray workers with 8% and Blue workers with 5% elasticity. In high ranked cluster, Yellow workers bring the highest level of elasticity in the table for 23%, followed by Green workers with 17% elasticity and both Blue and Gray workers have 10% worker elasticity. In mid ranked cluster, Blue workers provide 13% worker elasticity, Yellow have 8% , Gray 7% and Green 6% worker elasticity. These observations provide mid ranked cluster with the least level of elasticity of, on average, 8%. The result of the ANOVA test showed that task workers elasticity is not significantly different across all four workers clusters (i.e. p-value is 0.45).

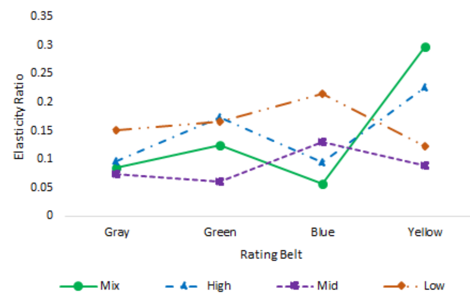

Figure 9: Average Workers' Elasticity Per Belt per Cluster of Workers

Finding 3.1: In general workers in mixed ranked cluster prefer to work on tasks that they have higher expertise on. Blue workers from high ranked cluster are bringing the maximum worker efficiency followed by Green workers in low ranked cluster.

Finding 3.2: On average, workers from low ranked cluster provide higher worker elasticity in taking tasks from same group of tasks.

5.4 Worker Strategy (RQ4)

To understand the strategies that workers with higher ranked in each cluster may take to guarantee their success or having easier competition level per task, we analyzed worker confidence, worker contest, and worker deceitfulness for high ranked group of workers per cluster. Table 4 summarized these metrics.

Table 4: Average Workers' Strategy per Cluster of Workers

Cluster	*R*	*CL*	*CT*	*DL*
Mixed Ranked	12	2	0.85	0.25
High Ranked	11	4	0.70	0.21
Mid Ranked	13	6	0.75	0.13
Low Ranked	10	3	0.79	0.20

Worker Confidence: As it is presented in table 4, high rank workers from mixed ranked cluster register for tasks with on average 12 registrants, however, they have the confidence to submit the tasks which received one more registrants beside them (i.e 2). In high ranked cluster, workers have a confidence level of 3 which means they make a submission when they are competing with another 3 workers. High ranked workers from this cluster register for tasks with, on average, 11 registrants. Mid ranked cluster workers register to compete on tasks with, on average, 13 registrants, however, their confidence level of making a submissions is 6 registrants per task . And, low rank cluster workers have the submissions confidence level of 3, while they register for tasks with average registration of 10.

Worker Contest: According to table 4, higher ranked workers in mixed ranked cluster try to have the easiest competition with contest level of, on average, 0.85. This means these workers register for tasks when, on average, 85% of registrants are from lower to mid ranked belts (i.e 10 out of 12). Workers from low ranked cluster are in the second rate of easy competition with contest level of 0.79, following by mid ranked cluster which has the contest level of 0.75. Workers from high ranked cluster have the contest level of 0.7 (8 out of 11 workers are from lower to mid ranked belt). These group are having the hardest competition level in terms of opponent expertise level among their peers.

Worker Deceitfulness: Mid ranked cluster provides the lowest level of worker deceitfulness, on average 0.13. Low and high ranked clusters are very close with 0.20 and 0.21 degree of deceitfulness, respectively. Interestingly, the highest level of deceitfulness belongs to mixed ranked cluster with 0.25.

Finding 4.1: Higher ranked workers from mix ranked cluster not only have the lowest confidence in making submissions, but also have the highest level of deceitfulness.

6 DISCUSSION

6.1 Workers Cluster Patterns

We found four worker cluster patterns in a network of workers. We identified that workers from high ranked cluster have the highest level of interaction among their cluster members, finding 1.1. Also, we observed that workers in low ranked cluster have the highest probability to register for the same task as any other pair of two workers in the same cluster, fining 1.2.

6.2 Workers Performance

To successfully crowdsource a software project in a CSD platform, it is important to understand workers sensitivity and performance in taking tasks. To that end, this research investigated workers' reliability, workers' trustworthiness, and workers' success per detected worker cluster. We observed that level of trustworthiness is higher in a cluster with more fairness in terms of workers' expertise level (i.e mixed ranked),however, the highest level of reliability happened in the cluster with least experienced workers, finding 2.1.

6.3 Worker Preference

To assure of having a successful project in CSD, beside the importance of attracting trustworthy and successful workers whom make a valid and acceptable submission, it is crucial to make sure to attract efficient workers who not only deliver the task but also have the required skill set to make a useful delivery. Also, to understand the reliability of each cluster of workers in resource shortage in per tasks, we analyzed workers' elasticity level. And, the results of investigating workers preference under different workers clusters presented that in general cluster of workers with mixed ranked expertise give higher weight to their skill sets for taking a new task, finding 3.1. And, the lower ranked cluster provides higher worker elasticity and therefore lower risk of resource shortage. However, there is no statistical difference among the clusters when analyzing workers' efficiency and elasticity, finding 3.2.

6.4 Worker Strategy

It is reported that higher rated workers are tempted to apply *"cheap talk"* in order to soften their competition [2][1]. To understand how higher ranked workers in different cluster may apply such strategies we investigate workers' confidence level in make a submission, workers' contest in making sure they have a soft and easy competition and workers' deceitfulness to check the result of their strategies/approach in task starvation. We observed that workers from the mixed ranked cluster have higher potential in applying strategies to assure their success, finding 4.1.

6.5 Threats to Validity

First, the study only focuses on competitive CSD tasks on the topcoder platform. Many more platforms do exist, and even though the results achieved are based on a comprehensive set of about 5,000 development tasks, the results cannot be claimed externally valid. There is no guarantee the same results would remain exactly the same in other CSD platforms.

Second, there are many different factors that may influence workers' preference, performance and decision in task selection and completion. Our worker cluster approach is based on known task-worker attributes in topcoder. Different approaches may lead us to different but almost similar results.

Third, the result is based on workers network only. Workers communication was not considered in this research. In future we need to add this level of research to the existing one.

7 CONCLUSION AND FUTURE WORK

To understand the probability of a probability of workers' success and fairness in a crowdsource platform, not only one should understand the active available cluster of workers in the platform but also, they need to understand the workers performance, preference and chosen strategy in competing on a task. This research investigated available cluster of workers by applying Clauset-Newman-Moore greedy algorithm and observed the workers behaviors within the identified clusters. Then analyzed workers performance, preferences, and strategies per cluster based on workers rating level in the platform.

Based on statistical analysis, this study can only support that the low ranked cluster provide the highest reliability level and mixed rand cluster contains the most trust worthiness workers.

In future, we would like to evaluate our finding in crowdsourced software development practice and testing the scalability of them in real time.

REFERENCES

[1] Nikolay Archak. 2010. Money, glory and cheap talk: analyzing strategic behavior of contestants in simultaneous crowdsourcing contests on TopCoder. com. In *Proceedings of the 19th international conference on World wide web*. 21–30.

[2] Nikolay Archak and Anindya Ghose. 2010. Learning-by-doing and project choice: a dynamic structural model of crowdsourcing. *LEARNING* 1 (2010), 1–2010.

[3] Sanat Kumar Bista, Surya Nepal, Nathalie Colineau, and Cecile Paris. 2012. Using gamification in an online community. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. IEEE, 611–618.

[4] Huseyin Cavusoglu, Zhuolun Li, and Ke-Wei Huang. 2015. Can gamification motivate voluntary contributions? The case of StackOverflow Q&A community. In *Proceedings of the 18th ACM conference companion on computer supported cooperative work & social computing*. 171–174.

[5] Dana Chandler and Adam Kapelner. 2013. Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior and Organization* 90 (2013), 123–133. https://doi.org/10.1016/j.jebo.2013.03.003

[6] Ruijia Cheng and Mark Zachry. 2020. Building Community Knowledge In Online Competitions: Motivation, Practices and Challenges. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–22.

[7] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Phys. Rev. E* 70 (Dec 2004), 066111. Issue 6. https://doi.org/10.1103/PhysRevE.70.066111

[8] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2016. Scheduling human intelligence tasks in multi-tenant crowd-powered systems. In *Proceedings of the 25th international conference on World Wide Web*. 855–865.

[9] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. 1013–1022.

[10] Carsten Eickhoff and Arjen P de Vries. 2013. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval* 16, 2 (2013), 121–137.

[11] Siamak Faradani, Björn Hartmann, and Panagiotis G Ipeirotis. 2011. What's the right price? pricing tasks for finishing on time. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[12] Joseph Farrell and Matthew Rabin. 1996. Cheap talk. *Journal of Economic perspectives* 10, 3 (1996), 103–118.

[13] Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99, 12 (2002), 7821–7826.

[14] Mary L Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. 2016. The crowd is a collaborative network. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. 134–147.

[15] Garrett Hardin. 1960. The competitive exclusion principle. *science* 131, 3409 (1960), 1292–1297.

[16] Panos Ipeirotis. 2010. Be a top mechanical turk worker: You need $5 and 5 minutes. *Blog: Behind Enemy Lines* (2010).

[17] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. 2014. Reputation-based worker filtering in crowdsourcing. *Advances in Neural Information Processing Systems* 27 (2014), 2492–2500.

[18] Aniket Kittur. 2010. Crowdsourcing, collaboration and creativity. *XRDS: crossroads, the ACM magazine for students* 17, 2 (2010), 22–26.

[19] Aniket Kittur, Ed H Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 453–456.

[20] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The future of crowd work. In *Proceedings of the 2013 conference on Computer supported cooperative work*. 1301–1318.

[21] Karim R Lakhani, David A Garvin, and Eric Lonstein. 2010. Topcoder (a): Developing software through crowdsourcing. *Harvard Business School General Management Unit Case* 610-032 (2010).

[22] Rochelle LaPlante, M Six Silberman, and Industriegewerkschaft Metall. 2016. Building trust in crowd worker forums: Worker ownership, governance, and work outcomes. *Proceedings of WebSci16. ACM* (2016), 35–63.

[23] Fragkiskos D Malliaros and Michalis Vazirgiannis. 2013. Clustering and community detection in directed networks: A survey. *Physics reports* 533, 4 (2013), 95–142.

[24] Yashar Moshfeghi, Alvaro F Huertas-Rosero, and Joemon M Jose. 2016. Identifying careless workers in crowdsourcing platforms: a game theory approach. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 857–860.

[25] Arnaud Pocheville. 2015. The ecological niche: history and recent controversies. In *Handbook of evolutionary thinking in the sciences*. Springer, 547–586.

[26] Anatol Rapoport and Albert M Chammah. 1966. The game of chicken. *American Behavioral Scientist* 10, 3 (1966), 10–28.

[27] Ganit Richter, Daphne R Raban, and Sheizaf Rafaeli. 2015. Studying gamification: the effect of rewards and incentives on motivation. In *Gamification in education and business*. Springer, 21–46.

[28] Niloufar Salehi, Lilly C Irani, Michael S Bernstein, Ali Alkhatib, Eva Ogbe, and Kristy Milland. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 1621–1630.

[29] Razieh Saremi, Ye Yang, Gregg Vesonder, Guenther Ruhe, and He Zhang. 2021. CrowdSim: A Hybrid Simulation Model for Failure Prediction in Crowdsourced Software Development. *arXiv preprint arXiv:2103.09856* (2021).

[30] Razieh L Saremi, Ye Yang, Guenther Ruhe, and David Messinger. 2017. Leveraging crowdsourcing for team elasticity: an empirical evaluation at TopCoder. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. IEEE, 103–112.

[31] Wael Soliman and Virpi Kristiina Tuunainen. 2015. Understanding continued use of crowdsourcing systems: An interpretive study. *Journal of theoretical and applied electronic commerce research* 10, 1 (2015), 1–18.

[32] Alexander Sorokin and David Forsyth. 2008. Utility data annotation with amazon mechanical turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 1–8.

[33] Osamuyimen Stewart, Juan M Huerta, and Melissa Sader. 2009. Designing crowdsourcing community for the enterprise. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*. 50–53.

[34] Ye Yang and Razieh Saremi. 2015. Award vs. worker behaviors in competitive crowdsourcing tasks. In *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–10.

[35] Bin Ye, Yan Wang, and Ling Liu. 2015. Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments. In *2015 IEEE International Conference on Web Services*. IEEE, 121–128.

[36] Ming Yin, Yiling Chen, and Yu-An Sun. 2014. Monetary interventions in crowdsourcing task switching. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

[37] Han Yu, Chunyan Miao, Zhiqi Shen, Cyril Leung, Yiqiang Chen, and Qiang Yang. 2015. Efficient task sub-delegation for crowdsourcing. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[38] Lixiu Yu and Jeffrey V Nickerson. 2011. Cooks or cobblers? Crowd creativity through combination. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1393–1402.

[39] Simon à Campo, Vasssilis-Javed Khan, Konstantinos Papangelis, and Panos Markopoulos. 2019. Community heuristics for user interface evaluation of crowdsourcing platforms. *Future Generation Computer Systems* 95 (2019), 775–789. https://doi.org/10.1016/j.future.2018.02.028