Socially-Optimal Design of Service Exchange Platforms with Imperfect Monitoring

YUANZHANG XIAO and MIHAELA VAN DER SCHAAR, University of California, Los Angeles

We study the design of service exchange platforms in which long-lived anonymous users exchange services with each other. The users are randomly and repeatedly matched into pairs of clients and servers, and each server can choose to provide high-quality or low-quality services to the client with whom it is matched. Since the users are anonymous and incur high costs (e.g., exert high effort) in providing high-quality services, it is crucial that the platform incentivizes users to provide high-quality services. Rating mechanisms have been shown to work effectively as incentive schemes in such platforms. A rating mechanism labels each user by a rating, which summarizes the user's past behaviors, recommends a desirable behavior to each server (e.g., provide higher-quality services for clients with higher ratings), and updates each server's rating based on the recommendation and its client's report on the service quality. Based on this recommendation, a low-rating user is less likely to obtain high-quality services, thereby providing users with incentives to obtain high ratings by providing high-quality services.

However, if monitoring or reporting is imperfect—clients do not perfectly assess the quality or the reports are lost—a user's rating may not be updated correctly. In the presence of such errors, existing rating mechanisms cannot achieve the social optimum. In this article, we propose the first rating mechanism that does achieve the social optimum, even in the presence of monitoring or reporting errors. On one hand, the socially-optimal rating mechanism needs to be complicated enough, because the optimal recommended behavior depends not only on the current rating distribution, but also (necessarily) on the history of past rating distributions in the platform. On the other hand, we prove that the social optimum can be achieved by "simple" rating mechanisms that use binary rating labels and a small set of (three) recommended behaviors. We provide design guidelines of socially-optimal rating mechanisms and a low-complexity online algorithm for the rating mechanism to determine the optimal recommended behavior.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General; J.4 [**Computer Applications**]: Social and Behavior Sciences—*Economics*

General Terms: Design, Economics, Theory, Human Factors

Additional Key Words and Phrases: Rating mechanism, incentive schemes, imperfect monitoring

ACM Reference Format:
Yuanzhang Xiao and Mihaela van der Schaar. 2015. Socially-optimal design of service exchange platforms with imperfect monitoring. ACM Trans. Econom. Comput. 3, 4, Article 25 (July 2015), 25 pages.
DOI: http://dx.doi.org/10.1145/2785627

1. INTRODUCTION

Service exchange platforms have proliferated as the medium that allows users to exchange valuable services. For instance, emerging new service exchange platforms include crowdsourcing systems (e.g., Amazon Mechanical Turk and CrowdSource) in

This work is supported by the National Science Foundation, under grant CCF-1218136.

This work was done when Yuanzhang Xiao was a Ph.D. student at the Electrical Engineering Department, University of California, Los Angeles. Y. Zhang is currently affiliated with Northwestern University.

Authors' addresses: Y. Xiao (corresponding author), Department of Electrical Engineering and Computer Science, Northwestern University; email: xyz.xiao@gmail.com; M. van der Schaar, Electrical Engineering Department, University of California, Los Angeles; email: mihaela@ee.ucla.edu.

DOI: http://dx.doi.org/10.1145/2785627

which the users exchange labor [Kittur et al. 2008; Zhang et al. 2014; Zhang and van der Schaar 2012b], online question-and-answer websites (e.g., Yahoo! Answers and Quora) in which the users exchange knowledge [Zhang et al. 2014; Zhang and van der Schaar 2012b], peer-to-peer (P2P) networks in which the users exchange files/packets [Zhang and van der Schaar 2012a; Blanc et al. 2005; Feldman et al. 2004], and online trading platforms (e.g., eBay) in which the users exchange goods [Dellarocas 2005]. In a typical service exchange platform, a user plays a dual role: as a *client*, who requests services, and as a *server*, who chooses to provide high-quality or low-quality services. Common features of many service exchange platforms are that the user population is large and users are anonymous. In other words, each user interacts with a randomly-matched partner without knowing its partner's identity (However, the platform does know the identify of the interacting users.). The absence of a fixed partner and the anonymity of the users create incentive problems—namely, the users tend to "free-ride" (i.e., receive high-quality services from others as a client, while providing low-quality services as a server). In addition, a user generally may not be able to perfectly monitor[1] its partner's action, which makes it even harder to incentivize the users to provide high-quality services.

An important class of incentive mechanisms for service exchange platforms are the rating mechanisms[2] in which each user is labeled with a rating based on its past behaviors in the system [Zhang et al. 2014; Zhang and van der Schaar 2012a, 2012b; Blanc et al. 2005; Feldman et al. 2004; Kandori 1992; Okuno-Fujiwara and Postlewaite 1995; Dal Bó 2007; Hasker 2007; Takahashi 2010; Deb 2012; Ellison 1994]. A rating mechanism consists of a rating update rule and a recommended strategy.[3] The recommended strategy specifies what the desirable behavior is under the current system state (e.g., the current rating profile of the users or the current rating distribution). For example, the rating mechanism may recommend providing high-quality services for all the users when the majority of users have high ratings, while recommending providing high-quality services only to high-rating users when the majority have low ratings. Then, based on each client's report on the quality of service, the rating mechanism revises each server's rating according to the rating update rule. Generally speaking, the ratings of the users who comply with (resp. deviate from) the recommended behaviors go up (resp. down). Hence, each user's rating summarizes its past behavior in the system. By keeping track of all the users' ratings and recommending them to reward (resp. punish) the users with high (resp. low) ratings, the rating mechanism gives incentives to the users to obtain high ratings by rewarding them indirectly, through recommending other users to provide them with high-quality services.

Existing rating mechanisms have been shown to work well when monitoring and reporting are perfect. However, when monitoring and reporting are subject to errors, existing rating mechanisms cannot achieve the social optimum. The errors, which are often encountered in practice, may arise either from the client's own incapability of accurate assessment (e.g., the client, who wants to translate some sentences into a foreign language, cannot accurately evaluate the server's translation), or from some system errors (e.g., the client's report on the server's service quality is missing due to

[1] The monitoring discussed throughout this article is a user's observation on its current partner's actions. Each user knows nothing about the ongoing interactions among the other pairs of users.

[2] Note that the rating mechanisms studied here focus on dealing with moral hazard problems, namely, the server's quality of service is not perfectly observable. They are different from the rating mechanisms dealing with adverse selection problems, namely, the problems of identifying the users' types. See Dellarocas [2005, Sec. I] for detailed discussions on these two classes of rating mechanisms.

[3] Different terminologies have been used in the existing literature. For example, Dellarocas [2005] and Kandori [1992] use "reputation" for "rating," and Kandori [1992] uses "social norm" for "recommended strategy,"

network errors).[4] In the presence of errors, the server's rating may be wrongly updated. Hence, even if the users follow the recommended desirable behavior, the platform may still fall into some "bad" states in which many users have low ratings due to erroneous rating updates. In these bad states, the users with low ratings receive low-quality services, resulting in large performance loss compared to the social optimum. This performance loss in the bad states is the major reason for the inefficiency of the existing rating mechanisms.

In this article, we propose the first rating mechanisms that can achieve the social optimum even under imperfect monitoring. A key feature of the proposed rating mechanism is the *nonstationary* recommended strategy, which recommends different behaviors under the same system state, depending on when this state occurs (e.g., the rating mechanism may not always recommend punishing users with low ratings in the bad states). Note, importantly, that the rating mechanism does not just randomize over different behaviors with a fixed probability in a state. Instead, it recommends different behaviors in the current state based on the history of past states. We design the recommended strategy carefully, such that the punishments happen frequently enough to provide sufficient incentives for the users, but not too frequently to reduce the performance loss incurred in the bad states. The more patient the users are (i.e. the larger discount factor they have), the less frequent the punishments. As a result, the designed rating mechanism can asymptotically achieve the social optimum as users become increasingly patient (i.e., as the discount factor approaches 1). This is in contrast with the existing rating mechanisms with *stationary* recommended strategies, whose performance loss does not vanish even as the users' patience increases. Another key feature of the proposed rating mechanism is the use of differential punishments that punish users with different ratings differently. In Section 4, we show that the absence of any one of these two features in our mechanism will result in performance loss that does not vanish even when the users are arbitrarily patient.

We prove that the social optimum can be achieved by simple rating mechanisms which assign binary ratings to the users and recommend a small set of three recommended behaviors. We provide design guidelines of the rating update rules in socially-optimal rating mechanisms and a low-complexity online algorithm to construct the nonstationary recommended strategies. The algorithm essentially solves a system of two linear equations with two variables in each period and can be implemented with a memory of a fixed size (although by the definition of nonstationary strategies, it appears that we may need a memory growing with time to store the history of past states), because we can appropriately summarize the history of past states (by the solution to the preceding linear equations).

The rest of the article is organized as follows. In Section 2, we discuss the differences between our work and related works. In Section 3, we describe the model of service exchange systems with rating mechanisms. Then we design the optimal rating mechanisms in Section 5. Simulation results in Section 6 demonstrate the performance improvement of the proposed rating mechanism. Finally, Section 7 concludes.

2. RELATED WORKS

2.1. Related Works on Rating Protocols

Rating mechanisms were originally proposed by Kandori [1992] for a large anonymous society in which users are repeatedly randomly matched to play the Prisoners'

[4] Note that the errors in this article are not caused by the strategic behaviors of the users. In other words, the clients report the service quality truthfully and do not misreport intentionally to manipulate the rating mechanism for their own interests. If the clients may report strategically, the mechanism can let the platform to assess the service quality (still, with errors) to avoid strategic reporting.

dilemma game. Assuming perfect monitoring, Kandori [1992] proposes a simple rating mechanism that can achieve the social optimum: any user who has defected will be assigned with the lowest rating forever and will be punished by its future partners. Subsequent research has been focusing on extending the results to more general games [Okuno-Fujiwara and Postlewaite 1995; Dal Bó 2007; Hasker 2007; Deb 2012], or on discovering alternative mechanisms (e.g., Takahashi [2010] shows that cooperation can be sustained if each user can observe its partner's past actions). However, all these works assume perfect monitoring and are aimed at dealing with the incentive problems caused by the anonymity of users and the lack of fixed partnership; they do not study the impact of imperfect monitoring. Under imperfect observation/reporting, the system will collapse under their rating mechanisms, because all the users will eventually end up with low ratings forever due to errors.

Some works [Zhang et al. 2014; Zhang and van der Schaar 2012a; Ellison 1994] assume imperfect monitoring but focus on the limited case when monitoring tends to be perfect. The conclusion of these works is that social optimum can be achieved in the limited case when monitoring becomes "almost perfect" (i.e., when the rating update error goes to zero).

Only a few works [Blanc et al. 2005; Feldman et al. 2004; Dellarocas 2005] analyze rating mechanisms under imperfect monitoring with *fixed nonzero* monitoring errors. For a variety of rating mechanisms studied [Blanc et al. 2005; Feldman et al. 2004; Dellarocas 2005], the performance loss with respect to social optimum is quantified in terms of the rating update error. These results confirm that existing rating mechanisms suffer from (severe) performance loss under rating update errors. Note that the model in Dellarocas [2005] is fundamentally different than ours. In Dellarocas [2005], there is only a single long-lived seller (server), while all the buyers (clients) are short lived. Under this model, it is shown that the rating mechanism is bounded away from social optimum even when nonstationary strategies are used. In contrast, we show that under our model with long-lived servers and clients, we can achieve social optimum by nonstationary strategies with differential punishments. In the following, we discuss the intuition of how to achieve social optimum under our model.

There are two sources of inefficiency. One source of inefficiency comes from the stationary recommended strategies, which recommends the same behavior under the same state. As we have discussed earlier, the inefficiency of the existing rating mechanisms comes from the punishments triggered in the "bad" states. Specifically, to give incentives for the users to provide high-quality services, the rating mechanism must punish the low-rating users under certain rating distributions (i.e., under certain "bad" states). When the users are punished (i.e., they are provided with low-quality services), the average payoffs in these states are far below the social optimum. In the presence of rating update errors, the bad states happen with a probability bounded above zero (the lower bound depends only on the rating update error). As a result, the low payoffs occur with a frequency bounded above zero, which incurs an efficiency loss that cannot vanish unless the rating update error goes to zero.

Another source of inefficiency is the lack of differential punishments. As will be proved in Section 4, rating mechanisms with no differential punishment have performance loss even when nonstationary recommended strategies are used.

This article is the first to propose a class of rating mechanisms that achieve the social optimum even when update errors do not tend to zero. Our mechanisms rely on (explicitly-constructed) *nonstationary* strategies with differential punishments. The key intuitions of why the proposed mechanism achieves social optimum are as follows. First, nonstationary strategies punish the users in the bad states only when necessary, depending on the history of past states. In this way, nonstationary strategies can lower the frequency of punishment in the bad states to a level just enough to provide sufficient incentives for the users to provide high-quality services. In addition, differential punishment further reduces the loss in social welfare by transferring payoffs from low-rating users to high-rating users, instead of lowering everyone's payoff with nondifferential punishment.

2.2. Related Works in Game Theory Literature

Our results are related to folk theorem results for repeated games [Fudenberg et al. 1994] and stochastic games [Hörner et al. 2011]. However, these existing folk theorem results [Fudenberg et al. 1994; Hörner et al. 2011] cannot be directly applied to our model. First, the results in Fudenberg et al. [1994] are derived for repeated games in which every stage game is the same. Our system is modeled as a stochastic game in which the stage games may be different because of the rating distributions.

Second, there do exist folk theorems for stochastic games [Hörner et al. 2011], but they also do not apply to our model. The folk theorems [Hörner et al. 2011] apply to standard stochastic games in which the state must satisfy the following properties: (1) the state, together with the action profile, uniquely determines the stage-game payoff, and (2) the state is known to all the users. In our model, since each user's stage game payoff depends on its own rating, each user's rating must be included in the state and be known to all the users. In other words, if we model the system as a standard stochastic game in order to apply the folk theorems, we need to define the state as the rating profile of all the users. However, in our model, the players do not know the full rating profile, but only know the rating distribution. Hence, the strategy can use only the information of rating distributions.[5] Whether such strategies can achieve the social optimum is not known according to the folk theorems; we need to prove the existence of socially optimal strategies that use only the information of rating distributions.

In addition, our results are fundamentally different from the folk theorem results [Fudenberg et al. 1994; Hörner et al. 2011] in nature. First, Fudenberg et al. [1994] and Hörner et al. [2011] focus on the limit case when the discount factor goes to one, which is not realistic because the users are not sufficiently patient. More importantly, their results are not constructive. They focus on *what* payoff profiles are achievable, but cannot show *how* to achieve those payoff profiles. They do not determine a lower bound on discount factors that admit equilibrium strategy profiles yielding the target payoff profile and hence cannot construct equilibrium strategy profiles. By contrast, we do determine a lower bound on discount factors that admit equilibrium strategy profiles yielding the target payoff profile, and do construct equilibrium strategy profiles.

2.3. Related Mathematical Frameworks

Rating mechanisms with stationary recommended strategies can be designed by extending Markov decision processes (MDPs) in two important and nontrivial ways [Zhang et al. 2014; Zhang and van der Schaar 2012a]: (1) since there are multiple users, the value of each state is a vector of all the users' values, instead of a scalar in standard MDPs, and (2) the incentive compatibility constraints of self-interested users need to be fulfilled (e.g., the values of "good" states, in which most users have high ratings, should be sufficiently larger than those of "bad" states, such that users are incentivized to obtain high ratings), while standard MDPs do not impose such constraints.

[5] We insist on restricting to strategies that depend only on the history of rating distributions, because in practice, (1) the platform may not publish the full rating profile due to informational and privacy constraints, and (2) even if the platform does publish such information, it is impractical to assume that the users can keep track of it.

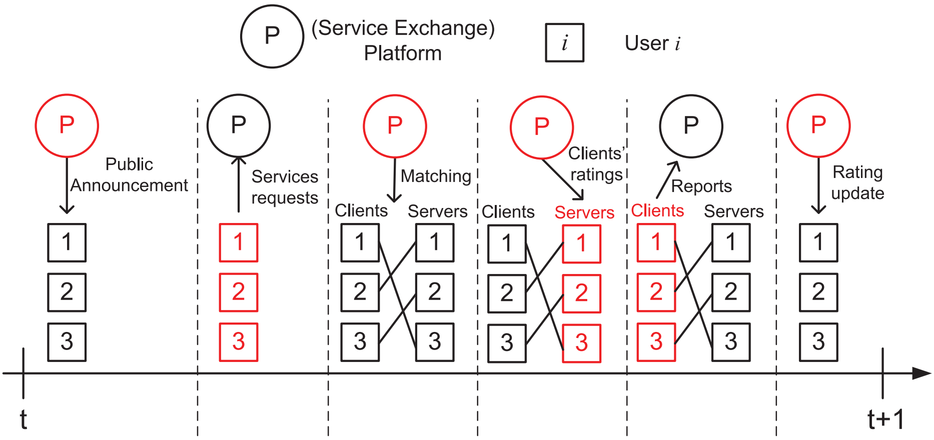


Fig. 1. Illustration of the rating mechanism in one period.

In this article, we make a significant step forward with respect to the state-of-the-art rating mechanisms with stationary strategies: we design rating mechanisms where the recommended strategies can be nonstationary. The proposed design leads to significant performance improvements but is also significantly more challenging from a theoretical perspective. The key challenge is that nonstationary strategies may choose different actions under the same state, resulting in possibly different current payoffs in the same state. Hence, the value function under nonstationary strategies are *set-valued*, which significantly complicates the analysis, compared to *single-valued* value functions under stationary strategies.[6]

The mathematical framework of analyzing nonstationary strategies with set-valued value functions is proposed as a theory of self-generating sets in Abreu et al. [1990]. It was widely used in game theory to prove folk theorems in repeated games [Fudenberg et al. 1994] and stochastic games [Hörner et al. 2011]. We have discussed our differences from the folk theorem results [Fudenberg et al. 1994; Hörner et al. 2011] in the previous subsection.

3. SYSTEM MODEL AND PROBLEM FORMULATION

3.1. System Model

3.1.1. The Rating Mechanism. We consider a service exchange platform with a set of N users, denoted by $\mathcal{N} = \{1, \ldots, N\}$. Each user can provide some services (e.g., data in P2P networks, labor in Amazon Mechanic Turk) valuable to the other users. The rating mechanism assigns each user i a binary label $\theta_i \in \Theta \triangleq \{0, 1\}$, and keeps records of the *rating profile* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$. Since the users usually stay in the platform for a long period of time, we divide time into periods indexed by $t = 0, 1, 2, \ldots$. In each period, the rating mechanism operates as illustrated in Figure 1, which can be roughly described as follows.

—Each user requests services as a *client*.

—Each user (as a *server*) is matched to another user (its client) based on a matching rule.

[6] In randomized stationary strategies, although different actions may be taken in the same state *after randomization*, the probability of actions chosen is fixed. In the Bellman equation, we need to use the *expected payoff before randomization* which is fixed in the same state, instead of the realized payoffs after randomization. Hence, the value function is still single valued.

—Each server chooses to provide high-quality or low-quality services.
—Each client reports its assessment of the service quality to the rating mechanism who will update the server's rating based on the report.

Next, we describe the key components in the rating mechanism in details.

Public Announcement. At the beginning of each period, the platform makes a public announcement to the users. The public announcement includes the rating distribution and the recommended plan in this period. The *rating distribution* indicates how many users have rating 1 and rating 0, respectively. Denote the rating distribution by $\boldsymbol{s}(\boldsymbol{\theta}) = (s_0(\boldsymbol{\theta}), s_1(\boldsymbol{\theta}))$, where $s_1(\boldsymbol{\theta}) = \sum_{i\in\mathcal{N}} \theta_i$ is the number of users with rating 1, and $s_0(\boldsymbol{\theta}) = N - s_1(\boldsymbol{\theta})$ is the number of users with rating 0. Denote the set of all possible rating distributions by S. Note that the platform does not disclose the rating profile $\boldsymbol{\theta}$ for privacy concerns. The platform also recommends a desired behavior in this period, called *recommended plan*. The recommended plan is a contingent plan of which service quality the server should choose based on its own rating and its client's rating. Formally, the recommended plan, denoted by α_0, is a mapping $\alpha_0 : \Theta \times \Theta \to \{0, 1\}$, where 0 and 1 represent "low-quality service" and "high-quality service," respectively. Then $\alpha_0(\theta_c, \theta_s)$ denotes the recommended service quality for a server with rating θ_s when it is matched to a client with rating θ_c. We write the set of recommended plans as $\mathcal{A} = \{\alpha | \alpha : \Theta \times \Theta \to \{0, 1\}\}$. We are particularly interested in the following three plans. Plan α^{a} is the *altruistic* plan:

$$\alpha^{\mathrm{a}}(\theta_c, \theta_s) = 1, \forall \theta_c, \theta_s \in \{0, 1\}, \tag{1}$$

where the server provides high-quality service regardless of its own and its client's ratings. Plan α^{f} is the *fair* plan:

$$\alpha^{\mathrm{f}}(\theta_c, \theta_s) = \begin{cases} 0 & \theta_s > \theta_c \\ 1 & \theta_s \leq \theta_c \end{cases}, \tag{2}$$

where the server provides high-quality service when its client has higher or equal ratings. Plan α^{s} is the *selfish* plan:

$$\alpha^{\mathrm{s}}(\theta_c, \theta_s) = 0, \forall \theta_c, \theta_s \in \{0, 1\}, \tag{3}$$

where the server provides low-quality service regardless of its own and its client's ratings. Note that we can consider the selfish plan as a nondifferential punishment in which everyone receives low-quality services, and consider the fair plan as a differential punishment in which users with different ratings receive different services.

Service Requests. The platform receives service requests from the users. We assume that there is no cost in requesting services and that each user always has demands for services. Hence, all the users will request services.

Matching. The platform matches each user i, as a client, to another user $m(i)$ who will serve i, where m is a matching $m : \mathcal{N} \to \mathcal{N}$. Since the platform cannot match a user to itself, we write the set of all possible matchings as $M = \{m : m \text{ bijective}, \ m(i) \neq i, \forall i \in \mathcal{N}\}$. The mechanism defines a random matching rule, which is a probability distribution μ on the set of all possible matchings M. In this article, we focus on the uniformly random matching rule, which chooses every possible matching $m \in M$ with the same probability. The analysis can be easily generalized to the cases with nonuniformly random matching rules, as long as the matching rules do not distinguish users with the same rating.

Clients' Ratings. The platform will inform each server of its client's rating such that each server can choose its service quality based on its own and its client's ratings.

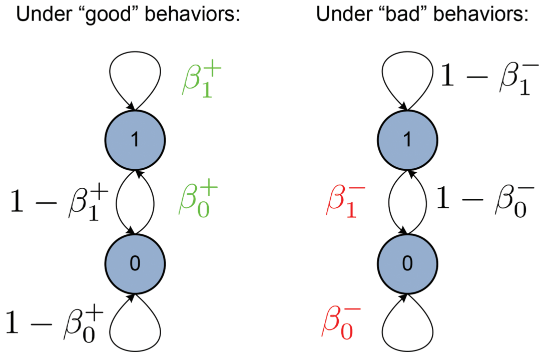


Fig. 2. Illustration of the rating update rule. The circle denotes the rating, and the arrow denotes the rating update with corresponding probabilities.

Reports. After the servers serve their clients, the platform elicits reports from the clients about their service quality. However, the report is *inaccurate*, either by the client's incapability of accurate assessment (e.g., the client, who wants to translate some sentences into a foreign language, cannot accurately evaluate the server's translation) or by some system error (e.g., the data/file sent by the server is missing due to network errors). We characterize the erroneous report by a mapping $R : \{0, 1\} \to \Delta(\{0, 1\})$, where $\Delta(\{0, 1\})$ is the probability distribution over $\{0, 1\}$. For example, $R(1|q)$ is the probability that the client reports “high quality” given the server's actual service quality q. In this article, we focus on reports of the following form.

$$R(r|q) = \begin{cases} 1-\varepsilon, & r = q \\ \varepsilon, & r \neq q \end{cases}, \ \forall r, q \in \{0, 1\}, \tag{4}$$

where $\varepsilon \in [0, 0.5)$ is the report error probability.[7] Note, however, that reporting is not strategic: the client reports truthfully but with errors. If the clients report strategically, the mechanism can let the platform assess the service quality (still, with errors) to avoid strategic reporting. For simplicity, we assume that the report error is symmetric in the sense that reporting high and low qualities have the same error probability. Extension to asymmetric report errors is straightforward.

Rating Update. Given clients' reports, the platform updates servers' ratings according to the *rating update rule*, which is defined as a mapping $\tau : \Theta \times \Theta \times \{0, 1\} \times \mathcal{A} \to \Delta(\Theta)$. For example, $\tau(\theta_s'|\theta_c, \theta_s, r, \alpha_0)$ is the probability of the server's updated rating being θ_s', given the client's rating θ_c, the server's own rating θ_s, the client's report r, and the recommended plan α_0. We focus on the following class of rating update rules (see Figure 2 for illustration):

$$\tau(\theta_s'|\theta_c, \theta_s, r, \alpha_0) = \begin{cases} \beta_{\theta_s}^+, & \theta_s' = 1, r \geq \alpha_0(\theta_c, \theta_s) \\ 1-\beta_{\theta_s}^+, & \theta_s' = 0, r \geq \alpha_0(\theta_c, \theta_s) \\ 1-\beta_{\theta_s}^-, & \theta_s' = 1, r < \alpha_0(\theta_c, \theta_s) \\ \beta_{\theta_s}^-, & \theta_s' = 0, r < \alpha_0(\theta_c, \theta_s) \end{cases}. \tag{5}$$

In this rating update rule, if the reported service quality is not lower than the recommended service quality, a server with rating θ_s will have rating 1 with probability

[7] We confine the report error probability ε to be smaller than 0.5. If the error probability ε is 0.5, the report contains no useful information about the service quality. If the error probability is larger than 0.5, the rating mechanism can use the opposite of the report as an indication of the service quality, which is equivalent to the case with the error probability smaller than 0.5.

Table I. Gift-Giving Game between a Client and a Server

	high-quality service	low-quality service
request	$(b, -c)$	$(0, 0)$

$\beta^+_{\theta_s}$; otherwise, it will have rating 0 with probability $\beta^-_{\theta_s}$. Other more elaborate rating update rules may be considered, but we show that this simple one is good enough to achieve the social optimum.

Recommended Strategy. The final key component of the rating mechanism is the recommended *strategy*, which determines what recommended plan should be announced in each period. In each period t, the mechanism keeps track of the history of rating distributions, denoted by $\boldsymbol{h}^t = (\boldsymbol{s}^0, \ldots, \boldsymbol{s}^t) \in S^{t+1}$, and chooses the recommended plan based on $\boldsymbol{h}^t$. In other words, the recommended strategy is a mapping from the set of histories to its plan set, denoted by $\pi_0 : \cup_{t=0}^{\infty} S^{t+1} \rightarrow \mathcal{A}$. Denote the set of all recommended strategies by Π. Note that although the rating mechanism knows the rating profile, it determines the recommended plan based on the history of rating distributions, because (1) this reduces the computational and memory complexity of the protocol, and (2) it is easy for the users to follow, since they do not know the rating profile. Moreover, since the plan set $\mathcal{A}$ has 16 elements, the complexity of choosing the plan is large. Hence, we consider the strategies that choose plans from a subset $\mathcal{B} \subseteq \mathcal{A}$, and define $\Pi(\mathcal{B})$ as the set of strategies restricted on the subset $\mathcal{B}$ of plans.

In summary, the rating mechanism can be represented by the design parameters: the rating update rule and the recommended strategy, and can be denoted by the tuple (τ, π_0).

3.1.2. Payoffs.

Once a server and a client are matched, they play the gift-giving game in Table I [Zhang et al. 2014; Kandori 1992; Takahashi 2010; Ellison 1994], where the row player is the client and the column player is the server. We normalize the payoffs received by the client and the server when a server provides low-quality services to 0. When a server provides high-quality services, the client gets a benefit of $b > 0$ and the worker incurs a cost of $c \in (0, b)$. In the unique Nash equilibrium of the gift-giving game, the server will provide low-quality services, which results in a zero payoff for both the client and the server. Note that as in these works, we assume that the same gift-giving game is played for all the client-server pairs. This assumption is reasonable when the number of users is large. Since b can be considered a user's expected benefit across different servers, and c as its expected cost of high-quality service across different clients, the users' expected benefits/costs should be approximately the same when the number of users is large. This assumption is also valid when the users have different realized benefits and costs in each period but the same expected benefit b and expected cost c across different periods.

Expected Payoff in One Period. Based on the gift-giving game, we can calculate each user's expected payoff obtained in one period. A user's expected payoff in one period depends on its own rating, the rating distribution, and the users' plans. We write user i's plan as $\alpha_i \in \mathcal{A}$, and the plan profile of all the users as $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$. Then user i's expected payoff in one period is $u_i(\theta_i, \boldsymbol{s}, \boldsymbol{\alpha})$. For illustration, we calculate the users' expected payoffs under several important scenarios, assuming that all the users follow the recommended plan (i.e., $\alpha_i = \alpha_0, \ \forall i \in \mathcal{N}$). When the altruistic plan α^{a} is recommended, all the users receive the same expected payoff in one period as

$$u_i(\theta_i, \boldsymbol{s}, \alpha^{\mathrm{a}} \cdot \mathbf{1}_N) = b - c, \ \forall i, \theta_i, \boldsymbol{s}, \tag{6}$$

where $\alpha \cdot \mathbf{1}_N$ is the plan profile in which every user chooses plan α. Similarly, when the selfish plan α^{s} is recommended, all the users receive zero expected payoff in one period, namely,

$$u_i(\theta_i, \boldsymbol{s}, \alpha^{\mathrm{s}} \cdot \mathbf{1}_N) = 0,\ \forall i, \theta_i, \boldsymbol{s}. \tag{7}$$

When the fair plan α^{f} is recommended, the users receive expected payoffs in one period as follows:

$$u_i(\theta_i, \boldsymbol{s}, \alpha^{\mathrm{f}} \cdot \mathbf{1}_N) = \begin{cases} \frac{s_0-1}{N-1} \cdot b - c, & \theta_i = 0 \\ b - \frac{s_1-1}{N-1} \cdot c, & \theta_i = 1 \end{cases}. \tag{8}$$

Under the fair plan, the users with rating 1 receive a payoff higher than $b - c$, because they get high-quality services from everyone and provide high-quality services only when matched to clients with rating 1. In contrast, users with rating 0 receive a payoff lower than $b - c$. Hence, the fair plan α^{f} can be considered as differential punishment.

Discounted Average Payoff. Each user i has its own strategy $\pi_i \in \Pi$. Write the joint strategy profile of all the users as $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$. Then given the initial rating profile $\boldsymbol{\theta}^0$, the recommended strategy π_0 and the joint strategy profile $\boldsymbol{\pi}$ induce a probability distribution over the sequence of rating profiles $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots$. Taking the expectation with respect to this probability distribution, each user i receives a discounted average payoff $U_i(\boldsymbol{\theta}^0, \pi_0, \boldsymbol{\pi})$ calculated as

$$U_i(\boldsymbol{\theta}^0, \pi_0, \boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots} \left\{ (1-\delta) \sum_{t=0}^{\infty} \delta^t u_i\left(\theta_i^t, \boldsymbol{s}(\boldsymbol{\theta}^t), \boldsymbol{\pi}(\boldsymbol{s}(\boldsymbol{\theta}^0), \ldots, \boldsymbol{s}(\boldsymbol{\theta}^t))\right) \right\}, \tag{9}$$

where $\delta \in [0, 1)$ is the common discount factor of all the users. The discount factor δ is the rate at which the users discount future payoffs and reflects the patience of the users. A more patient user has a larger discount factor. Note that the recommended strategy π_0 does affect users' discounted average payoffs by affecting the evolution of the rating profile (i.e., by affecting the expectation operator $\mathbb{E}_{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots}$).

3.1.3. Definition of the Equilibrium.

The platform adopts *sustainable rating mechanisms*, which specifies a tuple of rating update rule and recommended strategy (τ, π_0) such that the users find it in their self-interests to follow the recommended strategy. In other words, the recommended strategy should be an equilibrium.

Note that the interaction among the users is neither a repeated game [Fudenberg et al. 1994] nor a standard stochastic game [Hörner et al. 2011]. In a repeated game, every stage game is the same, which is clearly not true in the platform because users' stage-game payoff $u_i(\theta_i, \boldsymbol{s}, \boldsymbol{\alpha})$ depends on the rating distribution $\boldsymbol{s}$. In a standard stochastic game, the state must satisfy the following: (1) the state and the plan profile uniquely determine the stage-game payoff, and (2) the state is known to all users. In the platform, user's stage-game payoff $u_i(\theta_i, \boldsymbol{s}, \boldsymbol{\alpha})$ depends on its own rating θ_i, which should be included in the state and be known to all the users. Hence, if we were to model the interaction as a standard stochastic game, we need to define the state as the rating profile $\boldsymbol{\theta}$. However, the rating profile is not known to the users in our formulation.

To reflect our restriction on recommended strategies that depend only on rating distributions, we define the equilibrium as public announcement equilibrium (PAE), since the strategy depends on the publicly announced rating distributions. Before we define PAE, we need to define the continuation strategy $\pi|_{\boldsymbol{h}^t}$, which is a mapping $\pi|_{\boldsymbol{h}^t} : \cup_{k=0}^{\infty} \mathcal{H}^k \rightarrow A$ with $\pi|_{\boldsymbol{h}^t}(\boldsymbol{h}^k) = \pi(\boldsymbol{h}^t \boldsymbol{h}^k)$, where $\boldsymbol{h}^t \boldsymbol{h}^k$ is the concatenation of $\boldsymbol{h}^t$ and $\boldsymbol{h}^k$.

Definition 3.1. A pair of a recommended strategy and a symmetric strategy profile $(\pi_0, \pi_0 \cdot \mathbf{1}_N)$ is a PAE, if for all $t \geq 0$, for all $\tilde{\boldsymbol{h}}^t \in \mathcal{H}^t$, and for all $i \in \mathcal{N}$, we have

$$U_i(\tilde{\boldsymbol{\theta}}^t, \pi_0|_{\tilde{\boldsymbol{h}}^t}, \pi_0|_{\tilde{\boldsymbol{h}}^t} \cdot \mathbf{1}_N) \geq U_i(\tilde{\boldsymbol{\theta}}^t, \pi_0|_{\tilde{\boldsymbol{h}}^t}, (\pi_i|_{\tilde{\boldsymbol{h}}^t}, \pi_0|_{\tilde{\boldsymbol{h}}^t} \cdot \mathbf{1}_{N-1})),\ \forall \pi_i|_{\tilde{\boldsymbol{h}}^t} \in \Pi, \tag{10}$$

where $(\pi_i|_{\tilde{\boldsymbol{h}}^t}, \pi_0|_{\tilde{\boldsymbol{h}}^t} \cdot \mathbf{1}_{N-1})$ is the continuation strategy profile in which user i deviates to $\pi_i|_{\tilde{\boldsymbol{h}}^t}$ and the other users follow the strategy $\pi_0|_{\tilde{\boldsymbol{h}}^t}$.

Note that in the definition, we allow a user to deviate to any strategy $\pi_i \in \Pi$, even if the recommended strategy π_0 is restricted to a subset $\mathcal{B}$ of plans. Hence, the rating mechanism is robust in the sense that a user cannot gain even when it uses more complicated strategies. Note also that although a rating mechanism can choose the initial rating profile $\boldsymbol{\theta}^0$, we require a recommended strategy to fulfill the incentive constraints under all the initial rating profiles. This adds to the flexibility in choosing the initial rating profile.

PAE is stronger than the Nash equilibrium (NE), because PAE requires the users to not deviate following any history, while NE requires the users to not deviate following the histories that happen in the equilibrium. In this sense, PAE can be considered a special case of public perfect equilibrium (PPE) in standard repeated and stochastic games, where the strategies depend only on the rating distribution.

3.2. The Rating Protocol Design Problem

The goal of the rating mechanism designer is to choose a rating mechanism (τ, π_0) such that the social welfare at the equilibrium is maximized in the worst case (with respect to different initial rating profiles). Maximizing the worst-case performance gives us a much stronger performance guarantee than maximizing the performance under a given initial rating profile. Given the rating update error ε, the discount factor δ, and the subset $\mathcal{B}$ of plans, the rating mechanism design problem is formulated as

$$\begin{aligned} W(\varepsilon, \delta, \mathcal{B}) = \max_{\tau, \pi_0 \in \Pi(\mathcal{B})} \; & \min_{\boldsymbol{\theta}^0 \in \Theta^N} \frac{1}{N} \sum_{i \in \mathcal{N}} U_i(\boldsymbol{\theta}^0, \pi_0, \pi_0 \cdot \mathbf{1}_N) \\ s.t. \quad & (\pi_0, \pi_0 \cdot \mathbf{1}_N) \text{ is a PAE.} \end{aligned} \tag{11}$$

Note that $W(\varepsilon, \delta, \mathcal{B})$ is strictly smaller than the social optimum $b-c$ for any ε, δ, and $\mathcal{B}$. This is because to exactly achieve $b-c$, the protocol must recommend the altruistic plan α^{a} all the time (even when someone shirks), which cannot be an equilibrium. However, we can design rating mechanisms such that for any fixed $\varepsilon \in [0, 0.5)$, $W(\varepsilon, \delta, \mathcal{B})$ can be arbitrarily close to the social optimum. In particular, such rating mechanisms can be simple in that $\mathcal{B}$ can be a small subset of three plans (i.e., $\mathcal{B} = A^{\mathrm{afs}} \triangleq \{\alpha^{\mathrm{a}}, \alpha^{\mathrm{f}}, \alpha^{\mathrm{s}}\}$).

4. SOURCES OF INEFFICIENCY

To illustrate the importance of designing optimal yet simple rating schemes, as well as the challenges associated with determining such a design, in this section, we discuss several simple rating mechanisms that appear to work well intuitively and show that they are actually bounded away from the social optimum even when the users are arbitrarily patient. We will illustrate why they are inefficient, which gives us some insights on how to design socially-optimal rating mechanisms.

4.1. Stationary Recommended Strategies

4.1.1. Analysis. We first consider rating mechanisms with stationary recommended strategies, which determine the recommended plan solely based on the current rating distribution. Since the game is infinitely-repeatedly played, given the same rating distribution, the continuation game is the same regardless of when the rating distribution occurs. Hence, similar to MDP, we can assign value functions $V_\theta^{\pi_0} : S \rightarrow \mathbb{R}$, $\forall \theta$ for a stationary strategy π_0, with $V_\theta^{\pi_0}(\boldsymbol{s})$ being the continuation payoff of a user with rating θ at the rating distribution $\boldsymbol{s}$. Then, we have the following set of equalities that the value function needs to satisfy

$$V_{\theta_i}^{\pi_0}(\boldsymbol{s}) = (1-\delta)\cdot u_i(\pi_0(\boldsymbol{s}), \pi_0(\boldsymbol{s})\cdot \mathbf{1}_N) \\ + \delta \cdot \sum_{\theta_i', \boldsymbol{s}'} \Pr(\theta_i', \boldsymbol{s}' | \theta_i, \boldsymbol{s}, \pi_0(\boldsymbol{s}), \pi_0(\boldsymbol{s})\cdot \mathbf{1}_N)\cdot V_{\theta_i'}^{\pi_0}(\boldsymbol{s}'), \forall i \in \mathcal{N}, \tag{12}$$

where $\Pr(\theta_i', \boldsymbol{s}' | \theta_i, \boldsymbol{s}, \pi_0(\boldsymbol{s}), \pi_0(\boldsymbol{s})\cdot \mathbf{1}_N)$ is the transition probability. We can solve for the value function from the preceding set of equalities, which are similar to the Bellman equations in MDP. However, note that obtaining the value function is not the final step. We also need to check the incentive compatibility (IC) constraints. For example, to prevent user i from deviating to plan α', the following inequality has to be satisfied:

$$V_{\theta_i}^{\pi_0}(\boldsymbol{s}) \geq (1-\delta)\cdot u_i(\pi_0(\boldsymbol{s}), (\alpha', \pi_0(\boldsymbol{s})\cdot \mathbf{1}_{N-1})) \\ + \delta \cdot \sum_{\theta_i', \boldsymbol{s}'} \Pr(\theta_i', \boldsymbol{s}' | \theta_i, \boldsymbol{s}, \pi_0(\boldsymbol{s}), (\alpha', \pi_0(\boldsymbol{s})\cdot \mathbf{1}_{N-1}))\cdot V_{\theta_i'}^{\pi_0}(\boldsymbol{s}'), \forall i \in \mathcal{N}. \tag{13}$$

Given a rating mechanism with a stationary recommended strategy π_0, if its value function satisfies all the IC constraints, we can determine the social welfare of the rating mechanism. For example, suppose that all the users have an initial rating of 1. Then, all of them achieve the expected payoff $V_1^{\pi_0}(0, N)$, which is the social welfare achieved under this rating mechanism.

Note that given a recommended strategy π_0, it is not difficult to compute the value function by solving the set of linear equations in (12) and check the IC constraints according to the set of linear inequalities in (13). However, it is difficult to derive structural results on the value function (e.g., whether the state with more rating-1 users has a higher value), and thus difficult to know the structures of the optimal recommended strategy (e.g., whether the optimal recommended strategy is a threshold strategy). The difficulty mainly comes from the complexity of the transition probabilities $\Pr(\theta_i', \boldsymbol{s}' | \theta_i, \boldsymbol{s}, \pi_0(\boldsymbol{s}), \pi_0(\boldsymbol{s})\cdot \mathbf{1}_N)$. For example, assuming $\pi_0(\boldsymbol{s}) = \alpha^{\mathrm{a}}$, we have

$$\Pr(1, \boldsymbol{s}' | 1, \alpha^{\mathrm{a}}, \alpha^{\mathrm{a}}\cdot \mathbf{1}_N) = \\ x_1^+ \cdot \sum_{k=\max\{0, s_1'-1-(N-s_1)\}}^{\min\{s_1-1, s_1'-1\}} \begin{matrix} \binom{s_1-1}{k}(x_1^+)^k(1-x_1^+)^{s_1-1-k} \\ \binom{N-s_1}{s_1'-1-k}(x_0^+)^{s_1'-1-k}(1-x_0^+)^{N-s_1-(s_1'-1-k)} \end{matrix}, \tag{14}$$

where $x_1^+ \triangleq (1-\varepsilon)\beta_1^+ + \varepsilon(1-\beta_1^-)$ is the probability that a rating-1 user's rating remains 1, and $x_0^+ \triangleq (1-\varepsilon)\beta_0^+ + \varepsilon(1-\beta_0^-)$ is the probability that a rating-0 user's rating goes up to 1. We can see that the transition probability has combinatorial numbers in it and is complicated. Hence, although the stationary strategies themselves are simpler than the nonstationary strategies, they are harder to compute in the sense that it is difficult to derive structural results for rating mechanisms with stationary recommended strategy. In contrast, we are able to develop a unified design framework for socially-optimal rating mechanisms with nonstationary recommended strategies.

4.1.2. Inefficiency. We measure the efficiency of the rating mechanisms with stationary recommended strategies using the *price of stationarity* (PoStat), defined as

$$\mathrm{PoStat}(\varepsilon, \mathcal{B}) = \frac{\lim_{\delta\to 1} W^s(\varepsilon, \delta, \mathcal{B})}{b-c}, \tag{15}$$

where $W^s(\varepsilon, \delta, \mathcal{B})$ is the optimal value of a modified optimization problem (11) with an additional constraint that π_0 is stationary.

Note that $\mathrm{PoStat}(\varepsilon, \mathcal{B})$ measures the efficiency of a class of rating mechanisms (not a specific rating mechanism), because we optimize over all the rating update rules and

Table II. Normalized Social Welfare of Stationary Strategies Restricted on A^{afs}

δ	0.7	0.8	0.9	0.99	0.999	0.9999
Normalized welfare	0.690	0.700	0.715	0.720	0.720	0.720

Table III. Minimum Punishment Probabilities of Rating Mechanisms Restricted on A^{afs} when $\varepsilon = 0.1$

δ	0.7	0.8	0.9	0.99	0.999	0.9999	0.99999
Minimum β_1^-	0.8	0.8	0.6	0.6	0.3	0.3	0.3

stationary recommended strategies restricted on $\mathcal{B}$. PoStat is a number between 0 and 1. A small PoStat indicates low efficiency.

Through simulation, we can compute PoStat$(0.1, A^{\mathrm{afs}}) = 0.720$. In other words, even with differential punishment α^{f}, the performance of stationary strategies is bounded away from social optimum. We compute the PoStat in a platform with $N = 5$ users, benefit $b = 3$, cost $c = 1$, and $\varepsilon = 0.1$. Under each discount factor δ, we assign values between 0 and 1 with a 0.1 grid to β_θ^+, β_θ^- in the rating update rule, namely, we try 11^4 rating update rules to select the optimal one. For each rating update rule, we try all $3^{N+1} = 729$ stationary recommended strategies restricted on A^{afs}. In Table II, we list normalized social welfare under different discount factors.

As previously mentioned, the inefficiency of stationary strategies is due to the punishment exerted under certain rating distributions. For example, the optimal recommended strategies discussed recommend the selfish or fair plan when at least one user has rating 0, resulting in performance loss. One may think that when the users are more patient (i.e., when the discount factor is larger), we can use milder punishments by lowering the punishment probabilities β_1^- and β_0^-, such that the rating distributions with many low-rating users happen with less frequency. However, simulations on these strategies show that, to fulfill the IC constraints, the punishment probabilities cannot be made arbitrarily small even when the discount factor is large. For example, Table III shows the minimum punishment probability β_1^- (which is smaller than β_0^-) of rating mechanisms restricted on A^{afs} under different discount factors. In other words, the rating distributions with many low-rating users will happen with some probabilities bounded above zero, with a bound independent of the discount factor. Hence, the performance loss is bounded above zero regardless of the discount factor. Note that in a nonstationary strategy, we could choose whether to punish in rating distributions with many low-rating users, depending on the history of past rating distributions. This adaptive adjustment of punishments allows nonstationary strategies to potentially achieve the social optimum.

4.2. Lack of Differential Punishments

We have discussed in the previous section the inefficiency of stationary strategies. Now we consider a class of nonstationary strategies restricted on the subset of plans A^{as}. Under this class of strategies, all the users are rewarded (by choosing α^{a}) or punished (by choosing α^{s}) simultaneously. In other words, there is no differential punishment that can "transfer" some payoff from low-rating users to high-rating users. We quantify the performance loss of this class of nonstationary strategies restricted on A^{as} as follows.

PROPOSITION 4.1. *For any $\varepsilon > 0$, we have*

$$\lim_{\delta \to 1} W(\varepsilon, \delta, A^{\mathrm{as}}) \leq b - c - \zeta(\varepsilon), \tag{16}$$

where $\zeta(\varepsilon) > 0$ for any $\varepsilon > 0$.

Proof. The proof is similar to the proof of Dellarocas [2005, Proposition 6]; see the Online Appendix A. □

This proposition shows that the maximum social welfare achievable by $(\pi_0, \pi \cdot \mathbf{1}_N) \in \Pi(A^{\text{as}}) \times \Pi^N(A^{\text{as}})$ at the equilibrium is bounded away from the social optimum $b - c$, unless there is no rating update error. Note that the performance loss is independent of the discount factor. In contrast, we will show later that if we can use the fair plan α^{f}, the social optimum can be asymptotically achieve when the discount factor goes to 1. Hence, the differential punishment introduced by the fair plan is crucial for achieving the social optimum.

5. SOCIALLY OPTIMAL DESIGN

In this section, we design rating mechanisms that asymptotically achieve the social optimum at the equilibrium, even when the rating update rule $\varepsilon > 0$. In our design, we use the APS technique, named after the authors of the seminal paper [Abreu et al. 1990], which is also used to prove the folk theorem for repeated games in Fudenberg et al. [1994] and for stochastic games in Hörner et al. [2011]. We will briefly introduce the APS technique first. Meanwhile, more importantly, we will illustrate why we cannot use APS in our setting in the same way as Fudenberg et al. [1994] and Hörner et al. [2011] do. Then, we will show how we use APS in a different way in our setting in order to design the optimal rating mechanism and to construct the equilibrium strategy. Finally, we analyze the performance of a class of simple but suboptimal strategies, which sheds light on why the proposed strategy can achieve the social optimum.

5.1. The APS Technique

APS [Abreu et al. 1990] provides a characterization of the set of PPE payoffs. It builds on the idea of *self-generating sets* described as follows. Note that APS is used for standard stochastic games, and recall from our discussion in Section 2 that the state of the standard stochastic game is the rating profile $\boldsymbol{\theta}$. Then define a set $\mathcal{W}^{\boldsymbol{\theta}} \subset \mathbb{R}^N$ for every state $\boldsymbol{\theta} \in \Theta^N$, and write $(\mathcal{W}^{\boldsymbol{\theta}'})_{\boldsymbol{\theta}' \in \Theta^N}$ as the collection of these sets. Then we have the following definitions [Abreu et al. 1990; Hörner et al. 2011; Mailath and Samuelson 2006]. First, we say a payoff profile $\boldsymbol{v}(\boldsymbol{\theta}) \in \mathbb{R}^N$ is *decomposable* on $(\mathcal{W}^{\boldsymbol{\theta}'})_{\boldsymbol{\theta}' \in \Theta^N}$ given $\boldsymbol{\theta}$, if there exists a recommended plan α_0, an plan profile $\boldsymbol{\alpha}^*$, and a *continuation payoff function* $\boldsymbol{\gamma} : \Theta^N \to \cup_{\boldsymbol{\theta}' \in \Theta^N} \mathcal{W}^{\boldsymbol{\theta}'}$ with $\boldsymbol{\gamma}(\boldsymbol{\theta}') \in \mathcal{W}^{\boldsymbol{\theta}'}$, such that for all $i \in \mathcal{N}$ and for all $\alpha_i \in A$,

$$\begin{aligned} v_i &= (1-\delta) u_i(\boldsymbol{\theta}, \alpha_0, \boldsymbol{\alpha}^*) + \delta \sum_{\boldsymbol{\theta}'} \gamma_i(\boldsymbol{\theta}') q(\boldsymbol{\theta}' | \boldsymbol{\theta}, \alpha_0, \boldsymbol{\alpha}^*) \\ &\geq (1-\delta) u_i(\boldsymbol{\theta}, \alpha_0, \alpha_i, \boldsymbol{\alpha}^*_{-i}) + \delta \sum_{\boldsymbol{\theta}'} \gamma_i(\boldsymbol{\theta}') q(\boldsymbol{\theta}' | \boldsymbol{\theta}, \alpha_0, \alpha_i, \boldsymbol{\alpha}^*_{-i}). \end{aligned} \tag{17}$$

Then, we say a set $(\mathcal{W}^{\boldsymbol{\theta}})_{\boldsymbol{\theta} \in \Theta^N}$ is a self-generating set, if for any $\boldsymbol{\theta}$, every payoff profile $\boldsymbol{v}(\boldsymbol{\theta}) \in \mathcal{W}^{\boldsymbol{\theta}}$ is decomposable on $(\mathcal{W}^{\boldsymbol{\theta}'})_{\boldsymbol{\theta}' \in \Theta^N}$ given $\boldsymbol{\theta}$. The important property of self-generating sets is that any self-generating set is a set of PPE payoffs [Abreu et al. 1990; Hörner et al. 2011; Mailath and Samuelson 2006].

Based on the idea of self-generating sets, Fudenberg et al. [1994] and Hörner et al. [2011] prove the folk theorem for repeated games and stochastic games, respectively. However, we cannot use APS in the same way as Fudenberg et al. [1994] and Hörner et al. [2011] do for the following reason. We assume that the users do not know the rating profile of every user and restrict our attention on symmetric PA strategy profiles. This requires that each user i cannot use the continuation payoff function $\gamma_i(\boldsymbol{\theta})$ directly. Instead, each user i should assign the same continuation payoff for the rating profiles that have the same rating distribution, namely, $\gamma_i(\boldsymbol{\theta}) = \gamma_i(\boldsymbol{\theta}')$ for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ such that $\boldsymbol{s}(\boldsymbol{\theta}) = \boldsymbol{s}(\boldsymbol{\theta}')$.

5.2. Socially Optimal Design

As mentioned, the social optimum $b - c$ can be exactly achieved only by servers providing high-quality service all the time, which is not an equilibrium. Hence, we aim at achieving the social optimum $b-c$ asymptotically. We define the asymptotically socially optimal rating mechanisms as follows.

Definition 5.1 (*Asymptotically Socially Optimal Rating Mechanisms*). Given a rating update error $\varepsilon \in [0, 0.5)$, we say a rating mechanism $(\tau(\varepsilon), \pi_0(\varepsilon, \xi, \delta) \in \Pi)$ is asymptotically socially optimal under ε, if for any small performance loss $\xi > 0$, we can find a $\underline{\delta}(\xi)$, such that for any discount factor $\delta > \underline{\delta}(\xi)$, we have

—$(\pi_0(\xi, \delta), \pi_0(\xi, \delta) \cdot \mathbf{1}_N)$ is a PAE;
—$U_i(\boldsymbol{\theta}^0, \pi_0, \pi_0 \cdot \mathbf{1}_N) \geq b - c - \xi, \ \forall i \in \mathcal{N}, \ \forall \boldsymbol{\theta}^0$.

Note that in the asymptotically socially optimal rating mechanism, the rating update rule depends only on the rating update error and works for any tolerated performance loss ξ and for any the discount factor $\delta > \underline{\delta}$. The recommended strategy π_0 is a class of strategies parameterized by $(\varepsilon, \xi, \delta)$ and works for any $\varepsilon \in [0, 0.5)$, any $\xi > 0$, and any discount factor $\delta > \underline{\delta}$ under the rating update rule $\tau(\varepsilon)$.

First, we define a few auxiliary variables first for better exposition of the theorem. Define $\kappa_1 \triangleq \frac{b}{\frac{N-2}{N-1}b-c} - 1$ and $\kappa_2 \triangleq 1 + \frac{c}{(N-1)b}$. In addition, we write the probability that a user with rating 1 has its rating remain at 1 if it follows the recommended altruistic plan α^{a} as

$$x_1^+ \triangleq (1-\varepsilon)\beta_1^+ + \varepsilon(1-\beta_1^-). \tag{18}$$

Write the probability that a user with rating 1 has its rating remain at 1 if it follows the recommended fair plan α^{f} as

$$x_{s_1(\boldsymbol{\theta})} \triangleq \left[(1-\varepsilon)\frac{s_1(\boldsymbol{\theta})-1}{N-1} + \frac{N-s_1(\boldsymbol{\theta})}{N-1}\right]\beta_1^+ + \left(\varepsilon\frac{s_1(\boldsymbol{\theta})-1}{N-1}\right)(1-\beta_1^-). \tag{19}$$

Write the probability that a user with rating 0 has its rating increase to 1 if it follows the recommended plan α^{a} or α^{f}:

$$x_0^+ \triangleq (1-\varepsilon)\beta_0^+ + \varepsilon(1-\beta_0^-). \tag{20}$$

THEOREM 5.2. *Given any rating update error $\varepsilon \in [0, 0.5)$,*

—(Design rating update rules). A rating update rule $\tau(\varepsilon)$ that satisfies
—Condition 1: (following the recommended plan leads to a higher rating).

$$\beta_1^+ > 1 - \beta_1^- \text{ and } \beta_0^+ > 1 - \beta_0^-. \tag{21}$$

—Condition 2: (enough "reward" for users with rating 1).

$$x_1^+ = (1-\varepsilon)\beta_1^+ + \varepsilon(1-\beta_1^-) > \frac{1}{1+\frac{c}{(N-1)b}}. \tag{22}$$

—Condition 3: (enough "punishment" for users with rating 0).

$$x_0^+ = (1-\varepsilon)\beta_0^+ + \varepsilon(1-\beta_0^-) < \frac{1-\beta_1^+}{\frac{c}{(N-1)b}}, \tag{23}$$

can be the rating update rule in a asymptotically socially-optimal rating mechanism.

—*(Optimal recommended strategies). Given the rating update rule $\tau(\varepsilon)$ that satisfies these conditions, any small performance loss $\xi > 0$, and any discount factor $\delta \geq \underline{\delta}(\varepsilon, \xi)$ with $\underline{\delta}(\varepsilon, \xi)$ defined in Appendix A, the recommended strategy $\pi_0(\varepsilon, \xi, \delta) \in \Pi(A^{\text{afs}})$ constructed by Algorithm 1 is the recommended strategy in a asymptotically socially-optimal rating mechanism.*

Proof. See Online Appendix B for the entire proof. We provide a proof sketch here. The proof builds on the theory of self-generating sets [Abreu et al. 1990], which can be considered the extension of Bellman equations in dynamic programming to cases with multiple self-interested users using nonstationary strategies. We can decompose each user i's discounted average payoff into the current payoff and the continuation payoff as follows:

$$\begin{aligned} & U_i(\boldsymbol{\theta}^0, \pi_0, \boldsymbol{\pi}) \\ &= \mathbb{E}_{\boldsymbol{\theta}^1,\ldots} \left\{ (1-\delta) \sum_{t=0}^{\infty} \delta^t u_i\big(\theta_i^t, \boldsymbol{s}(\boldsymbol{\theta}^t), \boldsymbol{\pi}(\boldsymbol{s}(\boldsymbol{\theta}^0), \ldots, \boldsymbol{s}(\boldsymbol{\theta}^t))\big) \right\} \\ &= (1-\delta) \cdot \underbrace{u_i\big(\theta_i^0, \boldsymbol{s}(\boldsymbol{\theta}^0), \boldsymbol{\pi}(\boldsymbol{s}(\boldsymbol{\theta}^0))\big)}_{\text{current payoff at } t=0} \qquad (24) \\ &+ \underbrace{\delta \cdot \mathbb{E}_{\boldsymbol{\theta}^2,\ldots} \left\{ (1-\delta) \sum_{t=1}^{\infty} \delta^{t-1} u_i\big(\theta_i^t, \boldsymbol{s}(\boldsymbol{\theta}^t), \boldsymbol{\pi}(\boldsymbol{s}(\boldsymbol{\theta}^0), \ldots, \boldsymbol{s}(\boldsymbol{\theta}^t))\big) \right\}}_{\text{continuation payoff starting from } t=1}. \end{aligned}$$

We can see that the continuation payoff starting from $t = 1$ is the discounted average payoff as if the system starts from $t = 1$. Suppose that the users follow the recommended strategy. Since the recommended strategy and the rating update rule do not differentiate users with the same rating, we can prove that the users with the same rating have the same continuation payoff starting from any point. Hence, given π_0 and $\boldsymbol{\pi} = \pi_0 \cdot \mathbf{1}_N$, the preceding decomposition can be simplified to

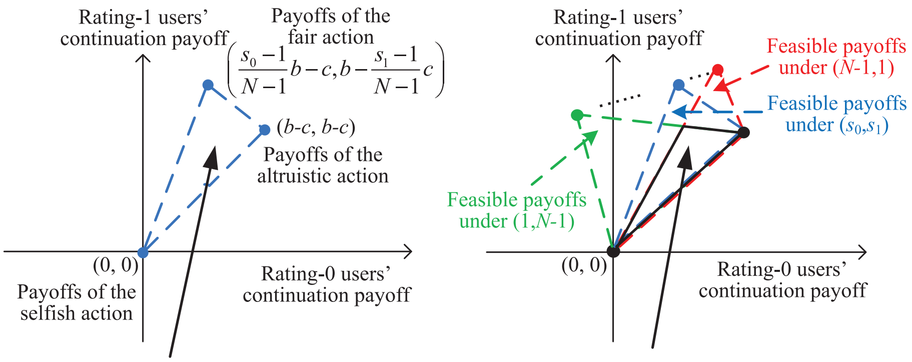
$$v_\theta^{\pi_0}(\boldsymbol{s}) = (1-\delta) \cdot u(\theta, \boldsymbol{s}, \alpha_0 \cdot \mathbf{1}_N) + \delta \cdot \sum_{\theta'=0}^{1} \sum_{\boldsymbol{s}'} q(\theta', \boldsymbol{s}' | \theta, \boldsymbol{s}, \alpha_0 \cdot \mathbf{1}_N) \cdot v_{\theta'}^{\pi_0}(\boldsymbol{s}'), \qquad (25)$$

where $q(\theta', \boldsymbol{s}' | \theta, \boldsymbol{s}, \alpha \cdot \mathbf{1}_N)$ is the probability that the user has rating θ', and the rating distribution is $\boldsymbol{s}'$ in the next period given the user's current rating θ, the current rating distribution $\boldsymbol{s}$, and the action profile $\alpha \cdot \mathbf{1}_N$, and $v_\theta^{\pi_0}(\boldsymbol{s})$ is the continuation payoff of the users with rating θ starting from the initial rating distribution $\boldsymbol{s}$.

The differences between (25) and the Bellman equations are (1) the "value" of state $\boldsymbol{s}$ in (25) is a vector comprised of rating-1 and rating-0 users' values, compared to scalar values in Bellman equations, and (2) the value of state $\boldsymbol{s}$ is not fixed in (25), because the action α_0 taken under state $\boldsymbol{s}$ is not fixed in a nonstationary strategy (this is also the key difference from the analysis of stationary strategies; see (12) where the action taken at state $\boldsymbol{s}$ is fixed to be $\pi_0(\boldsymbol{s})$). In addition, for an equilibrium recommended strategy, the decomposition needs to satisfy the following incentive constraints: for all α,

$$v_\theta^{\pi_0}(\boldsymbol{s}) \geq (1-\delta) \cdot u(\theta, \boldsymbol{s}, (\alpha, \alpha_0 \cdot \mathbf{1}_{N-1})) + \delta \cdot \sum_{\theta'=0}^{1} \sum_{\boldsymbol{s}'} \rho(\theta', \boldsymbol{s}' | \theta, \boldsymbol{s}, (\alpha, \alpha_0 \cdot \mathbf{1}_{N-1})) \cdot v_{\theta'}^{\pi_0}(\boldsymbol{s}'). \qquad (26)$$

To analyze nonstationary strategies, we use the theory of self-generating sets. Note, however, that Abreu et al. [1990] do not tell us how to construct a self-generating


Rating-1 users' continuation payoff
Payoffs of the fair action
$\left(\frac{s_0-1}{N-1}b-c, b-\frac{s_1-1}{N-1}c\right)$
(b-c, b-c)
Payoffs of the altruistic action
(0, 0)
Payoffs of the selfish action
Rating-0 users' continuation payoff
Rating-1 users' continuation payoff
Feasible payoffs under (N-1,1)
Feasible payoffs under (s_0,s_1)
Feasible payoffs under (1,N-1)
(0, 0)
Rating-0 users' continuation payoff

(a) Feasible payoffs in one state (rating distribution (s_0,s_1)).

(b) Common feasible payoffs in all the states (rating distributions).

Fig. 3. Illustration of how to build the self-generating set. The left figure shows the set of feasible payoffs in one state (i.e., under the rating distribution (s_0, s_1)). The right figure shows the sets of feasible payoffs in different states (i.e., rating distributions) and their intersection, namely, the set of common feasible payoffs in all the states (i.e., under all the rating distributions).

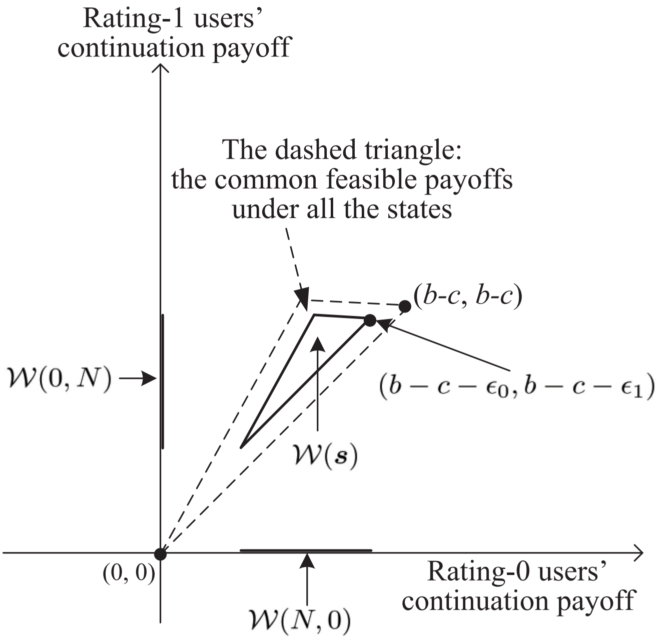

Rating-1 users' continuation payoff
The dashed triangle: the common feasible payoffs under all the states
(b-c, b-c)
$\mathcal{W}(0, N)$
$(b-c-\epsilon_0, b-c-\epsilon_1)$
$\mathcal{W}(\boldsymbol{s})$
(0, 0)
Rating-0 users' continuation payoff
$\mathcal{W}(N, 0)$

Fig. 4. Illustration of the self-generating set, which is a triangle within the set of common feasible payoffs in Figure 3.

set, which is exactly the major difficulty to overcome in our proof. In our proof, we construct the following self-generating set. First, since the strategies depend on rating distributions only, we let $\mathcal{W}(\boldsymbol{\theta}) = \mathcal{W}(\boldsymbol{\theta}')$ for any $\boldsymbol{\theta}$ and $\boldsymbol{\theta}$ that have the same rating distribution. Hence, in the rest of the proof sketch, we write the self-generating set as $\{\mathcal{W}(\boldsymbol{s})\}_{\boldsymbol{s}}$, which is illustrated in Figure 3 and Figure 4. Figure 3 shows how to construct the self-generating set. The left of Figure 3 shows the feasible payoffs in one state, and the right shows the common feasible payoffs in all the states (we consider the common feasible payoffs such that we can use the same $\mathcal{W}(\boldsymbol{s})$ under all the states $\boldsymbol{s}$). The self-generating set is a subset of the common feasible payoffs, as illustrated in Figure 4. When the users have different ratings (i.e., $1 \leq s_0 \leq N-1$), the set $\mathcal{W}(\boldsymbol{s})$ is the triangle shown in Figure 4, which is congruent to the triangle of common feasible payoffs (shown in dashed lines), and has the upper-right vertex at $(b-c-\epsilon_0, b-c-\epsilon_1)$ with $\epsilon_0, \epsilon_1 \leq \xi$. We have the analytical expression for the triangle in Online Appendix B. When all the users have the same rating (i.e., $s_0 = 0$ or $s_0 = N$), only one component in $\boldsymbol{v}(\boldsymbol{s})$ is relevant. Hence, the sets $\mathcal{W}((N, 0))$ and $\mathcal{W}((0, N))$ are line segments determined by the ranges of v_0 and v_1 in the triangle, respectively.

In addition, we simplify the decomposition (25) and (26) by letting the continuation payoffs $v_{\theta'}^{\pi_0}(\boldsymbol{s}') = v_{\theta'}^{\pi_0}$ for all $\boldsymbol{s}'$. Hence, for a given $\boldsymbol{s}$ and a payoff vector $\boldsymbol{v}(\boldsymbol{s})$, the continuation payoffs $\boldsymbol{v}' = (v_0', v_1')$ can be determined by solving the following two linear equations.

$$\begin{cases} v_0(\boldsymbol{s}) = (1-\delta)u(0, \boldsymbol{s}, \alpha_0 \mathbf{1}_N) + \delta \sum_{\theta'=0}^{1} q(\theta' | 0, \alpha_0 \mathbf{1}_N) v_{\theta'}' \\ v_1(\boldsymbol{s}) = (1-\delta)u(1, \boldsymbol{s}, \alpha_0 \mathbf{1}_N) + \delta \sum_{\theta'=0}^{1} q(\theta' | 1, \alpha_0 \mathbf{1}_N) v_{\theta'}' \end{cases} \tag{27}$$

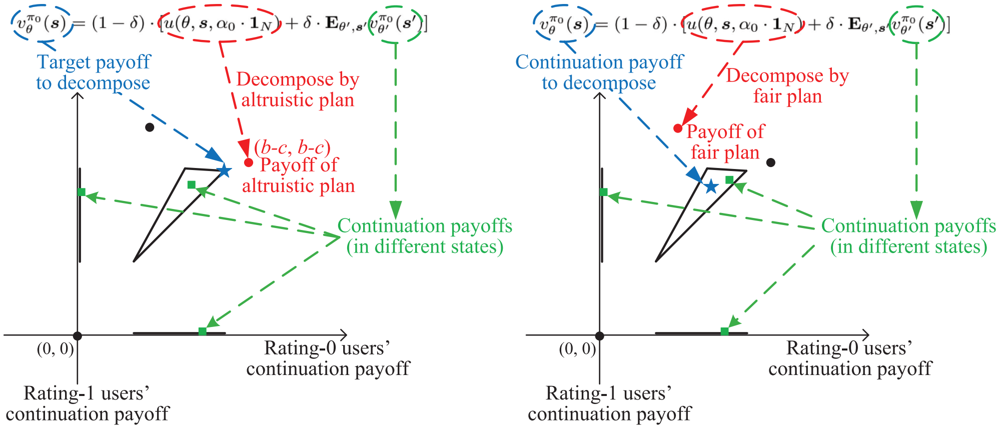
where $q(\theta'|\theta, \alpha_0)$ is the probability that the next rating is θ' for a user with rating θ under the plan profile $\alpha_0 \cdot \mathbf{1}_N$.

Based on this simplification, the collection of sets $\{\mathcal{W}(\boldsymbol{s})\}_{\boldsymbol{s}}$ in Figure 4 is a self-generating set, if for any $\boldsymbol{s}$ and any payoff vector $\boldsymbol{v}(\boldsymbol{s}) \in \mathcal{W}(\boldsymbol{s})$, we can find a plan α_0 such that the continuation payoffs $\boldsymbol{v}'$ calculated from (27) lie in the triangle and satisfy the incentive constraints in (26).

In summary, we can prove that the collection of sets $\{\mathcal{W}(\boldsymbol{s})\}_{\boldsymbol{s}}$ illustrated in Figure 4 is a self-generating set under certain conditions. Specifically, given a performance loss ξ, we construct the corresponding $\{\mathcal{W}(\boldsymbol{s})\}_{\boldsymbol{s}}$ and prove that it is a self-generating set under the following conditions: (1) the discount factor $\delta \geq \underline{\delta}(\varepsilon, \xi)$ with $\underline{\delta}(\varepsilon, \xi)$ defined in Appendix A, and (2) the three conditions on the rating update rule in Theorem 5.2. This proves the first part of Theorem 5.2.

The corresponding recommended strategy can be determined based on the decomposition of payoffs. Specifically, given the current rating distribution $\boldsymbol{s}$ and the current expected payoffs $\boldsymbol{v}(\boldsymbol{s}) \in \mathcal{W}(\boldsymbol{s})$, we find a plan α_0 such that the continuation payoffs $\boldsymbol{v}'$ calculated from (27) lie in the triangle and satisfy the incentive constraints. The decomposition is illustrated in Figure 5. One important issue in the decomposition is which plan should be used to decompose the payoff. We prove that we can determine the plan in the following way (illustrated in Figure 6). When the users have different ratings, choose the altruistic plan α^{a} when $\boldsymbol{v}(\boldsymbol{s})$ lies in the region marked by "a" in the triangle in Figure 6(b), and choose the fair plan α^{f} otherwise. When the users have the same rating 0 or 1, choose the altruistic plan α^{a} when $v_0(\boldsymbol{s})$ or $v_1(\boldsymbol{s})$ lies in the region marked by "a" in the line segment in Figure 6(a) or Figure 6(c), and choose the selfish plan α^{s} otherwise. Note that we can analytically determine the line that separates the two regions in the triangle and the threshold that separates the two regions in the line segment (analytical expressions are omitted due to space limitation; see Online Appendix C for details). This decomposition is repeated and is used to determine the recommended plan in each period based on the current rating distribution $\boldsymbol{s}$ and the current expected payoffs to achieve $\boldsymbol{v}(\boldsymbol{s})$. The procedure thus described is exactly the algorithm for constructing the recommended strategy, which is described in Algorithm 1. Due to space limitation, Algorithm 1 is illustrative but not specific. The detailed table that describes the algorithm can be found in Online Appendix C. □

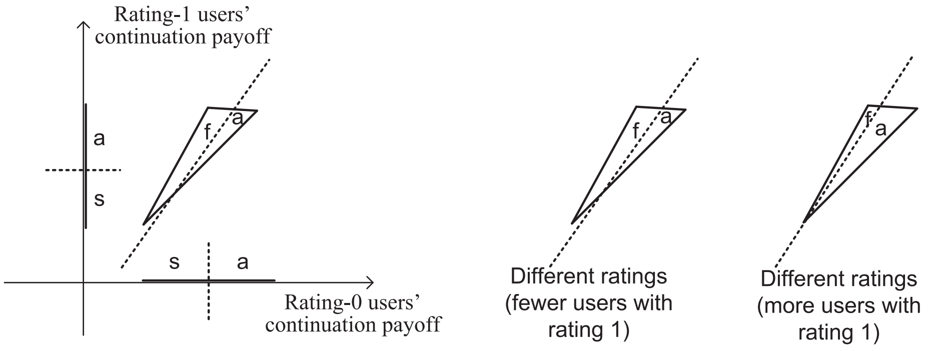
Theorem 5.2 proves that for any rating update error $\varepsilon \in [0, 0.5)$, we can design an asymptotically-optimal rating mechanism. The design of the asymptotically optimal rating mechanism consists of two parts. The first part is to design the rating update


$v_\theta^{\pi_0}(\boldsymbol{s}) = (1-\delta)\cdot[u(\theta,\boldsymbol{s},\alpha_0\cdot\mathbf{1}_N) + \delta\cdot\mathbf{E}_{\theta',\boldsymbol{s}'}v_{\theta'}^{\pi_0}(\boldsymbol{s}')]$
Target payoff to decompose
Decompose by altruistic plan
(b-c, b-c)
Payoff of altruistic plan
Continuation payoffs (in different states)
(0, 0)
Rating-0 users' continuation payoff
Rating-1 users' continuation payoff
$v_\theta^{\pi_0}(\boldsymbol{s}) = (1-\delta)\cdot[u(\theta,\boldsymbol{s},\alpha_0\cdot\mathbf{1}_N) + \delta\cdot\mathbf{E}_{\theta',\boldsymbol{s}'}v_{\theta'}^{\pi_0}(\boldsymbol{s}')]$
Continuation payoff to decompose
Decompose by fair plan
Payoff of fair plan
Continuation payoffs (in different states)
(0, 0)
Rating-0 users' continuation payoff
Rating-1 users' continuation payoff

(a) Decomposition in period 0.

(b) Decomposition in period 1 (when users have different ratings in period 1).

Fig. 5. The decomposition of payoffs. The left figure shows the decomposition in period 0 when the payoff to decompose is the target payoff $(b-c-\epsilon_0, b-c-\epsilon_1)$; the right figure shows the decomposition in period 1 when the payoff to decompose is the continuation payoff starting from period 1 and when the users have different ratings.


Rating-1 users' continuation payoff
a
s
f
a
s
a
Rating-0 users' continuation payoff
f
a
Different ratings (fewer users with rating 1)
f
a
Different ratings (more users with rating 1)

(a) How to choose the plan.

(b) The differences under different rating distributions.

Fig. 6. Illustration of how to choose the plan in order to decompose a given payoff. Each self-generating set is partitioned into two parts. In each period, a recommended plan (the altruistic plan "a," the fair plan "f," or the selfish plan "s") is chosen, depending on which part of the self-generating set the expected payoffs fall into.

rule. First, we should give incentives for the users to provide high-quality service by setting β_θ^+, the probability that the rating goes up when the service quality is not lower than the recommended quality, to be larger than $1-\beta_\theta^-$, the probability that the rating goes up when the service quality is lower than the recommended quality. Second, for a user with rating 1, the expected probability that its rating goes up when it complies should be larger than the threshold specified in Condition 2 ($x_{s_1}^+ > x_1^+$ implies that $x_{s_1}^+$ is larger than the threshold, too). This gives users incentives to obtain rating 1. Meanwhile, for a user with rating 0, the expected probability that its rating goes up when it complies, x_0^+, should be smaller than the threshold specified in Condition 3.

This provides necessary punishment for a user with rating 0. Note that Conditions 2 and 3 imply that $x_1^+ > x_0^+$. In this way, a user would prefer to have rating 1.

The second part is to construct the equilibrium recommended strategy. Theorem 5.2 proves that for any feasible discount factor δ no smaller than the lower-bound discount factor $\underline{\delta}(\varepsilon, \xi)$ defined in Appendix A, we can construct the corresponding recommended strategy such that each user can achieve an discounted average payoff of at least $b-c-\xi$. Now we show how to construct the recommended strategy. Note that determining the lower-bound discount factor $\underline{\delta}(\varepsilon, \xi)$ analytically is important for constructing the equilibrium $(\pi_0, \pi_0 \cdot \mathbf{1}_N)$, because a feasible discount factor is needed to determine the strategy. In Fudenberg et al. [1994] and Hörner et al. [2011], the lower bound for the discount factor cannot be obtained analytically. Hence, their results are not constructive.

Algorithm 1 constructs the optimal recommended strategy and works as follows. In each period, the algorithm updates the continuation payoffs (v_0, v_1) and determines the recommended plan based on the current rating distribution and the continuation payoffs. In Figure 6, we illustrate which plan to recommend based on where the continuation payoffs locate in the self-generating sets. Specifically, each set $\mathcal{W}(\boldsymbol{s})$ is partitioned into two parts (the partition lines can be determined analytically; see Online Appendix C for the analytical expressions). When all the users have rating 0 (or 1), we recommend the altruistic plan α^{a} if the continuation payoff v_0 (or v_1) is large, and the selfish plan α^{s} otherwise. When the users have different ratings, we recommend the altruistic plan α^{a} when (v_0, v_1) lies in the region marked by "a" in the triangle in Figure 6(b), and the fair plan α^{f} otherwise. Note that the partition of $\mathcal{W}(\boldsymbol{s})$ is different under different rating distributions (e.g., the region in which the altruistic plan is

ALGORITHM 1: Construction of Recommended Strategies

```
Input: $b, c, \varepsilon, \xi; \tau(\varepsilon), \delta \geq \underline{\delta}(\varepsilon, \xi); \boldsymbol{\theta}^0$
Output: The recommended strategy
initialization: $t = 0$; $\epsilon_0 = \xi$; $\epsilon_1 = \epsilon_0/(1 + \frac{\kappa_2}{\kappa_1})$; $v^\theta = b - c - \epsilon_\theta$; $\boldsymbol{\theta} = \boldsymbol{\theta}^0$;
repeat
    if $s_1(\boldsymbol{\theta}) = 0$ then
        if $(v^0, v^1)$ lies in region "a" of the horizontal line segment in Figure 6(a) then
            choose recommended plan $\alpha^{\mathrm{a}}$;
        else
            choose recommended plan $\alpha^{\mathrm{s}}$;
        end
    else if $s_1(\boldsymbol{\theta}) = N$ then
        if $(v^0, v^1)$ lies in region "a" of the vertical line segment in Figure 6(a) then
            choose recommended plan $\alpha^{\mathrm{a}}$;
        else
            choose recommended plan $\alpha^{\mathrm{s}}$;
        end
    else
        if $(v^0, v^1)$ lies in region "a" of the triangle in Figure 6(a) then
            choose recommended plan $\alpha^{\mathrm{a}}$;
        else
            choose recommended plan $\alpha^{\mathrm{f}}$;
        end
    end
until $\varnothing$;
determine the continuation payoffs $(v_0', v_1')$ according to (27);
$t \leftarrow t + 1$; determine the rating profile $\boldsymbol{\theta}^t$; set $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^t$; $(v_0, v_1) \leftarrow (v_0', v_1')$;
```

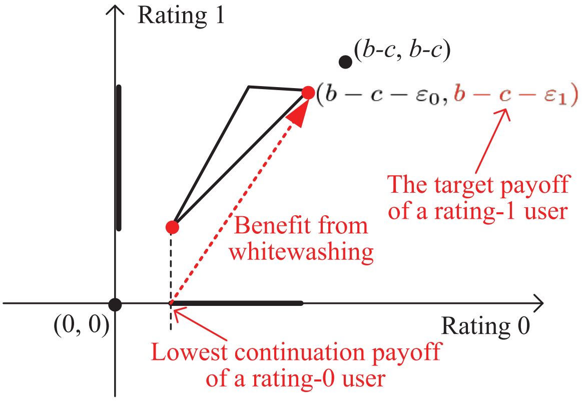

Rating 1
$(b\text{-}c, b\text{-}c)$
$(b-c-\varepsilon_0, b-c-\varepsilon_1)$
The target payoff of a rating-1 user
Benefit from whitewashing
(0, 0)
Rating 0
Lowest continuation payoff of a rating-0 user

Fig. 7. Illustration of the target payoff of a rating-1 user and the lowest continuation payoff of a rating-0 user.

chosen is larger when more users have rating 1). Figure 6 also illustrates why the strategy is nonstationary: the recommended plan depends on not only the current rating distribution $\boldsymbol{s}$, but also which region of $\mathcal{V}(\boldsymbol{s})$ the continuation payoffs (v_0, v_1) lie in.

Complexity. Although the design of the optimal recommended strategy is complicated, the implementation is simple. The computational complexity in each period comes from (1) identifying which region the continuation payoffs lie in, which is simple because the regions are divided by a straight line that is analytically determined, and (2) updating the continuation payoffs (v_0, v_1) by (27), which can be easily done by solving a set of two linear equations with two variables. The memory complexity is also low: because we summarize the history of past states by the continuation payoffs (v_0, v_1), the protocol does not need to store all the past states.

5.3. Whitewashing-Proofness

An important issue in rating mechanisms is whitewashing, namely, users with low ratings can register as a new user to clear its history of bad behaviors. We say a rating mechanism is whitewashing-proof if the cost of whitewashing (e.g., creating a new account) is higher than the benefit from whitewashing. The benefit from whitewashing is determined by the difference between the current continuation payoff of a low-rating user and the target payoff of a high-rating user. Since this difference is relatively small under the proposed rating mechanism, the proposed mechanism is robust to whitewashing.

Proposition 5.3. *Given the performance loss tolerance $\xi > 0$, the proposed rating mechanism is whitewashing-proof if the cost of whitewashing is larger than* $(1 - \frac{1}{\kappa_1} - \frac{1}{\kappa_2}) \cdot \xi$.

Proof. We illustrate the proof using Figure 7. In Figure 7, we show the self-generating set again and point out the target payoff of a rating-1 user and the lowest continuation payoff of a rating-0 user. The difference between these two payoffs is the highest benefit that a rating-0 user can get by whitewashing. Simple calculation tells us that the difference is $(1 - \frac{1}{\kappa_1} - \frac{1}{\kappa_2}) \cdot \xi$, which completes the proof of Proposition 5.3. □

6. SIMULATION RESULTS

We compare against the rating mechanism with threshold-based stationary recommended strategies. In particular, we focus on threshold-based stationary recommended

Period	0	1	2	3	4	5	6
Rating distribution	(0,10)	(1,9)	(3,7)	**(6,4)**	(5,5)	***(6,4)***	(2,8)
Threshold AF	A	A	A	**F**	A	***F***	A
Threshold AS	A	A	A	**S**	A	***S***	A
Threshold FS	F	F	F	**S**	F	***S***	F
Proposed	A	A	A	**F**	A	***A***	A

↑
The proposed strategy does not punish,
because users have been punished in period 3

Fig. 8. Evolution of states and recommended plans taken in different rating mechanisms.

strategies that use two plans. In other words, one plan is recommended when the number of rating-1 users is no smaller than the threshold, and the other plan is recommended otherwise. In particular, we consider threshold-based stationary recommended strategies restricted on A^{af}, A^{as}, and A^{fs}, and call them "Threshold AF," "Threshold AS," and "Threshold FS," respectively. We focus on threshold-based strategies because it is difficult to find the optimal stationary strategy in general when the number of users is large (the number of stationary strategies grows exponentially with the number of users). In our experiments, we fix the following system parameters: $N = 10, b = 3, c = 1$.

In Figure 8, we first illustrate the evolution of the states and the recommended plans taken under the proposed rating mechanism and the rating mechanism with the Threshold AF strategy. The threshold is set to be 5. Hence, it recommends the altruistic plan when at least half of the users have rating 1, and recommends the fair plan otherwise. We can see that in the proposed strategy, the plans taken can be different at the same state. In particular, in "bad" states (6,4) and (7,3) at time slot 3 and 5, respectively, the proposed rating mechanism may recommend the fair plan (as a punishment) and the altruistic plan (i.e., do not punish because the punishment happens in time slot 3), while the stationary mechanism always recommends the fair plan to punish the low-rating users.

Then in Figure 9, we show the price of stationarity of three representative stationary rating mechanisms: the one with the optimal Threshold AF strategy, the one with the optimal Threshold AS strategy, and the one with the optimal Threshold FS strategy. We can see from Figure 9 that as the rating update error increases, the efficiency of stationary rating mechanisms decreases and drops to 0 when the error probability is large (e.g., when $\varepsilon > 0.4$). In contrast, the proposed rating mechanism can achieve arbitrarily close to the social optimum.

In Figure 10, we illustrate the lower-bound discount factors under different performance loss tolerances and rating update errors. As expected, when the performance loss tolerance becomes larger, the lower-bound discount factor becomes smaller. What is unexpected is how the lower-bound discount factor changes with the rating update error. Specifically, the lower-bound discount factor decreases initially with the increase of the error, and then increases with the error. It is intuitive to see the discount factor increase with the rating update error, because the users need to be more patient when the rating update is more erroneous. The initial decrease of the discount factor in the error can be explained as follows. If the rating update error is extremely small, the punishment for the rating-0 users in the optimal rating update rule needs to be very severe (i.e., a smaller β_0^+ and a larger β_0^-). Hence, once a user is assigned with rating 0, it needs to be more patient to carry out the severe punishment (i.e., weigh the future payoffs more).

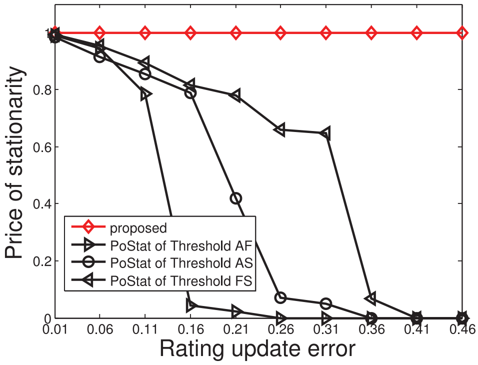


Fig. 9. Price of stationarity of different stationary rating mechanisms under different rating update errors.

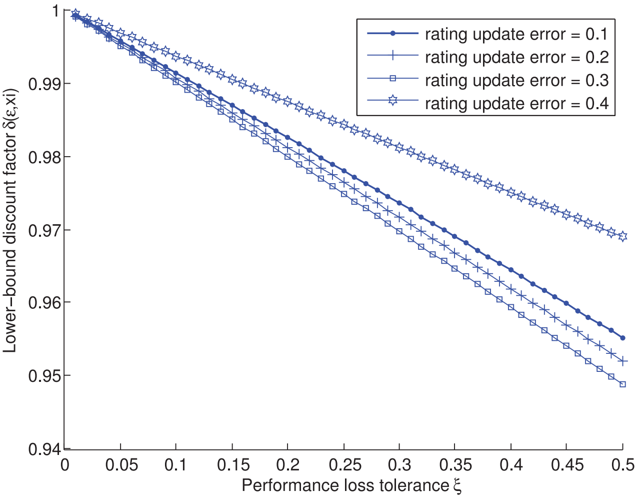


Fig. 10. Lower-bound discount factors under different performance loss tolerances and rating update errors.

Finally, we illustrate the robustness of the proposed mechanisms with respect to the estimation of rating update errors. Suppose that the rating update error is ε. However, the designer cannot accurately measure this error. Under the estimated error $\hat{\varepsilon}$, the rating mechanism will construct another recommended strategy. In Figure 11, we illustrate the performance gain/loss in terms of social welfare under the estimated error $\hat{\varepsilon}$, when the rating update error is ε. We can see that there is less than 5% performance variance when the estimation inaccuracy is less than 50%. The performance variance is larger when the rating update error is larger.

7. CONCLUSION

In this article, we propose a design framework for simple binary rating mechanisms that can achieve social optimum in the presence of rating update errors. We provide design guidelines for the optimal rating update rules, and an algorithm to construct the optimal nonstationary recommended strategy. The key design principles that enable the rating mechanism to achieve the social optimum are the differential punishments and the nonstationary strategies that reduce the performance loss while providing enough incentives. We also reduce the complexity of computing the recommended

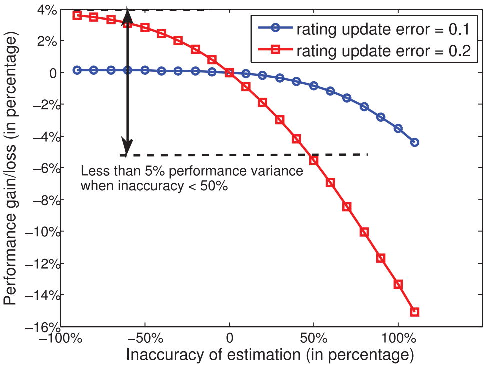


Fig. 11. Performance gain/loss (in percentage) under different inaccuracy of estimation (in percentage).

strategy by proving that using three recommended plans is enough to achieve social optimum. Our proposed rating mechanism is the first that can achieve social optimum even when the rating update errors are large. Simulation results demonstrate the significant performance gain of the proposed rating mechanism over the state-of-the-art mechanisms, especially when the rating update error is large.

APPENDIX

A. ANALYTICAL EXPRESSION OF THE LOWER-BOUND DISCOUNT FACTOR

The lower-bound discount factor $\underline{\delta}(\varepsilon, \xi)$ is the maximum of three critical discount factors, namely, $\underline{\delta}(\varepsilon, \xi) \triangleq \max\{\delta_1(\varepsilon, \xi), \delta_2(\varepsilon, \xi), \delta_3(\varepsilon, \xi)\}$, where

$$\delta_1(\varepsilon, \xi) \triangleq \max_{\theta \in \{0,1\}} \frac{c}{c + (1-2\varepsilon)(\beta_\theta^+ - (1-\beta_\theta^-))\big(\xi \frac{\kappa_2}{\kappa_1+\kappa_2}\big)},$$

$$\delta_2(\varepsilon, \xi) \triangleq \max_{s_1 \in \{1,\ldots,N-1\}: \frac{s_1}{N-1} b + \frac{N-s_1}{N-1} c > \xi \frac{\kappa_2}{\kappa_1+\kappa_2}} \left\{ \frac{\xi \frac{\kappa_2}{\kappa_1+\kappa_2} - \left(\frac{s_1}{N-1} b + \frac{N-s_1}{N-1} c\right)}{\left(\xi \frac{\kappa_2}{\kappa_1+\kappa_2}\right)\left(x_{s_1}^+ - x_0^+\right) - \left(\frac{s_1}{N-1} b + \frac{N-s_1}{N-1} c\right)} \right\},$$

and

$$\delta_3(\varepsilon, \xi) \triangleq \max_{\theta \in \{0,1\}} \frac{b - c + c \frac{x_\theta^+}{(1-2\varepsilon)\left[\beta_\theta^+ - (1-\beta_\theta^-)\right]}}{b - c + \frac{c \cdot x_\theta^+}{(1-2\varepsilon)\left[\beta_\theta^+ - (1-\beta_\theta^-)\right]} - \frac{(1+\kappa_1)(\xi \frac{\kappa_2}{\kappa_1+\kappa_2}) - z_2}{\kappa_1} - z_3}, \tag{28}$$

where $z_2 \triangleq -\kappa_1(b-c) + \kappa_1(1 - 1/\kappa_2)\xi + \xi \frac{\kappa_1}{\kappa_2}/(1 + \frac{\kappa_2}{\kappa_1})$, and $z_3 \triangleq z_2/(\kappa_1 + \kappa_2)$. Note that $\frac{(1+\kappa_1)(\xi \frac{\kappa_2}{\kappa_1+\kappa_2}) - z_2}{\kappa_1} + z_3 < 0$. We can see from these expressions that $\delta_1(\varepsilon, \xi) < 1$ and $\delta_2(\varepsilon, \xi) < 1$, as long as $\xi > 0$. For $\delta_3(\varepsilon, \xi)$, simple calculations tell us that ξ appears in the denominator in the form of $-\frac{(2\kappa_1+\kappa_2)\kappa_2}{(\kappa_1+\kappa_2)^2\kappa_1} \cdot \xi$. Since $\kappa_1 > 0$ and $\kappa_2 > 0$, we know that $-\frac{(2\kappa_1+\kappa_2)\kappa_2}{(\kappa_1+\kappa_2)^2\kappa_1} < 0$. Hence, $\delta_3(\varepsilon, \xi)$ is increasing in ξ. As a result, $\delta_3(\varepsilon, \xi) < 1$ as long as ξ is small enough.

Note that all the critical discount factors can be calculated analytically. Specifically, $\delta_1(\varepsilon, \xi)$ and $\delta_3(\varepsilon, \xi)$ are the maximum of two analytically-computed numbers, and $\delta_2(\varepsilon, \xi)$ is the maximum of at most $N-1$ analytically-computed numbers.

ACKNOWLEDGMENTS

We would like to thank Dr. Yu Zhang (Microsoft) for his very insightful initial discussions on rating mechanisms, and Prof. William Zame and Simpson Zhang (Department of Economics, UCLA) for their helpful comments that led to improvement of this article.

REFERENCES

Dilip Abreu, David Pearce, and Ennio Stacchetti. 1990. Toward a theory of discounted repeated games with imperfect monitoring. *Econometrica* 58, 5 (1990), 1041–1063.

Alberto Blanc, Yi-Kai Liu, and Amin Vahdat. 2005. Designing incentives for peer-to-peer routing. In *Proceedings of the IEEE INFOCOM*. 374–385. DOI:http://dx.doi.org/10.1007/978-3-642-10841-9_23

Pedro Dal Bó. 2007. Social norms, cooperation and inequality. *Econ. Theor.* 30, 1 (2007), 89–105.

Joyee Deb. 2012. Cooperation and community responsibility: A folk theorem for random matching games with names. *Rev. Econ. Stud.*

Chrysanthos Dellarocas. 2005. Reputation mechanism design in online trading environments with pure moral hazard. *Inform. Syst. Res.* 16, 2 (2005), 209–230. DOI:http://dx.doi.org/10.1016/j.geb.2006.11.002

Glenn Ellison. 1994. Cooperation in the prisoner's dilemma with anonymous random matching. *Rev. Econ. Stud.* 61, 3 (1994), 567–588.

Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. 2004. Robust incentive techniques for peer-to-peer networks. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC'04)*. ACM, New York, NY, 102–111. DOI:http://dx.doi.org/10.1145/988772.988788

Drew Fudenberg, David Levine, and Eric Maskin. 1994. The folk theorem with imperfect public information. *Econometrica* 62, 5 (1994), 997–1039.

Kevin Hasker. 2007. Social norms and choice: A general folk theorem for repeated matching games with endogenous matching rules. *Int. J. Game Theor.* 36 (2007), 137–146.

Johannes Hörner, Takuo Sugaya, Satoru Takahashi, and Nicolas Vielle. 2011. Recursive methods in discounted stochastic games: An algorithm for $\delta \rightarrow 1$ and a folk theorem. *Econometrica* 79, 4 (2011), 1277–1318.

Michihiro Kandori. 1992. Social norms and community enforcement. *Rev. Econ. Stud.* 59, 1 (1992), 63–80.

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 (CHI'08)*. ACM, New York, NY, 453–456. DOI:http://dx.doi.org/10.1145/1357054.1357127

George Mailath and Larry Samuelson. 2006. *Repeated Games and Reputations: Long-run Relationships*. Oxford University Press, Oxford.

Masahiro Okuno-Fujiwara and Andrew Postlewaite. 1995. Social norms and random matching games. *Games Econ. Behav.* 9, 1 (1995), 79–109.

Satoru Takahashi. 2010. Community enforcement when players observe partners' past play. *J. Econ. Theor.* 145, 1 (2010), 42–62.

Yu Zhang, Jaeok Park, and Mihaela van der Schaar. 2014. Rating protocols in online communities. *ACM Trans. Econ. Comput.* 2, 1, Article 4 (March 2014), 34 pages. DOI:http://dx.doi.org/10.1145/2560794

Yu Zhang and Mihaela van der Schaar. 2012a. Peer-to-peer multimedia sharing based on social norms. *Signal Process.: Image Commun.* (Special Issue on Advances in video streaming for P2P network) 27, 5 (2012), 383–400.

Yu Zhang and Mihaela van der Schaar. 2012b. Reputation-based incentive protocols in crowdsourcing applications. In *Proceedings of the IEEE INFOCOM*. 2140–2148. DOI:http://dx.doi.org/10.1007/978-3-642-10841-9_23

Received August 2013; accepted December 2014