

General Robustness Evaluation of Incentive Mechanism against Bounded Rationality Using Continuum-Armed Bandits

Zehong Hu,¹ Jie Zhang,² Zhao Li^{1,✉}

¹Alibaba Group, Hangzhou, China

²School of Computer Engineering, Nanyang Technological University, Singapore
zehong.hzh@alibaba-inc.com, zhangj@ntu.edu.sg, lizhao.lz@alibaba-inc.com

Abstract

Incentive mechanisms that assume agents to be fully rational, may fail due to the bounded rationality of agents in practice. It is thus crucial to evaluate to what extent mechanisms can resist agents' bounded rationality, termed robustness. In this paper, we propose a general empirical framework for robustness evaluation. One novelty of our framework is to develop a robustness formulation that is generally applicable to different types of incentive mechanisms and bounded rationality models. This formulation considers not only the incentives to agents but also the performance of mechanisms. The other novelty lies in converting the empirical robustness computation into a continuum-armed bandit problem, and then developing an efficient solver that has theoretically guaranteed error rate upper bound. We also conduct extensive experiments using various mechanisms to verify the advantages and practicability of our robustness evaluation framework.

Introduction

Incentive mechanism has been widely adopted to coordinate a group of self-interested agents, with two common objectives: 1) incentivize agents to follow the strategies desired by mechanism designers; 2) maximize a certain performance (e.g. social welfare). Current mechanism design often assumes agents to be fully rational, neglecting the uncertainties in agents' decision-making processes. However, in many real-world situations, practical agents are observed to violate this assumption, causing incentive mechanisms to fail in achieving the two objectives. To overcome this problem, various bounded rationality models have been built to depict practical agents (Simon 1979; Ortega and Braun 2013; Wang and Tang 2015). Then, the robustness against bounded rationality is measured as to what extent incentive mechanisms can resist the uncertainties overlooked in the design phase (Cabrales and Serrano 2011; Rivas 2015; Liu et al. 2015; Shnayder, Frongillo, and Parkes 2016). It provides a quantitative indication for the practicability of incentive mechanisms.

However, there is still a large room for existing studies on quantitative robustness analysis to improve. Firstly, focusing only on a specific type of incentive mechanisms and bounded rationality models, their robustness formulations

become incompatible (Jurca and Faltings 2007; Liu et al. 2015). A general robustness formulation is thus needed to set a common ground for the comparison of different incentive mechanisms and to avoid reinventing the wheel. Secondly, most studies focus only on the incentive compatibility (IC) objective, but ignore the performance objective. However, the performance of an incentive mechanism is also very important. For example, if an auctioneer cannot gain enough revenue to cover the cost, there is no point for him to employ the auction. Thirdly, when computing robustness through Monte Carlo simulation, existing studies often set an arbitrarily large number of simulations. Without the guarantee of error rate, the reliability of their computation is uncertain.

In this paper, we propose a general framework to reliably compute the robustness of incentive mechanisms. More specifically, we first propose a general robustness formulation as the maximum bounded rationality level satisfying the conditions that all agents have the incentive to follow the desired strategies and the performance is acceptable by mechanism designers. Since the robustness is formulated using the expected incentives or performance, the empirical robustness computation requires us to balance between the exploration of more bounded rationality levels to more accurately identify the maximum and the exploitation of the current bounded rationality level to more reliably judge the conditions mentioned above. We thus develop a continuum-armed bandit solver to adaptively adjust the exploration and the simulation scheme according to accumulated simulation data. We also derive the error rate upper bound for our solver and prove it to be more efficient than existing methods. To further validate our solver, we compare with other methods in a testbed where accurate solutions are known. To illustrate the applicability of our framework, we apply it to evaluate the robustness of various popular mechanisms.

Related Work

Existing quantitative studies on the robustness of incentive mechanisms have focused only on one specific type of incentive mechanisms or bounded rationality models, so that different robustness formulations have been proposed. For example, targeting on the adverse effects of agents' belief error, Jurca and Faltings (2007) propose the ϵ -robustness which requires peer-prediction mechanisms always to keep agents reporting truthfully when agents' be-

lief error is smaller than ϵ . Considering irrational invaders, Liu *et al.* (2015) propose another completely different robustness formulation as the maximum proportion of invaders such that truthful reporting is still the best strategy for rational agents in the Moran process. Lately, considering agents' biased behaviors, Shnayder *et al.* (2016) provide yet another robustness formulation as the basin of attraction of the truthful reporting strategy. Due to the need to incorporate different bounded rationality models, these robustness formulations adopt different standards to compute the robustness and thus are incompatible. In contrast, our robustness formulation is general and can be readily adopted by researchers for their interested incentive mechanisms and bounded rationality models.

Furthermore, most studies focus only on testing whether agents are incentivized to take the desired strategies (Cabralés and Serrano 2011; Tumennasan 2013). However, the performance of an incentive mechanism is also very important. For example, combinatorial auctions are designed to allocate items among agents to maximize the social welfare. Agents' dominant strategy is always to report their true value about the items. Nevertheless, if agents with bounded rationality cannot accurately figure out their own true value (e.g. the type misreporting model (Rivas 2015)), no matter how the incentive is designed, the social welfare will still be affected. The extensive studies on the approximation ratio of incentive mechanisms also reflect the importance of considering performance (Alkalay-Houlihan and Vetta 2014). Our robustness formulation considers both incentive compatibility and performance, to accurately indicate the practicability of incentive mechanisms.

Besides, existing studies usually compute the robustness through running Monte Carlo simulations for an arbitrarily large number of times (Liu *et al.* 2015; Shnayder, Frongillo, and Parkes 2016). For example, in (Shnayder, Frongillo, and Parkes 2016), the basin of attraction in the multidimensional strategy space is measured through simply running the simulation for 100 times. There is no guarantee for the reliability of their computation. By contrast, in our framework, the simulation scheme is always adjusted with the error rate considered. Moreover, to balance the need to test more bounded rationality levels to compute the formulated robustness more accurately and the need to conduct more simulations to get more reliable simulation results, we model the computation as a continuum-armed bandit problem and develop a solver with theoretically guaranteed error rate upper bound and efficiency.

Robustness Evaluation Framework

Our proposed robustness evaluation framework consists of three layers, as shown in Fig. 1. The desired error rate and exploring depth are fed to the continuum-armed bandit solver (CABS) in the first layer as inputs. CABS then adaptively explores the space of agents' bounded rationality levels and develops efficient simulation schemes to exploit the selected bounded rationality levels. To carry out the simulation scheme, CABS needs to call the margin calculator in the second layer repeatedly. The margin calculator quantifies the effects of bounded rationality on incentive mecha-

nisms. The specific formulation of the margin calculator is determined by the adopted robustness formulation, and different robustness formulations lead to different margin functions. The margin function is computed using the stochastic samples of agent's utility and the performance of incentive mechanisms, which are generated by the agent-mechanism simulator in the third layer.

Bounded rationality depicts practical agents' decision-making process. When participating in a mechanism, agents decide their actions based on their types and the knowledge about the mechanism and other agents. Mechanism designers usually assume that: 1) agents accurately know their types; 2) agents' knowledge is correct; 3) agents' utility functions are accurately known by mechanism designers; 4) agents only adopt the utility-maximizing strategy. Under these assumptions, the designed mechanism can incentivize agents to take the desired strategies. However, practical agents may violate these assumptions, causing their actions to deviate from the desired strategies. In this paper, we use bounded rationality to denote the decision-making process which violates any of these four assumptions. It is thus not limited to the scope of the classic definition of bounded rationality which only considers deviation from utility maximization (Simon 1979).

Agent-Mechanism Simulator

The stochastic world model depicts agents' characteristics, such as preferences and correlations between each other. In our framework, we use a joint probability function $f(\theta)$ to denote the overall distribution of agents' types. Besides, we introduce a set of conditional probability functions $b(\theta_i|\theta_k)$ to depict the possible correlations between agents. Note that many mechanisms (e.g. VCG) are designed without considering the world model. These mechanisms actually need to work well in all possible world models. The world model is required even when these mechanisms are evaluated for mainly two reasons. Firstly, working well in a particular world model is a necessary condition for the success of these mechanisms in all possible world models. Secondly, checking all possible world models with an empirical approach is infeasible in computation. For simplicity, we limit our robustness evaluation to a certain world model which models agents' characteristics in a specific application.

In our framework, we assume there are n participating agents. These agents are required to be able to store the generated types θ , decide their actions a and compute their utility u . When designing our framework, we are not concerned with the details of bounded rationality models and agents' decision-making processes. Thereby, we abstract them as a mapping from agents' types θ , bounded rationality level γ and the desired strategies s^* to agents' actions a . Note that the bounded rationality level γ might be interpreted differently in different bounded rationality models. For example, in the partial rationality model (Wang and Tang 2015), γ denotes the probability of choosing the wrong strategy. In the logistic quantal response model (Ortega and Lee 2014), γ equals the level of error which quantifies the importance of utility. Without loss of generality, we require the bounded rationality level to be a value $\gamma \in [0, 1]$, where $\gamma = 0$ and

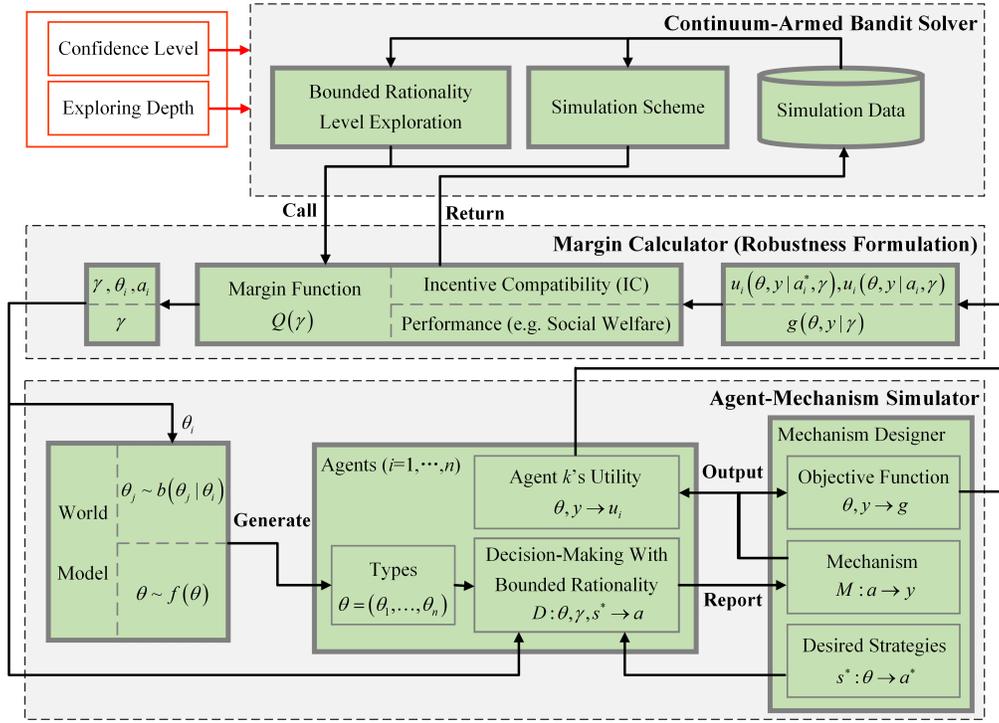


Figure 1: General robustness evaluation framework

$\gamma = 1$ denote the cases where agents are fully rational and irrational, respectively.

Mechanism designers need to know the objective function $G(\theta, y)$ and the desired strategies s^* before designing an incentive mechanism M . Thus, we use the mechanism designer module to store $G(\theta, y)$, s^* and M in our framework. Besides, when designing our framework, we treat the mechanism M as a black-box mapping from agents' actions a into a set of outputs y . This feature ensures the generality of our framework to incorporate different mechanisms.

Robustness Formulation

Robustness evaluates to what extent a mechanism can resist agents' bounded rationality. Mechanisms are usually designed to achieve two objectives: 1) incentivizing agents to follow the desired strategy s^* , i.e. achieving IC, and 2) maximizing the demanded objective function $G(\theta, y)$. Thus, we develop a general robustness formulation by measuring the effects of bounded rationality on these two objectives.

IC requires the mechanism to ensure that all agents can achieve utility maximization by taking the desired strategy s^* . Bounded rationality causes agents to deviate from s^* . However, the mechanism should at least not encourage agents to deviate from s^* . Thus, any agent k should not get a higher utility when deviating from s^* . Otherwise, the mechanism fails. Formally, we define the IC robustness R_I as

$$R_I = \min_{k \in \{1, \dots, n\}, \theta_k, a_k} R_a(k, \theta_k, a_k) \quad (1)$$

$$R_a(k, \theta_k, a_k) = \max\{\bar{\gamma} | \forall \gamma \in [0, \bar{\gamma}], \mathbb{E}Q_I(\gamma) = \mathbb{E}[u_k(\gamma, \theta_k, a_k^*) - u_k(\gamma, \theta_k, a_k)] \geq 0\}$$

Here, the IC margin function Q_I measures the remaining advantage of the desired strategy $a_k^* = s_k^*(\theta_k)$ over another available action a_k . R_a needs to be solved for every possible agent-type-action combination, and the IC robustness R_I is the minimal value of R_a . In computation, we can use the symmetry between agents to skip some combinations.

The demanded objective function $G(\theta, y)$ quantifies the benefits of using an incentive mechanism. For example, the classic VCG mechanism can maximize the social welfare of item allocation. However, if bounded rationality results in a lower social welfare than random allocation, it becomes meaningless to use VCG. Thereby, to ensure the effectiveness of a mechanism, the performance degradation caused by bounded rationality should be within an acceptable range. Formally, we define the performance robustness as

$$R_P = \max\{\bar{\gamma} | \forall \gamma \in [0, \bar{\gamma}], \mathbb{E}Q_P(\gamma) = \mathbb{E}[g(\gamma) - C_0] \geq 0\} \quad (2)$$

where $g(\gamma) = G(\theta, y)|_\gamma$ denotes the performance when the bounded rationality level is γ . C_0 denotes the minimum performance acceptable to mechanism designers. The performance margin function Q_P indicates the remaining benefits of a mechanism under the effects of bounded rationality.

If a mechanism works well with bounded rational agents, both the IC and performance should be ensured. Thus, we formulate the robustness of mechanism M as

$$R_M = \min\{R_I, R_P\} \quad (3)$$

In summary, our robustness formulation adopts a two-level structure to cover both IC and performance. At the micro-level of individual agents, the IC robustness keeps the incentive of mechanisms by requiring agents' utility-maximizing

strategies always to be s^* . This requirement follows the robustness studies of Jurca and Faltings on mechanisms with Bayesian-Nash incentive compatibility (Jurca and Faltings 2007). At the macro-level, our performance robustness maintains the benefits of adopting a mechanism by limiting the overall adverse effects of bounded rationality into an acceptable range. This setting shares the same idea as the extensive studies on the approximation ratio of incentive mechanisms (Alkalay-Houlihan and Vetta 2014). All in all, we develop the general robustness formulation by combining the separate robustness studies on IC and performance.

Continuum-Armed Bandit Solver

Solving the formulated robustness R_M relies on computing the IC robustness $R_a(k, \theta_k, a_k)$ and the performance robustness R_P . Since R_a and R_P have similar forms, we can uniformly formulate their computation as solving

$$R = \max\{\bar{\gamma} \in [0, 1] \mid \forall \gamma \in [0, \bar{\gamma}], \mathbb{E}Q(\gamma) \geq 0\} \quad (4)$$

where $Q(\gamma)$ is equal to either Q_I or Q_P . Correspondingly, R denotes R_a or R_P . In practice, most of the time, the margin function $Q(\gamma)$ gradually decreases with increasing γ . Hence, we can assume $Q(\gamma)$ to be continuous. Furthermore, we denote the minimal root of the margin equation $\mathbb{E}Q(\gamma) = 0$ as γ^* , namely $\gamma^* = \min\{\gamma \mid \mathbb{E}Q(\gamma) = 0\}$. In this case, the desired robustness in Equation 4 satisfies $R = \gamma^*$.

Given a solution interval $[0, \gamma_c]$, if the margin equation $\mathbb{E}Q(\gamma) = 0$ has only one root¹, we can convert empirically solving Equation 4 into a continuum-armed bandit problem. At each step, we choose an arm $\gamma \in [0, \gamma_c]$. Then, we can obtain a stochastic signal $Q(\gamma) \in [\bar{Q}, \underline{Q}]$ by calling the margin calculator for one time, where \bar{Q} and \underline{Q} denote the maximal and minimal values of Q . The reward for selecting γ is $-\mathbb{E}Q(\gamma)$, and we intend to maximize the accumulated rewards. Since selecting γ^* produces the highest rewards, we need to find γ^* as early as possible, which is also aligned with our objective to efficiently compute the formulated robustness. Besides, to measure the computation efficiency of bandits solvers, we define the regret $L(T)$ as

$$L(T) = \sum_{t=1}^T |\mathbb{E}Q(\gamma_t)| \quad (5)$$

where T denotes the number of simulation runs. Note that, since continuum-armed bandit solvers cannot reach the optimal arm within a finite number of simulation runs, we compute the regret instead of directly using the number of simulation runs to quantify the efficiency. In other words, the solver is expected to more efficiently approach the region with smaller $|\mathbb{E}Q(\gamma)|$, which corresponds to a lower regret.

There have been many solutions to the continuum-armed bandit problem (Jia and Mannor 2011; Kleinberg 2004). However, most of these solutions require the reward function to satisfy certain continuity assumptions (e.g. Lipschitz continuity). In our framework, these assumptions cannot be ensured. Thus, we present a new algorithm which only requires the reward function to be continuous. Firstly, we consider a simple multi-armed bandit problem with a finite set

¹If this assumption is violated, our solver computes the interval which contains all the roots of the margin equation.

Algorithm 1: Simulation Scheme-I

- Input:** A set of $m + 1$ arms, $\epsilon > 0$, and $\delta > 0$
- 1 Sample arms $2, \dots, m$ sequentially, until each arm has been simulated $\frac{(\bar{Q}-\underline{Q})^2}{2\epsilon^2} \log(\frac{m-1}{\delta})$ times;
 - 2 Let the simulation-average of arm i be \hat{Q}_i . Compute $j = \max\{s \mid \hat{Q}_{i < s} > \epsilon\}$ and $k = \min\{s \mid \hat{Q}_{k > s} < -\epsilon\}$.
 - 3 Output $j^* = j - 1$, $k^* = k + 1$ and $\min_{i \in \{2, \dots, m\}} |\hat{Q}_i|$.
-

Algorithm 2: Simulation Scheme-II

- Input:** A set of $m + 1$ arms and $\delta > 0$
- 1 Initialize $\epsilon = \bar{Q} - \underline{Q}$, $j^* = 1$, and $k^* = m + 1$.
 - 2 **while** $j^* < k^* - 1$ **or** $\epsilon \geq \epsilon_l$ **do**
 - 3 Run Algorithm 1, and update ϵ as $\min_i |\hat{Q}_i|$.
 - 4 Output j^* and k^* .
-

of arms $\{1, \dots, m + 1\}$. Besides, we assume that $\mathbb{E}Q_1 \geq 0$ and $\mathbb{E}Q_{m+1} \leq 0$ are known. Algorithm 1 is proposed to solve this problem with a probably approximately correct guarantee. It takes the error bound ϵ and the error rate δ as inputs. The outputs k^* and j^* satisfy the following property:

Proposition 1. With probability $1 - \delta$, $\gamma^* \in [\gamma_{k^*}, \gamma_{j^*}]$.

Proof. According to (Auer, Cesa-Bianchi, and Fischer 2002), we can have $Pr(\hat{Q}_i - \mathbb{E}Q_i \geq \epsilon) \leq \frac{\delta}{m-1}$ or $Pr(\hat{Q}_i - \mathbb{E}Q_i \leq -\epsilon) \leq \frac{\delta}{m-1}$. Hence, in Algorithm 1, $Pr(\mathbb{E}Q_{i \leq j^*} > 0) \geq 1 - \frac{\delta}{m-1}$, and $Pr(\mathbb{E}Q_{i \geq k^*} < 0) \geq 1 - \frac{\delta}{m-1}$. Since $\mathbb{E}Q(\gamma)$ is continuous and $\mathbb{E}Q(\gamma) = 0$ has only one root γ^* , we can conclude that $\mathbb{E}Q(\gamma \leq \gamma^*) \geq 0$ and $\mathbb{E}Q(\gamma \geq \gamma^*) \leq 0$. Therefore, $Pr(\gamma^* \in [\gamma_{k^*}, \gamma_{j^*}]) \geq (1 - \frac{\delta}{m-1})^{m-1} \geq 1 - \delta$. \square

Secondly, as the number of simulation runs increases, we need to decrease the error bound ϵ to ensure the obtained interval in Algorithm 1 to converge to the optimal arm γ^* . Existing solutions to the continuum-armed bandit problem usually update ϵ based on the assumed parameters of the reward function, for example, the Lipschitz constants. However, these parameters are not available in our framework. Therefore, we propose Algorithm 2 which adjusts ϵ according to the minimal value of $|\hat{Q}_i|$ (returned by Algorithm 1). Our objective is to greedily decrease $[j^*, k^*]$ to the utmost. Besides, the smaller ϵ is, the more simulation runs are needed. Since the computation capability is limited, we introduce ϵ_l as the lower bound of ϵ .

Lastly, we propose Algorithm 3 to update the set of arms. In each level of exploration, our algorithm generates new arms using bisection. This operation is widely adopted in existing solutions to the continuum-armed bandit problem (Bull 2015). It keeps improving the precision of searching for the optimal arm γ^* by repeatedly zooming in on the solution interval. In our framework, since γ^* must be in the

interval where the values of $\mathbb{E}Q(\gamma)$ at the two ends have different signs, we only conduct the “zooming in” operation at the two ends of the solution interval $[\gamma_1, \gamma_{-1}]$. Then, according to the simulation results, the algorithm decides whether to remove the end arms γ_1 and γ_{-1} to shrink the solution interval. Besides, in Algorithm 3, if A_n has only two elements (e.g. A_0), we can have $\gamma_1 = \gamma_{-2}$ and $\gamma_2 = \gamma_{-1}$. In this case, we can skip line 5 to simplify the computation.

Algorithm 3: Bounded Rationality Level Exploration

Input: $\delta > 0$ and the exploration depth N

- 1 Initialize $\delta_s = 1 - (1 - \delta)^{1/N}$, and $A_0 = \{0, \gamma_c\}$.
 - 2 **for** $n = 1, \dots, N$ **do**
 - 3 Sort all elements in A_{n-1} in ascending order.
 Denote the first two elements by γ_1 and γ_2 .
 Denote the last two elements by γ_{-2} and γ_{-1} .
 - 4 Run Algorithm 2 with $\frac{\delta_s}{2}$ and $\{\gamma_1, e_{12}, \gamma_2\}$ as inputs,
 where $e_{12} = \frac{\gamma_1 + \gamma_2}{2}$. If $k^* = 1$,
 $A_n = A_{n-1} \cup \{e_{12}\}$;
 otherwise, $A_n = (A_{n-1} - \{\gamma_1\}) \cup \{e_{12}\}$.
 - 5 Run Algorithm 2 with $\frac{\delta_s}{2}$ and $\{\gamma_{-1}, e_{-12}, \gamma_{-2}\}$ as inputs, where $e_{-12} = \frac{\gamma_{-1} + \gamma_{-2}}{2}$.
 If $k^* = 1$, $A_n = A_{n-1} \cup \{e_{-12}\}$;
 otherwise, $A_n = (A_{n-1} - \{\gamma_{-1}\}) \cup \{e_{-12}\}$.
 - 6 Output $\gamma_L = \min A_n$ and $\gamma_U = \max A_n$.
-

In summary, the proposed solver, CABS, takes the error rate δ and the exploration depth N as inputs. It outputs an interval $[\gamma_L, \gamma_U]$, and we can approximately calculate the robustness as $R \approx (\gamma_L + \gamma_U)/2$.

Theoretical Analysis

We perform theoretical analysis on the reliability and efficiency of our solver. First, the error rate is ensured by Proposition 2, revealing the high reliability of our framework.

Proposition 2. $Pr(R \notin [\gamma_L, \gamma_U]) \leq \delta$.

Proof. Proposition 1 provides the probability guarantee that $Pr(\gamma^* \in [\gamma_1(n), \gamma_{-1}(n)]) \geq (1 - \delta_s/2)^2 \geq 1 - \delta_s$ under the assumption: $\gamma^* \in [\gamma_1(n-1), \gamma_{-1}(n-1)]$. Here, $\gamma_1(n)$ and $\gamma_{-1}(n)$ denote the minimal and maximal values of A_n in Algorithm 3. Thus, we can obtain $Pr(\gamma_L \leq R \leq \gamma_U) \geq (1 - \delta_s)^N = 1 - \delta$, which concludes Proposition 2. \square

Then, the regret of our solver is analyzed in Proposition 3.

Proposition 3. If we set ϵ_l in Algorithm 2 as 0, with probability $1 - \delta$, the regret of CABS is $O(\sqrt{T})$.

Proof. Since the set of arms always has only three elements in Algorithm 3, the number of simulation runs for arm γ_i is $T_i = -\log(1 - \delta_t) \cdot \Delta^2 / (2\hat{Q}^2(\gamma_i))$, where $\delta_t = \delta_s/2$ and $\Delta = \bar{Q} - \underline{Q}$. Thus, the total number of simulation runs satisfies

$$T = \sum T_i = -\log(1 - \delta_t) \cdot \Delta^2 / 2 \cdot (\sum_{i=1}^K 1 / \hat{Q}^2(\gamma_i)) \quad (6)$$

where K denotes the total number of explored arms. Meanwhile, the regret of our solver can be calculated as

$$\begin{aligned} L &= \sum_{i=1}^K T_i \cdot |\mathbb{E}Q(\gamma_i)| \\ &= -\log(1 - \delta_t) \cdot \Delta^2 / 2 \cdot \sum_{i=1}^K |\mathbb{E}Q(\gamma_i)| / \hat{Q}^2(\gamma_i) \end{aligned} \quad (7)$$

To bridge $\hat{Q}(\gamma_i)$ and $\mathbb{E}Q(\gamma_i)$, we define a new variable x_i as

$$x_i = \hat{Q}(\gamma_i) - \mathbb{E}Q(\gamma_i) \quad (8)$$

Thus, we compute L with the following decomposition:

$$\begin{aligned} \sum_{i=1}^K |\mathbb{E}Q(\gamma_i)| / \hat{Q}^2(\gamma_i) &\leq \\ &\sum_{i=1}^K 1 / |\hat{Q}(\gamma_i)| + \sum_{i=1}^K |x_i| / \hat{Q}^2(\gamma_i) \end{aligned} \quad (9)$$

The first item in the right-hand side of Equation 9 satisfies

$$\sum_{i=1}^K 1 / |\hat{Q}(\gamma_i)| \leq \sqrt{K \sum_{i=1}^K 1 / \mathbb{E}Q(\gamma_i)^2} \quad (10)$$

Meanwhile, according to the Chernoff-Hoeffding bound (Auer, Cesa-Bianchi, and Fischer 2002), with probability $1 - \delta_t$, we have

$$x_i \geq -\sqrt{\frac{1}{2T_i} \log\left(\frac{1}{\delta_t}\right)} \quad \text{or} \quad x_i \leq \sqrt{\frac{1}{2T_i} \log\left(\frac{1}{\delta_t}\right)} \quad (11)$$

Hence, with probability $1 - \delta$, the second item in the right-hand side of the Equation 9 satisfies

$$\begin{aligned} \sum_{i=1}^K |x_i| / \hat{Q}^2(\gamma_i) &\leq \sum_{i=1}^K \sqrt{(1/2T_i) \cdot \log(1/\delta_t)} / \hat{Q}^2(\gamma_i) \\ &\leq C \cdot \sum_{i=1}^K 1 / |\hat{Q}(\gamma_i)| \\ &\leq C \cdot \sqrt{K \sum_{i=1}^K 1 / \mathbb{E}Q(\gamma_i)^2} \end{aligned} \quad (12)$$

where C is a finite constant. Besides, Equation 6 ensures

$$\sum_{i=1}^K 1 / \mathbb{E}Q(\gamma_i)^2 = T / C_\delta \quad (13)$$

where $C_\delta = -\log(1 - \delta_t) \cdot \Delta^2$. Thus, we can have $L \leq C_\delta \cdot (1 + C) \cdot \sqrt{K \cdot T / C_\delta}$, which concludes Proposition 3. \square

The lowest regret achieved by existing continuum-armed bandit algorithms is $O(\sqrt{T})$ (Bull 2015; Auer, Ortner, and Szepesvári 2007). However, these algorithm rely on a very strong assumption that the reward function has quadratic global maxima. By contrast, with a weak assumption that $\mathbb{E}Q(\gamma)$ is continuous, the regret of CABS also reaches $O(\sqrt{T})$. This reveals the great advantage of our solver in computing the robustness. In addition to the regret analysis, by assuming Algorithm 2 always reaches the error lower bound ϵ_l , we can know the worst time complexity of CABS is $O(N/\epsilon_l \cdot \log(2/\delta_s))$.

Experiments

We first validate the reliability and efficiency of our framework in a hypothetical testbed where accurate results are known, and then illustrate its generality and applicability on five popular peer prediction mechanisms and the widely-adopted procurement mechanisms in crowdsourcing.

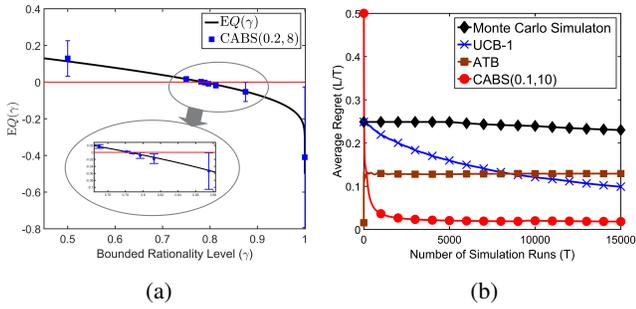


Figure 2: (a) Accurate $\mathbb{E}Q(\gamma)$ v.s. results of CABS; (b) Average regrets comparison of different algorithms

Hypothetical Testbed

For the hypothetical testbed, we consider a simple scenario where only one agent exists. Suppose agent’s type θ follows the uniform distribution $U[0.1, 10]$. Given θ , agent’s action a follows the beta distribution $Be(\theta, t)$, where $t = 1 - 2 \log(1 - \gamma)$. The performance measurement g is set as $g(\gamma) = a$, and the performance robustness R_P is calculated using our framework. In this case, $\mathbb{E}Q(\gamma)$ can be theoretically calculated as $1 - C_0 + \frac{t}{9.9} \log \frac{t+0.1}{t+10}$. Figure 2a shows the computation results of CABS when $C_0 = 0.5$. The explored bounded rationality levels and the computed averages are represented as squares. The solid line denotes the accurate value of $\mathbb{E}Q(\gamma)$. The error bounds computed with Algorithm 2 are marked with bars. We can observe that the error bar becomes shorter as $|\mathbb{E}Q|$ approaches 0. This reveals the adaptive strategy of CABS to minimize the number of simulations according to $\mathbb{E}Q$. As a matter of fact, this adaptive strategy is the foundation for the high efficiency of CABS.

Furthermore, we compare the average regrets of different algorithms in Figure 2b. UCB-1 is a famous bandit algorithm which requires a finite set of arms (Auer, Cesa-Bianchi, and Fischer 2002). To be consistent with our solver, we evenly divide $(0, 1]$ into 512 intervals and set the interval endpoints as arms. ATB is a newly proposed continuum-armed bandit algorithm (Bull 2015). It also repeatedly zooms in on the solution interval according to the simulation results. Monte Carlo simulation is widely adopted for robustness computation in the literature (Liu et al. 2015; Shnayder, Frongillo, and Parkes 2016). Given a finite set of arms, it sequentially carries out an arbitrarily large number of simulations on each arm. To ensure the computation accuracy, we set the needed simulation runs as 5000. From Figure 2b, we can conclude that the performance of CABS on regret is far better than the other three algorithms. Note that the average regret of ATB is almost unchanged from $T = 1000$. The reason for this observation is that the “zooming in” operation of ATB is very costly. ATB even cannot bisect $(0, 1]$ within 15,000 simulation runs. Besides, the initial average regret of CABS is high because we set γ_c as 2 in Algorithm 3, forcing CABS to test the worst case ($\gamma = 1$) at first.

Besides, we can measure the error rate of CABS as $Pr(R \notin [\gamma_L, \gamma_U])$. According to Proposition 2, the error

Table 1: Empirically measured error rate in 200 runs

C_0	0.5		0.6		0.7	
δ	0.99	0.9	0.99	0.9	0.99	0.9
Error Rate	0.02	0.005	0.035	0.005	0.075	0.0

rate of CABS is not larger than δ . To validate this point, we empirically compute the error rate for different values of C_0 by running CABS for 200 times. From the results listed in Table 1, we can see that the empirical error rate is far lower than δ . This is because the solution bound calculation in Algorithm 1 (line 2) and the error rate allocation in Algorithm 3 (line 1) are both very conservative. In the following experiments, we keep δ as 0.2 to ensure the reliability of the evaluation results.

Peer Prediction Mechanisms

Peer prediction mechanisms are widely adopted to elicit truthful information from self-interest agents. In the basic settings of peer prediction, each agent i privately receives a signal θ_i at first. All the signals are identically and independently conditioned on the phenomenon type h , meaning that all agents share the same conditional probability function $Pr(\theta|h)$. Then, according to the observed signal θ_i , agent i updates its belief about other agents’ signals $b_{i(-i)}(\theta_{-i}|\theta_i)$. Based on θ_i and $b_{i(-i)}$, agent i decides its report (action) a_i and obtains the reward u_i from the mechanism. Different peer prediction mechanisms lead to different mappings from $a = (a_1, \dots, a_n)$ to $u = (u_1, \dots, u_n)$. We consider five popular and distinct peer prediction mechanisms:

- 1) **OA** (Von Ahn and Dabbish 2004). For agent i , a reference agent j is selected. If these two agents have the same report, $u_i = 1$. Otherwise, $u_i = 0$.
- 2) **MRZ** (Miller, Resnick, and Zeckhauser 2005). For agent i , a reference agent j is selected. u_i is determined by $W(Pr(a_j|a_i))$, where $W(\cdot)$ denotes the scoring function. Three best known MRZ mechanisms are developed using quadratic, spherical and logarithmic scoring functions, respectively.
- 3) **JF09** (Jurca, Faltings, and others 2009). For agent i , there are three or more reference agents. u_i can be computed according to Proposition 4.2 in (Jurca, Faltings, and others 2009).
- 4) **RF15** (Radanovic and Faltings 2015). For agent i , there is a group of similar peers. Given the fractions $z_p(a_i)$ and $z_g(a_i)$ of peers and global population respectively reporting the same as agent i , $u_i = \log(z_p/z_g)$.
- 5) **DGMS** (Shnayder et al. 2016). Agents i and j have many tasks. They also have at least one overlapping tasks. The rewards for overlapping tasks are computed with the reporting patterns on the non-overlapping tasks considered. u_i is the sum of rewards across all tasks.

To reconcile the requirements of different mechanisms, we focus on a basic scenario where h only has two states, 0 and 1. Correspondingly, θ also only has two states, and $Pr(\theta = 0|h = 0) = 0.8$, $Pr(\theta = 1|h = 1) = 0.8$.

Table 2: The decision conditions used by JF09

$\Pr(Q_L)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Cond A	-	-	-	-	-	+	+	-	-
Cond B	-	-	+	+	-	-	-	-	-

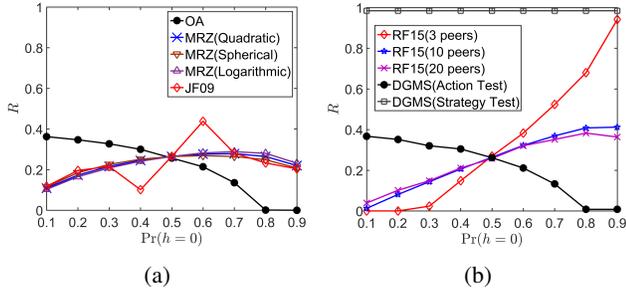


Figure 3: Robustness of peer prediction mechanisms

For bounded rationality models, we consider the malicious bad-mouthing agents that intentionally report 0 (Du, Huang, and others 2013). The robustness reveals the maximum ratio of bad-mouthing agents at which truthful reporting is still encouraged.² Besides, we change $\Pr(h = 0)$ from 0.1 to 0.9 in our experiment to generate different world models. This makes our experiments more informative than (Shnayder, Frongillo, and Parkes 2016) which only considers $\Pr(h = 0) = 0.5$. This also reveals the convenience of our framework to analyze different cases.

Figure 3 shows the robustness comparison between the aforementioned five peer prediction mechanisms. For OA, if $\Pr(h = 0) \geq 0.8$, then $\Pr(\theta_j = 1 | \theta_i = 1) \leq 0.5$ and the mechanism will totally lose its effectiveness. This observation is consistent with the fact that the simple OA cannot always ensure IC. For MRZ, the three rules have similar robustness performance. By giving high incentives to rare signals, they obtain far better performance than OA for large $\Pr(h = 0)$ cases. Nevertheless, they also sacrifice a little performance in small $\Pr(h = 0)$ cases. JF09 utilizes two conditions to decide the payment, and their values are listed in Table 2. When the two conditions are kept negative, JF09 has similar robustness as MRZs. The changing of two conditions will bring fluctuation in robustness. Sometimes, the robustness of JF09 can even be worse than MRZs, which is consistent with the analysis in (Shnayder, Frongillo, and Parkes 2016). Through introducing new information from global population, RF15 can enhance the incentives for agents to report the truthful information for rare signals³. However, if the number of peers is smaller, the performance of RF15 may become extremely poor, which confirms the conclusion in (Shnayder, Frongillo, and Parkes 2016). For DGMS, if we evaluate its robustness by comparing the action on shared tasks, the reward function defined in (Shnayder et al. 2016) is similar as OA, which leads to a similar robustness as OA in Figure 3. If the robustness is tested

²Peer prediction mechanisms only focus on incentivizing agents to report truthfully. Thus, R_P is not considered.

³When $\Pr(h = 0)$ is large, $\theta = 1$ will become the minority.

via comparing the report strategy shared across all tasks, the robustness is always very good. This is because the strategy information of agents is introduced into the reward function in DGMS. The above analysis reveals that the evaluation results of our framework are reasonable, which verifies the applicability of our robustness formulation and computation to practical incentive mechanisms. Besides, there is an interesting phenomenon that all mechanisms show similar robustness performance for $\Pr(h = 0) = 0.5$. This is because our symmetrical settings of $\Pr(\theta|h)$ ensure h and θ both have the same probability for 0 and 1 at this middle point.

Procurement in Crowdsourcing

In crowdsourcing, given a limited budget, how to procure an enough number of workers is usually a challenging problem (Singer 2010; Singla and Krause 2013). In the literature, two kinds of procurement mechanisms have been proposed to solve this problem: the procurement auction requires workers (agents) to report their desired wages, and the mechanism will decide the wage according to these reports (e.g. BFM (Singer 2010)); the posted-price auction requires the mechanism to provide a take-it-or-leave-it wage offer to workers, and workers only need to decide whether to accept this wage (e.g. BP-UCB (Singla and Krause 2013)). The posted-price auction minimizes workers' thinking efforts, which prevents possible errors in agents' decision-making process. However, to recruit more workers with the given budget, the mechanism needs to learn workers' preferences through continuous interactions with them. This learning process may make the wage too high or too low at the beginning, which can lead to a loss in the number of employees. By contrast, the procurement auction collects full information, and can thus achieve better performance. It relies on workers' truthful reports, and truthful reporting is always the dominant strategy for rational workers.

In the literature, the difference between these two approaches is only qualitatively discussed as above. From mechanism designers' point of view, if agents are fully rational, there is no point to utilize BP-UCB which will necessarily cause less workers to be enrolled. Actually, an implicit benefit of minimizing agents' thinking efforts is to make the mechanism more robust against agents' bounded rationality. Thus, we use our formulation to study robustness of BFM, which can provide a quantitative evaluation to help decide which mechanism to choose. This experiment illustrates the practical usability of our robustness formulation, which cannot be achieved by existing quantitative robustness studies. Specifically, we assume workers' true desired wage θ follows $Be(\psi_1, \psi_2)$. Besides, worker's bounded rational report $a = \theta + (1 - \theta) \cdot \delta$, where $\delta \sim Be(1, -1.5 \log(\gamma))$ and $\gamma \in [0, 1]$ can be regarded as workers' bounded rational level⁴. Moreover, since truthful reporting is always the dominant strategy in BFM, $R_I \equiv 1$. This is because dominant strategy mechanisms need to ensure that the desired strategy $a_i^* = s_i^*(\theta_i)$ has advantages over another available action a_i no matter what actions is taken by other

⁴Here, we use a simple model for illustration. More practical decision-making models can be utilized in further studies.

Table 3: The Robustness of BFM Against Biased Workers

n	300	350	400	450	500	550
$\theta \sim Be(0.5, 0.5)$	0.89	0.77	0.63	0.55	0.53	0.43
$\theta \sim Be(1.0, 1.0)$	1.00	1.00	0.90	0.80	0.72	0.70
$\theta \sim Be(2.0, 2.0)$	1.00	1.00	1.00	1.00	1.00	0.97

agents, namely $u_i(\theta, y|a_i^*, a_{-i}) \geq u_i(\theta, y|a_i, a_{-i})$. Thus, $u_i(\theta, y|a_i^*, \gamma) \geq u_i(\theta, y|a_i, \gamma)$ always holds in Equation 1, which leads to $R_I \equiv 1$. In this case, we only focus on R_P . The performance measurement g is set as the number of recruited workers, and the performance of BP-UCB in the same settings is employed as C_0 . The robustness measurement reveals the maximum bounded rationality level of workers at which BFM performs better than BP-UCB.

Table 3 shows the robustness evaluation results for BFM in different parameter settings. Here, n denotes the number of participating workers, and the total budget is set as 75. From the table, we can find that the robustness of BFM degrades as the number of workers increases. Actually, the performance of BFM changes a little when more workers come in. The main reason for this phenomenon lies in the increase of the performance of BP-UCB which is used as C_0 . In BP-UCB, the learning process keeps the same for a fixed θ distribution. Thus, more workers can make the adverse effects of the learning process less obvious, resulting in a better overall performance. Meanwhile, it is an obvious phenomenon that the robustness increases when the distribution of θ changes from $Be(0.5, 0.5)$ to $Be(2.0, 2.0)$. Similar as the previous phenomenon, this is also caused by the variation in C_0 . These reasonable evaluation results validate the soundness of our robustness formulation.

Conclusions

Robustness against bounded rationality indicates the practicability of an incentive mechanism. To develop a general robustness evaluation framework, we first propose a general robustness formulation which is applicable for different types of incentive mechanisms and bounded rationality models. Then, an efficient continuum-armed bandit solver is built to compute the formulated robustness with the theoretically guaranteed error rate upper bound. Experimental results validate the error rate upper bound and the advantage of our solver on efficiency. The generality and soundness of our robustness formulation and solver are illustrated by applying our framework to five peer prediction mechanisms and the procurement mechanisms in crowdsourcing. Currently, our framework cannot efficiently handle the case where the bounded rationality leads to multiple possible actions at one time. In future, we will investigate optimization methods to automatically look for the worst action.

References

Alkalay-Houlihan, C., and Vetta, A. 2014. False-name bidding and economic efficiency in combinatorial auctions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 538–544.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Auer, P.; Ortner, R.; and Szepesvári, C. 2007. Improved rates for the stochastic continuum-armed bandit problem. In *Proceedings of the Twentieth Annual Conference on Learning Theory*, 454–468.

Bull, A. D. 2015. Adaptive-treed bandits. *Bernoulli* 21(4):2289–2307.

Cabreres, A., and Serrano, R. 2011. Implementation in adaptive better-response dynamics: Towards a general theory of bounded rationality in mechanisms. *Games and Economic Behavior* 73(2):360–374.

Du, N.; Huang, H.; et al. 2013. Can online trading survive bad-mouthing? an experimental investigation. *Decision Support Systems* 56:419–426.

Jia, Y. Y., and Mannor, S. 2011. Unimodal bandits. In *Proceedings of the Twenty-Eighth International Conference on International Conference on Machine Learning*, 41–48.

Jurca, R., and Faltings, B. 2007. Robust incentive-compatible feedback payments. In *Agent-Mediated Electronic Commerce. Automated Negotiation and Strategy Design for Electronic Markets*. Springer. 204–218.

Jurca, R.; Faltings, B.; et al. 2009. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research* 34(1):209.

Kleinberg, R. D. 2004. Nearly tight bounds for the continuum-armed bandit problem. In *Proceedings of the Seventeenth International Conference on Neural Information Processing Systems*, 697–704.

Liu, Y.; Zhang, J.; An, B.; and Sen, S. 2015. A simulation framework for measuring robustness of incentive mechanisms and its implementation in reputation systems. *Autonomous Agents and Multi-Agent Systems* 1–20.

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51(9):1359–1373.

Ortega, P. A., and Braun, D. A. 2013. Thermodynamics as a theory of decision-making with information-processing costs. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 469.

Ortega, P. A., and Lee, D. D. 2014. An adversarial interpretation of information-theoretic bounded rationality. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2483–2489.

Radanovic, G., and Faltings, B. 2015. Incentive schemes for participatory sensing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent System*, 1081–1089.

Rivas, J. 2015. Mechanism design and bounded rationality: The case of type misreporting. *Mathematical Social Sciences* 78:6–13.

Shnayder, V.; Agarwal, A.; Frongillo, R.; and Parkes, D. C. 2016. Informed truthfulness in multi-task peer prediction.

In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 179–196.

Shnayder, V.; Frongillo, R. M.; and Parkes, D. C. 2016. Measuring performance of peer prediction mechanisms using replicator dynamics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2611–2617.

Simon, H. A. 1979. Rational decision making in business organizations. *The American economic review* 69(4):493–513.

Singer, Y. 2010. Budget feasible mechanisms. In *Proceedings of the Fifty-First Annual Symposium on Foundations of Computer Science*, 765–774.

Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the Twenty-Second International Conference on World Wide Web*, 1167–1178.

Tumennasan, N. 2013. To err is human: Implementation in quantal response equilibria. *Games and Economic Behavior* 77(1):138 – 152.

Von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 319–326.

Wang, Z., and Tang, P. 2015. Optimal auctions for partially rational bidders. In *Proceedings of the Twenty-Fourth International Conference on Artificial Intelligence*, 118–124.