Algorithmic Mechanisms
for Reliable Master-Worker Internet-based Computing *

Evgenia Christoforou
University of Cyprus
evgenia.christoforou@gmail.com

Antonio Fernández Anta
Inst. IMDEA Networks and URJC
antonio.fernandez@imdea.org

Chryssis Georgiou
University of Cyprus
chryssis@cs.ucy.ac.cy

Miguel A. Mosteiro
Rutgers University and URJC
mosteiro@cs.rutgers.edu

Abstract

We consider Internet-based master-worker computations, where a master processor assigns, across the Internet, a computational task to a set of untrusted worker processors, and collects their responses. Examples of such computations are the "@home" projects such as SETI. Prior work dealing with Internet-based task computations has either considered only rational, or only malicious and altruistic workers. Altruistic workers always return the correct result of the task, malicious workers always return an incorrect result, and rational workers act based on their self-interest. However, in a massive computation platform, such as the Internet, it is expected that all three type of workers coexist. Therefore, in this work we study Internet-based master-worker computations in the presence of malicious, altruistic, and rational workers. A stochastic distribution of the workers over the three types is assumed. In addition, we consider the possibility that the communication between the master and the workers is not reliable, and that workers could be unavailable. Considering all the three types of workers renders a combination of game-theoretic and classical distributed computing approaches to the design of mechanisms for reliable Internet-based computing. Indeed, in this work we design and analyze two algorithmic mechanisms to provide appropriate incentives to rational workers to act correctly, despite the malicious' workers actions and the unreliability of the network. Only when necessary, the incentives are used to force the rational players to a certain equilibrium (which forces the workers to be truthful) that overcomes the attempt of the malicious workers to deceive the master. Finally, the mechanisms are analyzed in two realistic Internet-based master-worker settings, a SETI-like one and a contractor-based one, such as Amazon's Mechanical Turk. We also present plots that illustrate the trade-offs between reliability and cost, under different system parameters.

Technical Report TR-11-09
Department of Computer Science
University of Cyprus
December 2011

*This work is supported in part by the Cyprus Research Promotion Foundation grant ΤΠΕ/ΠΛΗΡΟ/0609(ΒΕ)/05, Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TIN2008–06735-C02-01, and NSF grant 0937829. Parts of this work reporting preliminary results have appeared in the Proceedings of IPDPS 2010 and NCA 2011.

1 Introduction

1.1 Motivation and Prior Work

As an alternative to expensive supercomputing parallel machines, the Internet has recently become feasible as a computational platform for processing complex computational jobs. Several Internet-oriented systems and protocols have been designed to operate on top of this global computation infrastructure; examples include Grid systems [16, 55], the "@home" projects [5], such as SETI [36], Amazon's Mechanical Turk [4], and peer-to-peer computing–P2PC [22,58]. Although the potential is great, the use of Internet-based computing is limited by the untrustworthy nature of the platform's components [5,25,30]. Let us take SETI as an example. In SETI, data is distributed for processing to millions of voluntary machines around the world. At a conceptual level, in SETI there is a machine, call it the *master*, that sends jobs, across the Internet, to these computers, call them the *workers*. These workers execute and report back the result of the task computation. However, these workers are not trustworthy, and hence might report incorrect results. In SETI, the master attempts to minimize the impact of these bogus results by assigning the same task to several workers and comparing their outcomes (that is, redundant task allocation is employed [5]), but there are also other methods [13, 34, 57].

This problem has recently been studied under two different views: from a "classical" distributed computing view [20, 35, 51] and from a game-theoretic view [21, 58]. Under the first view, the workers are classified as either *malicious* (Byzantine) or *altruistic*, based on a predefined behavior. The malicious workers have a "bad" behavior which results in reporting an incorrect result to the master. This behavior is, for example, due to a hardware or a software error or due to an ill-state of the worker such as being a wrongdoer intentionally. Altruistic workers exhibit a "good" behavior, that is, they compute and return the correct task result. From the perspective of the master, the altruistic workers are the "correct" ones. Under this view, "classical" distributed computing models are defined (e.g., a fixed bound on the probability of a worker being malicious is assumed) and typical malicious-tolerant voting protocols are designed.

Under the game-theoretic view, workers act on their own *self-interest* and they do not have an a priori established behavior, that is, they are assumed to be *rational* [2, 25, 52]. In other words, the workers decide on whether they will be *honest* and report the correct task result, or *cheat* and report a bogus result, depending on which strategy increases their benefit or *utility.* Under this view, Algorithmic Mechanisms [2, 11, 46] are employed, where games are designed to provide the necessary incentives so that processors' interests are best served by acting "correctly." In particular, the master provides some reward (resp. penalty) should a worker be honest (resp. cheat). The design objective is for the master to force a desired unique *Nash equilibrium* (NE) [45], i.e., a strategy choice by each worker such that none of them has incentive to change it. That Nash equilibrium is the one in which the master achieves a desired probability of obtaining the correct task result. (It is known that Nash Equilibria do not always lead to optimal solutions for rational players, but as argued in [47, Chapter 1], it is a "safe" way for the players to obtain high utility satisfaction, and more importantly, a Nash Equilibrium is *stable*, that is, once proposed, the players do not want to individually deviate.)

The above views could complement one another, if a certain computation includes only malicious and altruistic workers, or only rational workers. However, the pragmatic situation on the Internet is different: all three type of workers might co-exist in a given computation. One could assume that all workers are rational, but what, for example, if a software bug occurs that makes a worker deviate from its protocol, and hence compute and return an incorrect result? This worker is no longer exhibiting a rational behavior, but rather an erroneous or irrational one. From the master's point of view such behavior can be seen as malicious.

In this paper we consider the possibility that all three types of workers co-exist. Furthermore we consider the possibility that the communication between the master and workers is not reliable. This communication uncertainty can either be due to communication-related failures or due to workers being slow in processing messages (or even crashing while doing so). For instance, Heien at al. [30] have found that in BOINC only around 5% of the workers are available more than 80% of the time, and that half of the workers are available less than 40% of the time. This fact, combined with the length of the computation [33], justifies the interest of considering in the Internet-based master-worker framework the possibility of workers not replying. In order to introduce this possibility in our model, we assume that there is some positive probability that the master does not receive a reply from a given worker. Since it is possible for a worker's reply not to reach the master, we also allow workers to abstain from the computation. Imagine the situation where a rational worker decides to compute and truthfully return the task result but its reply is not received by the master. As we explain later (Section 2), in this case the master provides no reward to the worker, while the worker has incurred the cost of performing the task. Hence, it is only natural to provide to the workers the choice of not replying (especially when the reliability of the network is low). This issue makes the task of the master even more challenging, as it needs to provide the necessary incentives to encourage rational workers to reply and do so truthfully, even in the presence of low network reliability.

1.2 Contributions

We study Internet-based master-worker computations under the assumption that each worker's behavior is either malicious, altruistic or rational. Furthermore, we also assume that a worker's output may never be received. The presence of all three types of workers, naturally renders a combination of game-theoretic and classical approaches to the design of algorithmic mechanisms for distributed computing. Our model captures the hardest shortcomings of an Internet-based platform, yielding mechanisms that are resilient to undesired worker behavior and uncertainty of reply. In particular our contributions are as follows:

- A collection of realistic payoff parameters and reward models are identified and the considered Internet-based master-worker computation problem is formulated as a *Bayesian game* [29] (Section 2). We assume a probability distribution of workers among the worker types. The master and the workers do not know the type of other workers, only the probability distribution. The rational workers play a game looking for a Nash Equilibrium, choosing to be honest, cheat or abstain while the malicious and altruistic workers have a predefined strategy to cheat or be honest, respectively. The master does not participate in the game, it designs the game to be played. The network unreliability is modeled by a parametric probability.

- We develop and analyze two algorithms (a time-based algorithm and a reply-based one) that provide the necessary incentives for the rational workers to truthfully compute and return the task result, despite the malicious workers' actions and the network unreliability (Section 3). The algorithms are parametrized in terms of a probability of auditing $p_{\mathcal{A}}$ (defined in Section 2) and d, a parametric probability modeling networks unreliability. Each of the algorithms implements an instance of the Bayesian game. Under a general type probability distribution, we analyze the master's utility and probability of success (probability of obtaining the correct task result) and identify the conditions under which the game has Nash Equilibria.

  Under specific type probability distributions, a protocol in which the master chooses the values of $p_{\mathcal{A}}$ to guarantee a parametrized bound on the probability of success is also designed (Section 3). Once this is achieved, the master also attempts to maximize its utility. This protocol together with each of the above-mentioned algorithms comprise a mechanism. Note that the mechanisms designed (and their analyses) are general in that reward models can either be fixed exogenously or be chosen by the master. It is also shown that our mechanisms are the only feasible approaches for the master to achieve a given bound on the probability of success.

- Under the constrain of the bounded probability of success, it is shown how to maximize the master utility in two real-world scenarios (Section 4). The first scenario abstracts a system of volunteering computing like SETI [36]. The second scenario abstracts a contractor-based application where a company buys computational power from Internet users and sells it to computation-hungry consumers. One such application is Amazon's Mechanical Turk [4] where the master and the workers can be in fact humans that contribute time for solving problems in exchange of economic rewards.

- Finally, to provide a better insight on the usability of our mechanisms, and to illustrate the trade-offs between reliability and cost, we have characterized the utility of the master for the above-mentioned scenarios via plots by choosing system parameters as derived by empirical evaluations of master-worker Internet-based systems in [15] and [18].

1.3 Related work

Prior examples of game theory in distributed computing include work on Internet routing [24,37,42,49], resource/facility location and sharing [23, 26], containment of viruses spreading [44], secret sharing [2, 28], P2P services [3, 38, 39] and task computations [21, 58]. For more discussion on the connection between game theory and distributed computing we refer the reader to the surveys by Halpern [27] and by Abraham, Alvisi and Halpern [1], and the book by Nisan et al [47].

Eliaz [17] seems to be the first to formally study the co-existence of Byzantine (malicious) and rational players. He introduces the notion of *k-fault-tolerant Nash Equilibrium* as a state in which no player benefits from unilaterally deviating despite up to k players acting maliciously. He demonstrates this concept by designing simple mechanisms that implement the constrained Walrasian function and a choice rule for the efficient allocation of an indivisible good (e.g., in auctions). Abraham et al [2] extend Eliaz's concept to accommodate colluding rational players. In particular they design a secret sharing protocol and prove that it is (k, t)-robust, that is, it is correct despite up to k colluding rational players and t Byzantine ones.

Aiyer et al. [3] introduce the BAR model to reason about systems with Byzantine (malicious), Altruistic, and Rational participants. They also introduce the notion of a protocol being BAR-tolerant, that is, the protocol is resilient to both Byzantine faults and rational manipulation. (With this respect, one might say that our algorithmic mechanisms designed in this work is BAR-tolerant.) As an application, they designed a cooperative backup service for P2P systems, based on a BAR-tolerant replicated state machine. Li et al [39] also considered the BAR model to design a P2P live streaming application based on a BAR-tolerant gossip protocol. Both works employ incentive-based game theoretic techniques (to remove the selfish behavior), but the emphasis is on building a reasonably practical system (hence, formal analysis is traded for practicality). Recently, Li et al [38] developed a P2P streaming application, called FlightPath, that provides a highly reliable data stream to a dynamic set of peers. FlightPath, as opposed to the abovementioned BAR-based works, is based on mechanisms for *approximate equilibria* [10], rather than strict equilibria. In particular, ϵ-Nash equilibria are considered, in which rational players deviate if and only if they expect to benefit by more than a factor of ϵ. As the authors claim, the less restrictive nature of these equilibria enables the design of incentives to limit selfish behavior rigorously, while it provides sufficient flexibility to build practical systems.

Gairing [24] introduced and studied *malicious Bayesian congestion games*. These games extend congestion games [50] by allowing players to act in a malicious way. In particular, each player can either be rational or, with a certain probability, be malicious (with the sole goal of disturbing the other players). As in our work, players are not aware of each other's type, and this uncertainty is described by a probability distribution. Among other results, Gairing shows that, unlike congestion games, these games do not in general possess a Nash Equilibrium in pure strategies. Also he studies the impact of malicious types on the social cost (the overall performance of the system) by measuring the so-called

Price of Malice. This measure was first introduced by Moscibroda et al [44] to measure the influence of malicious behavior for a virus inoculation game involving both rational (selfish) and malicious nodes.

Distributed computation in presence of selfishness was studied within the scope of combinatorial agencies in Economics [6–8, 14]. The basic model considered is a combinatorial variant of the classical principal-agent problem [41]: A master (principal) must motivate a collection of workers (agents) to exert costly effort on the masters behalf, but the workers actions are hidden from the master. Instead of focusing on each worker's actions, the focus is on complex combinations of the efforts of the workers that influence the outcome. In [6], where the problem was first introduced, the goal was to study how the utility of the master is affected if the equilibria space is limited to pure strategies. To that extent, the computation of a few Boolean functions is evaluated. In [8] mixed strategies were considered: if the parameters of the problem yield multiple mixed equilibrium points, it is assumed that workers accept one suggested by the master. This is contrasted with our work as we require the master to enforce a single equilibrium point (referred as *strong implementation* in [6]). The work in [14] investigates the effect of auditing by allowing the master to audit some workers (by random sampling) and verify their work. In our work, the master decides probabilistically whether to verify all workers or none.

In general, the spirit of the framework considered in combinatorial agency is similar to the one we consider in the present work in the sense that there is a master wishing a specific outcome and it must provide necessary incentives to rational workers so to reach that outcome (exerting effort can be considered as the worker performing the task, and not, as the worker not performing the task and reporting a bogus result). However, there are several differences. First of all, we consider the co-existence of selfish, malicious and altruistic workers (we are not aware of any work in combinatorial agency that considers all these three types). Furthermore, we consider network unreliability (again, we are not aware of any work in Combinatorial agency with such assumption). Even if we consider a special case of our framework where we have a type distribution with only rational/selfish workers and communication is reliable, there are still many differences. One difference is that in our framework, the worker's actions cannot really be viewed as *hidden*. The master receives a response by each worker and it is aware that either the worker has truthfully performed the task or not. The outcome is affected by each workers action in the case that no verification is performed (in a similar fashion as the majority boolean technology in Combinatorial agency) but via verification the master can determine the exact strategy used by each worker and apply a specific reward/punishment scheme. In the framework considered in combinatorial agency, the master witnesses the outcome of the computation, but it has no knowledge of the possible actions that the worker might take. For this purpose, the master needs to devise contracts for each worker based on the observed outcome of the computation and not on each workers possible action (as in our framework). Another important difference includes the fact that our scheme considers worker punishment, as opposed to the schemes in combinatorial agency where workers cannot be fined (limited liability constraint); this is possible in our framework as worker's actions are contractible (either it performs a task or not).

2 Model and Definitions

2.1 Master-workers Framework and Worker Types

We consider a distributed system consisting of a master processor that assigns, over the Internet, a computational task to a set of n workers to compute and return the task result. The master, based on the received replies, must decide on the value it believes is the correct outcome of the task. The tasks considered in this work are assumed to have a unique solution; although such limitation reduces the scope of application of the presented mechanisms [54], there are plenty of computations where the correct solution is unique: e.g., any mathematical function.

Each of the n workers has one of the following types, *rational*, *malicious*, or *altruistic*. The exact number of workers of each type is unknown. However, it is known that each worker is independently of one of the three types with probabilities p_ρ, p_μ, p_α, respectively, where $p_\rho + p_\mu + p_\alpha = 1$. Malicious and altruistic workers always cheat and are honest, respectively, independently of how such a behavior impacts their utilities. In the context of this paper, being honest means truthfully compute and return the correct task result, and cheating means returning some incorrect value. On the other hand, rational workers are assumed to be selfish in a game-theoretic sense, that is, their aim is to maximize their benefit (utility) under the assumption that other workers do the same. So, a rational worker decides to be honest, cheat or not reply to the master (workers can abstain and choose not to reply) depending on which strategy maximizes its utility. As a result, each rational worker cheats with probability $p_\mathcal{C}$, it is honest with probability $p_\mathcal{H}$, and does not reply with probability $p_\mathcal{N}$, such that $p_\mathcal{C} + p_\mathcal{H} + p_\mathcal{N} = 1$. It is understood that if a worker decides not to reply, then it does not perform the task.

In order to model the individuality of the non-monetary part of each rational worker benefit/penalty, the distribution over types could be generalized to different types of rational workers instead of one. More precisely, define a probability distribution over each possible combination of payoffs in $\mathbb{R}^4$, restricting signs appropriately, so that each rational worker draws independently its strategic normal form from this distribution. However, the analysis presented here would be the same but using expected payoffs, the expectation taken over such distribution. Thus, for the sake of clarity and without loss of generality, we assume that the strategic normal form is unique for all players, i.e., all rational workers are of the same type.

2.2 Network Unreliability

The communication network is considered to be unreliable, and workers could be unavailable, which are very realistic assumptions for Internet-based master-worker computations, as suggested, for example, by the work of Heien at al. [30]. We model this shortcoming by assuming that the communication with each worker fails stochastically and independently of other workers.

Furthermore, we assume two settings, one where the probability of communication failure depends on time (the more the master waits for replies the larger the probability of obtaining more replies), and a second one where the probability of communication failure is fixed (hence, the more workers the master hires the larger the number of replies). As we will see in Section 3, the first setting leads to a *time-based* mechanism and the second one to a *reply-based* mechanism.

In our analysis, we let d_1 be the probability of any worker being available and receiving the task assignment message by the master, d_2 be the probability of the master receiving the worker's response (has the worker chosen to reply), and $d = d_1 \cdot d_2$ be the probability of a round trip, that is, the probability that the master receives the reply from a given worker. Hence, d_2 is the probability value that the master achieves by waiting T time (for the time-based mechanism) or hiring n workers (for the reply-based mechanism). We also assume that there is some chance of a message being delivered to its destination, i.e. $d > 0$, a realistic assumption for today's Internet's infrastructure.

2.3 Master's Objectives, Auditing, Payoffs and Reward Models

The objective of the master is twofold. First, the master has to guarantee that the decided value is correct with probability at least $1 - \varepsilon$, for a known constant $0 \leq \varepsilon < 1$. Then, having achieved this, the master wants to maximize its own benefit (utility). As, for example, in [51], [20] and [21], while it is assumed that workers make their decision individually and with no coordination, it is assumed that all the (malicious and rational) workers that cheat return the same incorrect value. This yields a worst case scenario (and hence analysis) for the master with respect to its probability of obtaining the correct result; it subsumes models where cheaters do not necessarily return the same answer. (In some sense, this can be seen as a cost-free, weak form of collusion.)

To achieve its objectives, the master employs, if necessary, *auditing* and *reward/penalizing* schemes. The master might decide to audit the response of the workers (at a cost). In this work, auditing means that the master computes the task by itself, and checks which workers have been truthful or not. We denote by $p_{\mathcal{A}}$ the probability of the master auditing the responses of the workers.

Furthermore, the master can reward and punish workers, which can be used (possibly combined with auditing) to encourage rational workers to be honest (altruistic workers need no encouragement, and malicious workers do not care about their utility). When the master audits, it can accurately reward and punish workers. When the master does not audit, it decides on the majority of the received replies, and may apply different reward/penalizing schemes. In this work we consider three reward models shown in Table 1. Each reward model is essentially different from the others and can be used depending on the specifics of the application considered.

$\mathcal{R}_{\mathrm{m}}$	the master rewards the majority only
$\mathcal{R}_{\mathrm{a}}$	the master rewards all workers
$\mathcal{R}_{\emptyset}$	the master does not reward any worker

Table 1: Reward models

Auditing or not, the master neither rewards nor punishes a worker from whom it did not receive its response. Due to the unreliability of the network, when the master does not receive a reply from a worker it can not distinguish whether the worker decided to abstain, or there was a communication failure in the round trip (it could be the case that the worker did not even receive the task assignment message). Hence, it would be unfair to punish a worker for not getting its response; imagine the case where the worker received the request, performed the task and replied to the master, but this last message got lost! On the other hand, if it is indeed the case that a worker received the task assignment message but decided to abstain, then it gets no reward. If the reward is much bigger than the worker's cost for computing the task, this alone can be a counter incentive to such a strategy.

The payoff parameters considered in this work are detailed in Table 2. All these parameters are non-negative. Note that the first letter of the parameter's name identifies whose parameter it is. M stands for master and W for worker. Then, the second letter gives the type of parameter. P stands for punishment, C for cost, and B for benefit.

$WP_{\mathcal{C}}$	worker's punishment for being caught cheating
$WC_{\mathcal{T}}$	worker's cost for computing the task
$WB_{\mathcal{Y}}$	worker's benefit from master's acceptance
$MP_{\mathcal{W}}$	master's punishment for accepting a wrong answer
$MC_{\mathcal{Y}}$	master's cost for accepting the worker's answer
$MC_{\mathcal{A}}$	master's cost for auditing worker's answers
$MC_{\mathcal{S}}$	master's cost for not getting any reply
$MB_{\mathcal{R}}$	master's benefit from accepting the right answer

Table 2: Payoffs

Observe that there are different parameters for the reward $WB_{\mathcal{Y}}$ to a worker and the cost $MC_{\mathcal{Y}}$ of this reward to the master. This models the fact that the cost to the master might be different from the benefit for a worker. In fact, in some applications they may be completely unrelated, as for example in the SETI-like scenario presented in Section 4.1. Although workers are not penalized for not replying, our model allows the possibility for the master to be penalized for not getting any replies (parameter $MC_{\mathcal{S}}$). This provides an incentive for the master to choose (when it can) more workers to assign the task (especially if d is small) or to increase their incentives for replying; if convenient, $MC_{\mathcal{S}}$ could be set to zero. Among the parameters involved, we assume that the master has the freedom of choosing $WB_{\mathcal{Y}}$ and $WP_{\mathcal{C}}$; by tuning these parameters and choosing n, the master can achieve the desired trade-offs between correctness and cost. All other parameters can either be fixed because they are system parameters or may also be chosen by the master.

2.4 Game Theory Concepts and Problem Formulation

We study the problem under the assumption that the rational workers, or *players*, will play a game looking for an equilibrium (recall that malicious and altruistic workers have a predefined strategy to cheat or be honest, respectively). The master does not play the game, it only defines the protocol and the parameters to be followed (i.e., it designs the game or mechanism). The master and the workers do not know the type of other workers, only the probability distribution. Hence, the game played is a so-called game with imperfect information or *Bayesian game* [29]. The action space is the set of pure strategies $\{\mathcal{C}, \mathcal{H}, \mathcal{N}\}$, and the belief of a player is the probability distribution over types.

More formally, the Internet-based Master-Worker computation considered in this work is formulated as the following Bayesian game

$$\mathcal{G}(W, \varepsilon, \mathcal{D}, A, p_{\mathcal{A}}, d_1, d_2, \mathcal{R}, pfs),$$

where W is the set of n workers, $1-\varepsilon \in [0,1]$ is the desired success probability of the master obtaining the correct task result, $\mathcal{D}$ is the type probability distribution $(p_\rho, p_\mu, p_\alpha)$, $A = \{\mathcal{C}, \mathcal{H}, \mathcal{N}\}$ is the workers' actions space, $p_{\mathcal{A}}$ is the probability of the master auditing the workers' responses, d_1 and d_2 are the probabilities characterizing the unreliability of the network ($d = d_1 \cdot d_2$), $\mathcal{R}$ is one of the reward models given in Table 1, and pfs are the payoffs as described in Table 2. Each player knows in advance the distribution over types $\mathcal{D}$, the total number of workers (n), the probability characterizing the networks unreliability (d_1, d_2) and its normal strategic form, which is assumed to be unique.

The core of the mechanisms we develop is the computation of $p_{\mathcal{A}}$. Based on the type distribution, the master must choose a value of $p_{\mathcal{A}}$ that would yield a *Nash Equilibrium* that best serves its purposes. Recall from [48], that for any finite game, a mixed strategy profile σ is a *mixed-strategy Nash equilibrium* (MSNE) if, and only if, for each player i,

$$\begin{aligned}
&U_i(s_i, \sigma_{-i}) = U_i(s_i', \sigma_{-i}), \forall s_i, s_i' \in supp(\sigma_i),\\
&U_i(s_i, \sigma_{-i}) \geq U_i(s_i', \sigma_{-i}),\\
&\qquad\qquad\qquad \forall s_i, s_i' : s_i \in supp(\sigma_i), s_i' \notin supp(\sigma_i),
\end{aligned}$$

where s_i is the strategy used by player i in the strategy profile s, σ_i is the probability distribution over pure strategies used by player i in σ, σ_{-i} is the probability distribution over pure strategies used by each player but i in σ, $U_i(s_i, \sigma_{-i})$ is the expected utility of player i when using strategy s_i with mixed strategy profile σ, and $supp(\sigma_i)$ is the set of strategies in σ with positive probability.

In words, given a MSNE with mixed-strategy profile σ, for each player i, the expected utility, assuming that all other players do not change their choice, is the same for each pure strategy that the player can choose with positive probability in σ, and it is not less than the expected utility of any pure strategy with probability zero of being chosen in σ. We denote by $\Delta U_{S_1 S_2}$ the difference on the expected utilities of a rational worker when choosing strategy S_1 over strategy S_2.

Then, for the purposes of the game we consider, in order to find conditions for equilibria, we want to study for each player i

$$\begin{cases} \Delta U_{\mathcal{HC}} = \boldsymbol{\pi}_{\mathcal{H}} \cdot \boldsymbol{w}_{\mathcal{H}} - \boldsymbol{\pi}_{\mathcal{C}} \cdot \boldsymbol{w}_{\mathcal{C}} \\ \Delta U_{\mathcal{HN}} = \boldsymbol{\pi}_{\mathcal{H}} \cdot \boldsymbol{w}_{\mathcal{H}} - \boldsymbol{\pi}_{\mathcal{N}} \cdot \boldsymbol{w}_{\mathcal{N}} \end{cases} \tag{1}$$

The expression $\cdot\, \boldsymbol{\pi}_\bullet \cdot \boldsymbol{w}_\bullet$ denotes the utility of the worker when choosing strategy $\bullet$; we present the components of the expression in detail in Section 3. If we show conditions such that $\Delta U_{\mathcal{HC}} = 0$ and

$W = \{1, 2, \ldots, n\}$	set of n workers
M	master processor
d_1	probability of a worker being available and receiving the task assignment message by the master
d_2	probability of the master receiving the worker's response (has the worker chosen to reply)
d	$d = d_1 \cdot d_2$, probability that the master receives a reply from a given worker
p_ρ	probability of a worker to be of rational type
p_μ	probability of a worker to be of malicious type
p_a	probability of a worker to be of altruistic type
$p_\mathcal{A}$	probability that the master audits (computes task and checks worker answers)
P_{succ}	probability that the master obtains correct answer
ε	known constant $\varepsilon \in [0, 1]$, $1 - \varepsilon$ desired bound on the probability of success
$\{\mathcal{C}, \mathcal{H}, \mathcal{N}\}$	action space of a worker
$p_\mathcal{C}$	probability of a worker to cheat
$p_\mathcal{H}$	probability of a worker to be honest
$p_\mathcal{N}$	probability of a worker not replying
s	strategy profile (a mapping from players to pure strategies)
s_i	strategy used by player i in the strategy profile s
s_{-i}	strategy used by each player but i in the strategy profile s
σ	mixed strategy profile (mapping from players to prob. distrib. over pure strat.)
σ_i	probability distribution over pure strategies used by player i in σ
σ_{-i}	probability distribution over pure strategies used by each player but i in σ
$U_i(s_i, \sigma_{-i})$	expected utility of player i with mixed strategy profile σ
$supp(\sigma_i)$	set of strategies of player i with probability > 0 in σ
$\Delta U_{S_1 S_2}$	difference on the expected utilities of a rational worker when choosing strategy S_1 over strategy S_2
$\mathbf{P}_q^{(n)}(a, b)$	$\sum_{i=a}^{b} \binom{n}{i} q^i (1-q)^{n-i}$

Table 3: Summary of Symbols

$\Delta U_{\mathcal{HN}} = 0$, then we have a MSNE $0 \neq p_\mathcal{C} \neq 1$. On the other hand, if we show conditions that make $\Delta U_{\mathcal{HC}} \geq 0$ and $\Delta U_{\mathcal{HN}} \geq 0$ for each player i, we know that there is a pure strategies NE where all players choose to be honest, i.e. $p_\mathcal{H} = 1$. (There is no NE where some players choose a pure strategy and others do not because the game is symmetric for all rational players. If a distribution over many types of rational players is defined, then we would have to consider such a NE.)

The following notation will be used throughout.

$$\mathbf{P}_q^{(n)}(a, b) \triangleq \sum_{i=a}^{b} \binom{n}{i} q^i (1-q)^{n-i}$$

The notation used throughout the paper is summarized in Table 3.

3 Algorithmic Mechanisms

In this section we present the mechanisms we design and show their analysis. In particular, we show two different algorithms that the master runs in order to obtain the result of the task. Each of these algorithms is essentially an instance of the game we defined in the previous section. Before running one of the algorithms, the master must chose an appropriate value of $p_\mathcal{A}$; it does so by running a protocol we also present in this section. This protocol, together with each of the algorithms the master runs to obtain the tasks, comprises a mechanism.

send(*task*, $p_{\mathcal{A}}$, *certificate*) **to** n *workers*
wait *time T for replies*
upon *expire of time T* **do**
audit the answers with probability $p_{\mathcal{A}}$
if *the answers were not audited* **then**
    accept the majority
end if
apply the reward model

Figure 1: Master Algorithm for the Time-based Mechanism

send(*task*, $p_{\mathcal{A}}$, *certificate*) **to** n *workers*
if *at least* k *replies are received* **then**
    audit the answers with probability $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept the majority
    end if
    apply the reward model
end if

Figure 2: Master Algorithm for the Reply-based Mechanism

3.1 Algorithms

As discussed in Section 2.2, we consider two different settings for modeling network unreliability, which yield two different algorithms.

Figure 1 presents the *time-based* algorithm. Based on how the probability of communication failure depends on time, the master fixes a time T, it sends the specification of the task to be computed to n workers, and waits for replies. Once time T is reached, the master gathers all received replies, and chooses to audit the answers with probability $p_{\mathcal{A}}$. If the answers were not audited, it accepts the result of the majority (ties are broken at random). Then, it applies the corresponding reward model.

Figure 2 presents the *reply-based* algorithm. Here the master, by appropriately choosing n, fixes k, an estimate of the minimum number of replies that wants to receive with high probability. (We discuss in the next subsection how k is computed and what is the probability of not receiving at least that many answers). The master sends the task specification to the n workers and gets replies. If at least k replies are received, then the master chooses to audit the answers with probability $p_{\mathcal{A}}$ and proceeds as the other protocol. In case that less than k replies are received, then the master does nothing and it incurs penalty $MC_{\mathcal{S}}$.

Notice that both algorithms are one-shot, in the sense that they terminate after one round of communication between the master and the workers. This enables fast termination and avoids using complex cheater detection and worker reputation mechanisms. The benefit of one-round protocols is also partially supported by the work of Kondo et al. [33] that have demonstrated experimentally that tasks may take much more than one day of CPU time to complete.

Each of the above algorithms basically implements an instance of the game we presented in Section 2.4. The master designs the game and the rational workers play looking for a Nash Equilibrium (NE) in an effort to maximize their benefit. Therefore, based on the type distribution, the master must choose the value of $p_{\mathcal{A}}$ that would yield a *unique* NE that best serves its purposes. The reason for uniqueness is to force all workers to the same strategy; this is similar to *strong implementation* in Mechanism Design, cf., [6, 46]. (Multiple equilibria could be considered that could perhaps favor the utility of the master. However, in this work, correctness is the priority which, as shown later, our mechanisms guarantee.) For computational reasons, along with the task specification and the chosen value of $p_{\mathcal{A}}$, and the task to be computed, the master also sends a *certificate* to the workers. The certificate includes the

if *Pr*[*majority honest* | *all rationals honest*] $< 1-\varepsilon$ **then** /* P_{succ} is small, even if $p_{\mathcal{H}} = 1$ */
  $p_{\mathcal{C}} \leftarrow 1; p_{\mathcal{N}} \leftarrow 0; p_{\mathcal{A}} \leftarrow 1-\varepsilon / \sum_{i=k}^{n} r_i c_i;$ /* cf. Lemma 2 */
elseif *Pr*[*majority honest* | *all rationals cheat*] $\geq 1-\varepsilon$ **then** /* P_{succ} is big, even if $p_{\mathcal{C}} = 1$ */
  $p_{\mathcal{C}} \leftarrow 1; p_{\mathcal{N}} \leftarrow 0; p_{\mathcal{A}} \leftarrow 0;$ /* cf. Lemma 3 */
elseif *Pr*[*majority honest* | *all rationals honest*] $\geq 1-\varepsilon$ **and**
    $\Delta U_{\mathcal{HC}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **and** $\Delta U_{\mathcal{HN}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **then** /* $p_{\mathcal{H}} = 1$, even if $p_{\mathcal{A}} = 0$ */
  $p_{\mathcal{C}} \leftarrow 0; p_{\mathcal{N}} \leftarrow 0; p_{\mathcal{A}} \leftarrow 0;$ /* cf. Lemma 3 */
else /* $p_{\mathcal{C}} = 0$ and $p_{\mathcal{N}} = 0$ enforced */
  $p_{\mathcal{C}} \leftarrow 0; p_{\mathcal{N}} \leftarrow 0;$ *set* $p_{\mathcal{A}}$ *as in* Lemma 4; /* cf. Lemma 4 */
if $U_M(p_{\mathcal{A}}, p_{\mathcal{N}}, p_{\mathcal{C}}) < U_M\left(p_{\mathcal{A}} = (1-\varepsilon) / \sum_{i=k}^{n} r_i, p_{\mathcal{N}} = 1, p_{\mathcal{C}} = 0\right)$ **then**
  $p_{\mathcal{N}} \leftarrow 1; p_{\mathcal{A}} \leftarrow (1-\varepsilon) / \sum_{i=k}^{n} r_i;$ /* cf. Lemma 1 */

Figure 3: Master protocol to choose $p_{\mathcal{A}}$. The expressions of k, r_i, and c_i are defined in Section 3.2

strategy that if the rational workers play will lead them to the unique NE, together with the appropriate data (system parameters/payoff values and reward model) to demonstrate this fact. More details for the use of the certificate are given in Section 3.4.

Recall that the main objective of the master is to achieve probability of accepting the correct task result of at least $1-\varepsilon$. Once this is achieved, then it seeks to maximize its utility as well. Based on the type distribution, it could be the case that the master may achieve this without relying on actions of the rational workers (e.g., the vast majority of workers are altruistic). Such cases fall into what we call the *free rationals scenario*. The cases in which the master needs to enforce the behavior of rational workers ($p_{\mathcal{H}}$) fall into what we call the *guided rationals scenario*. In this scenario, the master must choose $p_{\mathcal{A}}$ such that the benefit of the rational workers is maximized when $p_{\mathcal{C}} = p_{\mathcal{N}} = 0$; in other words, rational workers choose to be honest ($p_{\mathcal{H}} = 1$) and hence they compute and truthfully return the correct task result. The protocol ran by the master for choosing $p_{\mathcal{A}}$ is presented in Figure 3. Together with each of the algorithms in Figures 1 and 2 comprise our ***mechanisms***. The analysis of the mechanisms and the lemmas referenced in Figure 3 are given in the next subsection.

Note that both designed mechanisms are useful and can be used depending on the setting. For example:
(a) As discussed in Section 2.2, the probability of the communication failure could depend on time, or be fixed. The master could have knowledge (e.g., based on statistics) of only one of the two settings. In such a case, it has no choice other than using the mechanism designed for that setting.
(b) It is not difficult to see that the time-based mechanism is more likely to use auditing than the other one, on the other hand, the reply-based mechanism runs the risk of not receiving enough replies. Hence, the time-based mechanism would be more preferable in case the cost of auditing is low, and the reply-based mechanism in case the cost of auditing is high and the value of parameter $MC_{\mathcal{S}}$ is small.

Also observe that in the case of reliable communication ($d = 1$), the two mechanisms converge, that is, they become the same. Since the master enforces rational workers to be honest (and hence reply), altruistic and malicious always reply, and communication is reliable, the master can wait until it receives messages from all workers and then proceed. Furthermore, as it can be observed in the next section, the analysis of the two mechanisms in the case of reliable communication is identical.

3.2 Equilibria Conditions and Analysis

We begin the analysis of our mechanisms by elucidating the following probabilities, expected utilities, and equilibria conditions. For succinctness, the analysis of both mechanisms is presented for a minimum number of replies k, where $k = 1$ for the time-based mechanism and $k \geq 1$ for the reply-based mechanism. For the latter, for a given worker type distribution, the choice of n workers, and d, even if all rational workers choose not to reply, the master will receive at least $\boldsymbol{E} = nd(p_\alpha + p_\mu)$ replies in expectation. Thus, using Chernoff bounds, it can be shown that the master will receive at least $k = \boldsymbol{E} - \sqrt{2\boldsymbol{E}\ln(1/\zeta)}$ replies with probability at least $1-\zeta$, for $0 < \zeta < 1$ and big enough n (e.g., $\zeta = 1/n$).

3.2.1 Probabilities and expected utilities.

Given the description of the mechanisms and the system parameters, it is not difficult to compute the following:

Pr(worker cheats|worker replies): $q = \frac{p_\mu + p_\rho p_\mathcal{C}}{1-p_\rho p_\mathcal{N}}$

Pr(worker does not cheat|worker replies): $\overline{q} = \frac{p_\alpha + p_\rho p_\mathcal{H}}{1-p_\rho p_\mathcal{N}} = 1-q$

Pr(reply received from worker): $r = d(1-p_\rho p_\mathcal{N})$

Pr(reply not received from worker): $\overline{r} = 1-r$

Then, $r(q+\overline{q}) + \overline{r} = 1$.

Pr(i out of n replies received): $r_i = \binom{n}{i} r^i \overline{r}^{n-i}$

Pr(majority honest | i replies received):

$$h_i = \sum_{j=0}^{\lfloor i/2 \rfloor -1} \binom{i}{j} q^j \overline{q}^{i-j} + (1 + \lceil i/2 \rceil - \lfloor i/2 \rfloor)\frac{1}{2}\binom{i}{\lfloor i/2 \rfloor} q^{\lfloor i/2 \rfloor}\overline{q}^{\lceil i/2 \rceil}.$$

Pr(majority cheats | i replies received):

$$c_i = \sum_{j=\lceil i/2 \rceil +1}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + (1 + \lceil i/2 \rceil - \lfloor i/2 \rfloor)\frac{1}{2}\binom{i}{\lceil i/2 \rceil} q^{\lceil i/2 \rceil}\overline{q}^{\lfloor i/2 \rfloor}.$$

Pr(master obtains correct answer):

$$P_{succ} = \sum_{i=k}^{n} r_i \left(p_\mathcal{A} + (1-p_\mathcal{A})h_i\right) \tag{2}$$

E(utility of master):

$$U_M = -\sum_{i=0}^{k-1} r_i \cdot MC_\mathcal{S} + \sum_{i=k}^{n} r_i \big(p_\mathcal{A}\alpha_i + (1-p_\mathcal{A})\beta_i\big) \tag{3}$$

where,

$$\alpha_i = MB_\mathcal{R} - MC_\mathcal{A} - nd(p_\alpha + p_\rho p_\mathcal{H})MC_\mathcal{Y}$$
$$\beta_i = MB_\mathcal{R} h_i - MP_\mathcal{W} c_i - MC_\mathcal{Y}\gamma_i$$

and where, $\gamma_i = 0$ for $\mathcal{R}_\emptyset$, $\gamma_i = i$ for $\mathcal{R}_\mathrm{a}$, and for $\mathcal{R}_\mathrm{m}$ is,

$$\gamma_i = \sum_{j=\lceil i/2 \rceil +1}^{i} \binom{i}{j} j(\overline{q}^j q^{i-j} + q^j \overline{q}^{i-j})$$
$$+ (1 + \lceil i/2 \rceil - \lfloor i/2 \rfloor)\frac{1}{2}\binom{i}{\lceil i/2 \rceil}\lceil i/2 \rceil(\overline{q}^{\lceil i/2 \rceil} q^{\lfloor i/2 \rfloor} + q^{\lceil i/2 \rceil}\overline{q}^{\lfloor i/2 \rfloor}).$$

3.2.2 General Equilibria Conditions

Recall from Section 2.4 that Equation (1) states the conditions we want to study for each player i. In particular, as discussed there, we want $\Delta U_{\mathcal{HC}} \geq 0$ and $\Delta U_{\mathcal{HN}} \geq 0$.

The components of the vectors denoted by $\boldsymbol{w}_\bullet$ in (1) correspond to the different payoffs received by the given worker for each of the various events that may outcome from the game when the worker has chosen strategy $\bullet$, and the components of the vectors denoted by $\boldsymbol{\pi}_\bullet$ correspond to the probabilities that those events occur. Their detail values are given in Tables 4, 5, and 6; Table 7 lists the used notation. These conditions are defined so that a pure NE where $p_{\mathcal{H}} = 0$ is precluded.

3.2.3 Analysis Based on the Worker-type Distribution

Appropriate strategies to carry out the computation with the desired probability of success under the free rationals and guided rationals scenarios are considered in this section. It is important to stress again that, in order to obtain a mechanism that is useful for any of those scenarios we do not restrict ourselves to a particular instance of payoffs or reward models leaving those variables as parameters. Thus, we focus our study here on how to choose $p_{\mathcal{A}}$ to have the probability of success bounded by $1 - \varepsilon$ for each of the reward models assuming that the payoffs have already been chosen by the master or are fixed exogenously. For settings where payoffs and reward models are a choice of the master, its utility can be easily maximized choosing those parameters conveniently in Equation 3, as demonstrated in Section 4.

Although known, the worker-type distribution is assumed to be arbitrary. Likewise, the particular value of ε is arbitrary given that it is an input of the problem. Finally, although the priority is to obtain $P_{succ} \geq 1 - \varepsilon$, it is desirable to maximize the utility of the master under such restriction. Therefore, as it can be seen in Figure 3, the protocol the master runs for choosing $p_{\mathcal{A}}$ takes into account both the free rationals and guided rationals scenarios as discussed in Section 3.1.

We now proceed to analyze the different cases, first considering the free rationals scenario and then the guided rationals one.

Free Rationals

Here we study the various cases where the behavior of rational workers does not need to be enforced. As mentioned before, the main goal is to carry out the computation obtaining the correct output with probability at least $1 - \varepsilon$. Provided that this goal is achieved, it is desirable to maximize the utility of the master. Hence if, for a given instance of the problem, the expected utility of the master utilizing the mechanism presented is smaller than the utility of just setting $p_{\mathcal{A}}$ big enough to guarantee the desired probability of correctness, independently of the outcome of the game, the latter is used. We establish this observation in the following lemma.

Lemma 1. *In order to guarantee $P_{succ} \geq 1 - \varepsilon$, it is enough to set $p_{\mathcal{A}} = (1 - \varepsilon) / \sum_{i=k}^{n} r_i$, making $p_{\mathcal{N}} = 1$.*

Proof. Conditioning Equation 2 to be $\geq 1 - \varepsilon$, it is enough to make $p_{\mathcal{A}} \geq \dfrac{1 - \varepsilon}{\sum_{i=k}^{n} r_i}$. Given that $\sum_{i=k}^{n} r_i$ is the probability that k or more replies are received, it is minimized when $p_{\mathcal{N}} = 1$. Therefore, the claim follows. □

We consider now pessimistic worker-type distributions, i.e., distributions where p_μ is so large that, even if all rationals choose to be honest, the probability of obtaining the correct answer is too small. Hence, the master has to audit with a probability big enough, perhaps bigger than the minimum needed to ensure that all rationals are honest. Nevertheless, for such $p_{\mathcal{A}}$, rational workers still might use some NE where $p_{\mathcal{H}} < 1$. Thus, the worst case for P_{succ} has to be assumed. Formally,

		$\mathcal{R}_{\mathrm{m}}$	$\mathcal{R}_{\mathrm{a}}$	$\mathcal{R}_{\emptyset}$
	$w_{\mathcal{C}}^{\mathcal{AR}}$	$-WP_{\mathcal{C}}$	$-WP_{\mathcal{C}}$	$-WP_{\mathcal{C}}$
$\boldsymbol{w}_{\mathcal{C}}$	$w_{\mathcal{C}}^{\mathcal{CR}}$	$WB_{\mathcal{Y}}$	$WB_{\mathcal{Y}}$	0
	$w_{\mathcal{C}}^{\mathcal{HR}}$	0	$WB_{\mathcal{Y}}$	0
	$w_{\mathcal{C}}^{\mathcal{X}\overline{\mathcal{R}}}$	0	0	0
	$w_{\mathcal{H}}^{\mathcal{AR}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$
$\boldsymbol{w}_{\mathcal{H}}$	$-WP_{\mathcal{C}} - WC_{\mathcal{T}}$	$-WC_{\mathcal{T}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$	$-WC_{\mathcal{T}}$
	$w_{\mathcal{H}}^{\mathcal{HR}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$	$WB_{\mathcal{Y}} - WC_{\mathcal{T}}$	$-WC_{\mathcal{T}}$
	$w_{\mathcal{H}}^{\mathcal{X}\overline{\mathcal{R}}}$	$-WC_{\mathcal{T}}$	$-WC_{\mathcal{T}}$	$-WC_{\mathcal{T}}$
$\boldsymbol{w}_{\mathcal{N}}$	$w_{\mathcal{N}}^{\mathcal{XX}}$	0	0	0

Table 4: Payoff vectors. Refer to Table 7 for notation.

	$\pi_{\mathcal{C}}^{\mathcal{AR}}$	$dp_{\mathcal{A}}$
	$\pi_{\mathcal{C}}^{\mathcal{CR}}$	$d(1-p_{\mathcal{A}})\sum_{i=0}^{n-1}\binom{n-1}{i}r^i\overline{r}^{n-1-i}$ $\left(\sum_{j=\lceil i/2\rceil}^{i}\binom{i}{j}q^j\overline{q}^{i-j}+\left(\lceil i/2\rceil-\lfloor i/2\rfloor\right)\frac{1}{2}\binom{i}{\lfloor i/2\rfloor}q^{\lfloor i/2\rfloor}\overline{q}^{\lceil i/2\rceil}\right)$
$\boldsymbol{\pi}_{\mathcal{C}}$	$\pi_{\mathcal{C}}^{\mathcal{HR}}$	$d(1-p_{\mathcal{A}})\sum_{i=0}^{n-1}\binom{n-1}{i}r^i\overline{r}^{n-1-i}$ $\left(\sum_{j=0}^{\lfloor i/2\rfloor-1}\binom{i}{j}q^j\overline{q}^{i-j}+\left(\lceil i/2\rceil-\lfloor i/2\rfloor\right)\frac{1}{2}\binom{i}{\lfloor i/2\rfloor}q^{\lfloor i/2\rfloor}\overline{q}^{\lceil i/2\rceil}\right)$
	$\pi_{\mathcal{C}}^{\mathcal{X}\overline{\mathcal{R}}}$	$d_1(1-d_2)$
	$\pi_{\mathcal{H}}^{\mathcal{AR}}$	$dp_{\mathcal{A}}$
	$\pi_{\mathcal{H}}^{\mathcal{CR}}$	$d(1-p_{\mathcal{A}})\sum_{i=0}^{n-1}\binom{n-1}{i}r^i\overline{r}^{n-1-i}$ $\left(\sum_{j=\lceil i/2\rceil+1}^{i}\binom{i}{j}q^j\overline{q}^{i-j}+\left(\lceil i/2\rceil-\lfloor i/2\rfloor\right)\frac{1}{2}\binom{i}{\lfloor i/2\rfloor}q^{\lceil i/2\rceil}\overline{q}^{\lfloor i/2\rfloor}\right)$
$\boldsymbol{\pi}_{\mathcal{H}}$	$\pi_{\mathcal{H}}^{\mathcal{HR}}$	$d(1-p_{\mathcal{A}})\sum_{i=0}^{n-1}\binom{n-1}{i}r^i\overline{r}^{n-1-i}$ $\left(\sum_{j=0}^{\lfloor i/2\rfloor}\binom{i}{j}q^j\overline{q}^{i-j}+\left(\lceil i/2\rceil-\lfloor i/2\rfloor\right)\frac{1}{2}\binom{i}{\lceil i/2\rceil}q^{\lceil i/2\rceil}\overline{q}^{\lfloor i/2\rfloor}\right)$
	$\pi_{\mathcal{H}}^{\mathcal{X}\overline{\mathcal{R}}}$	$d_1(1-d_2)$
$\boldsymbol{\pi}_{\mathcal{N}}$	$\pi_{\mathcal{N}}^{\mathcal{XX}}$	d_1

Table 5: Probability vectors for the time-based mechanism. Refer to Table 7 for notation.

$\boldsymbol{\pi}_{\mathcal{C}}$	$\pi_{\mathcal{C}}^{\mathcal{A}\mathcal{R}}$	$dp_{\mathcal{A}} \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i}$
	$\pi_{\mathcal{C}}^{\mathcal{C}\mathcal{R}}$	$d(1-p_{\mathcal{A}}) \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i} \left(\sum_{j=\lceil i/2 \rceil}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right) \frac{1}{2} \binom{i}{\lfloor i/2 \rfloor} q^{\lfloor i/2 \rfloor} \overline{q}^{\lceil i/2 \rceil} \right)$
	$\pi_{\mathcal{C}}^{\mathcal{H}\mathcal{R}}$	$d(1-p_{\mathcal{A}}) \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i} \left(\sum_{j=0}^{\lfloor i/2 \rfloor - 1} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right) \frac{1}{2} \binom{i}{\lfloor i/2 \rfloor} q^{\lfloor i/2 \rfloor} \overline{q}^{\lceil i/2 \rceil} \right)$
	$\pi_{\mathcal{C}}^{\mathcal{X}\overline{\mathcal{R}}}$	$d_1(1-d_2) + d \sum_{i=0}^{k-2} \binom{n-1}{i} r^i \overline{r}^{n-1-i}$
$\boldsymbol{\pi}_{\mathcal{H}}$	$\pi_{\mathcal{H}}^{\mathcal{A}\mathcal{R}}$	$dp_{\mathcal{A}} \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i}$
	$\pi_{\mathcal{H}}^{\mathcal{C}\mathcal{R}}$	$d(1-p_{\mathcal{A}}) \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i} \left(\sum_{j=\lceil i/2 \rceil + 1}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right) \frac{1}{2} \binom{i}{\lfloor i/2 \rfloor} q^{\lceil i/2 \rceil} \overline{q}^{\lfloor i/2 \rfloor} \right)$
	$\pi_{\mathcal{H}}^{\mathcal{H}\mathcal{R}}$	$d(1-p_{\mathcal{A}}) \sum_{i=k-1}^{n-1} \binom{n-1}{i} r^i \overline{r}^{n-1-i} \left(\sum_{j=0}^{\lfloor i/2 \rfloor} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right) \frac{1}{2} \binom{i}{\lceil i/2 \rceil} q^{\lceil i/2 \rceil} \overline{q}^{\lfloor i/2 \rfloor} \right)$
	$\pi_{\mathcal{H}}^{\mathcal{X}\overline{\mathcal{R}}}$	$d_1(1-d_2) + d \sum_{i=0}^{k-2} \binom{n-1}{i} r^i \overline{r}^{n-1-i}$
$\boldsymbol{\pi}_{\mathcal{N}}$	$\pi_{\mathcal{N}}^{\mathcal{X}\mathcal{X}}$	d_1

Table 6: Probability vectors for the reply-based mechanism. Refer to Table 7 for notation.

$w_{\bullet}^{\bullet\bullet}$	payoff of event $\bullet \wedge \bullet \wedge \bullet$
$\pi_{\circ}^{\bullet\bullet}$	probability of event $\bullet \wedge \bullet$, conditioned on the event $\circ$
$\ell_{j}^{\bullet\bullet}$	the worker has choosen strategy $j \in \{\mathcal{C}, \mathcal{H}, \mathcal{N}\}$
$\ell_{\bullet}^{\mathcal{A}\bullet}$	the master audits
$\ell_{\bullet}^{\mathcal{C}\bullet}$	the master does not audit and the majority cheats
$\ell_{\bullet}^{\mathcal{H}\bullet}$	the master does not audit and the majority does not cheat
$\ell_{\bullet}^{\bullet\mathcal{R}}$	the communication is successful and the master receives enough replies
$\ell_{\bullet}^{\bullet\overline{\mathcal{R}}}$	the communication fails or the master does not receive enough replies
$\mathcal{X}$	true (equivalent to "any value")

Table 7: Notation for Tables 4, 5, and 6; $\ell \in \{w, \pi\}$.

Lemma 2. *In order to guarantee $P_{succ} \geq 1-\varepsilon$, it is enough to set $p_\mathcal{A} = 1 - \varepsilon / \sum_{i=k}^{n} r_i c_i$, making $p_\mathcal{C} = 1$ and $p_\mathcal{N} = 0$.*

Proof. Conditioning Equation 2 to be $\geq 1-\varepsilon$, $p_\mathcal{A} \geq 1 - \varepsilon \Big/ \sum\limits_{i=k}^{n} r_i c_i$. Given that $\sum_{i=k}^{n} r_i c_i$ is the probability that k or more replies are received and the majority of them cheat, it is maximized when $p_\mathcal{C} = 1$ (hence, $p_\mathcal{N} = 0$). Therefore, the claim follows. □

Now, we consider cases where no audit is needed to achieve the desired probability of correctness. I.e., we study conditions under the assumption that $p_\mathcal{A} = 0$. The first case occurs when the type-distribution is such that, even if all rational workers cheat, the probability of having a majority of correct answers is at least $1-\varepsilon$. A second case happens when the particular instance of the parameters of the game force a unique NE such that rationals are honest, even if they know that the result will not be audited. We establish those cases in the following lemma.

Lemma 3. *If any of the following holds:*

- *$\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon$ making $p_\mathcal{C} = 1$ and $p_\mathcal{N} = 0$; or*
- *$\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon$ making $p_\mathcal{C} = 0$ and $p_\mathcal{N} = 0$ and there is a unique NE for $p_\mathcal{H} = 1$ and $p_\mathcal{A} = 0$,*

then, in order to guarantee $P_{succ} \geq 1-\varepsilon$, it is enough to set $p_\mathcal{A} = 0$.

Proof. Conditioning Equation 2 to be $\geq 1-\varepsilon$ under the assumption that $p_\mathcal{A} = 0$, it is enough

$$\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon. \tag{4}$$

To find the condition for the case where even if all rationals cheat the probability of success is big enough, we replace $p_\mathcal{C} = 1$ and $p_\mathcal{N} = 0$ in Eq.(4). For the condition when the NE corresponds to some $p_\mathcal{C} < 1$, we observe the following. Replacing in $\Delta U_{\mathcal{HC}}$ and $\Delta U_{\mathcal{HN}}$ for each reward model the value $p_\mathcal{A} = 0$, it can be shown that $\Delta U_{\mathcal{HC}}(p_\mathcal{C}, p_\mathcal{A} = 0)$ is non-increasing in the interval $p_\mathcal{C} \in [0,1]$ for all three reward models, and $\Delta U_{\mathcal{HN}}(p_\mathcal{N}, p_\mathcal{A} = 0)$ is non-increasing in the interval $p_\mathcal{N} \in [0,1]$ for all three reward models as well. Thus, if $\Delta U_{\mathcal{HC}}(p_\mathcal{C} = 1, p_\mathcal{A} = 0) \geq 0$ and $\Delta U_{\mathcal{HN}=1}(p_\mathcal{N} = 1, p_\mathcal{A} = 0) \geq 0$, the rate of growth of $\Delta U_{\mathcal{HC}}$ and $\Delta U_{\mathcal{HN}}$ implies a single pure NE at $p_\mathcal{H} = 1$. Then, replacing $p_\mathcal{C} = 0$ and $p_\mathcal{N} = 0$ in Eq.(4) the claim follows. □

Guided Rationals

We now study worker-type distributions such that the master can take advantage of a specific NE to achieve the desired bound on the probability of success. Given that the scenario where all players cheat was considered in the free rationals scenario, here it is enough to study $\Delta U_{\mathcal{HC}}$ and $\Delta U_{\mathcal{HN}}$ for each reward model, conditioning $\Delta U_{\mathcal{HC}}(p_\mathcal{C} = 1) \geq 0$ and $\Delta U_{\mathcal{HN}}(p_\mathcal{N} = 1) \geq 0$ to obtain appropriate values for $p_\mathcal{A}$. As proved in the following lemma, the specific value $p_\mathcal{A}$ assigned depends on the reward model, and it is set so that a unique pure NE is forced at $p_\mathcal{H} = 1$ (rendering the rationals truthful), and the correctness probability is achieved.

Lemma 4. *If $\sum_{i=k}^{n} r_i h_i < 1-\varepsilon$ making $p_\mathcal{C} = 1$ and $p_\mathcal{N} = 0$, and $\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon$ making $p_\mathcal{C} = 0$ and $p_\mathcal{N} = 0$ then, in order to guarantee $P_{succ} \geq 1-\varepsilon$, it is enough to set $p_\mathcal{A}$ as follows.*

For $\mathcal{R}_\emptyset$,

$$p_\mathcal{A} = \frac{WC_\mathcal{T}}{d_2\, WB_\mathcal{Y} \sum_{i=k-1}^{n-1} r'_i} \tag{5}$$

For $\mathcal{R}_{\mathrm{a}}$*,*

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}}{d_2(WB_{\mathcal{Y}} + WP_{\mathcal{C}})\sum_{i=k-1}^{n-1} r'_i} \tag{6}$$

$$d_2\, WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r'_i \geq WC_{\mathcal{T}} \tag{7}$$

For $\mathcal{R}_{\mathrm{m}}$*,*

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}/d_2 - WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i(h'_i - c'_i)}{(WB_{\mathcal{Y}} + WP_{\mathcal{C}})\sum_{i=k-1}^{n-1} r'_i - WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i(h'_i - c'_i)} \tag{8}$$

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}/d_2 - WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i h'_i}{WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i - WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i h'_i} \tag{9}$$

Where

$$r'_i = \tbinom{n-1}{i} r^i \overline{r}^{n-1-i},$$
$$h'_i = \textstyle\sum_{j=0}^{\lfloor i/2 \rfloor} \tbinom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right)\tfrac{1}{2}\tbinom{i}{\lceil i/2 \rceil} q^{\lceil i/2 \rceil}\overline{q}^{\lfloor i/2 \rfloor},$$
$$c'_i = \textstyle\sum_{j=\lceil i/2 \rceil}^{i} \tbinom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right)\tfrac{1}{2}\tbinom{i}{\lfloor i/2 \rfloor} q^{\lfloor i/2 \rfloor}\overline{q}^{\lceil i/2 \rceil},$$

for $p_{\mathcal{C}} = 1$ *in conditions (6) and (8), and for* $p_{\mathcal{N}} = 1$ *in conditions (5), (7) and (9).*

Proof. We compute the general conditions for each reward model from Equations (1). (Refer to Tables 4, 5, and 6 for details.) Recall that, for succinctness, the analysis of both mechanisms is presented for a number of replies k, where $k = 1$ for the time-based mechanism and $k = nd(p_\alpha + p_\mu)\left(1 - \sqrt{\frac{2\ln(1/\zeta)}{nd(p_\alpha+p_\mu)}}\right)$ for the reply-based mechanism.

Conditions for reward model $\mathcal{R}_{\emptyset}$:

$$\Delta U_{\mathcal{HC}} = dp_{\mathcal{A}}(WB_{\mathcal{Y}} + WP_{\mathcal{C}})\sum_{i=k-1}^{n-1} r'_i - WC_{\mathcal{T}} d_1 \geq 0$$

$$\Delta U_{\mathcal{HN}} = dp_{\mathcal{A}}\, WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i - WC_{\mathcal{T}} d_1 \geq 0$$

Thus, it is enough to use the latter condition only.

Conditions for the reward model $\mathcal{R}_{\mathrm{a}}$:

$$\Delta U_{\mathcal{HC}} = dp_{\mathcal{A}}(WB_{\mathcal{Y}} + WP_{\mathcal{C}})\sum_{i=k-1}^{n-1} r'_i - WC_{\mathcal{T}} d_1 \geq 0$$

$$\Delta U_{\mathcal{HN}} = d\, WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i - WC_{\mathcal{T}} d_1 \geq 0$$

Conditions for the reward model $\mathcal{R}_{\mathrm{m}}$:

$$\Delta U_{\mathcal{HC}} = dp_{\mathcal{A}}(WB_{\mathcal{Y}} + WP_{\mathcal{C}})\sum_{i=k-1}^{n-1} r'_i - d_1\, WC_{\mathcal{T}} + d(1 - p_{\mathcal{A}})\, WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i(h'_i - c'_i) \geq 0 \tag{10}$$

$$\Delta U_{\mathcal{HN}} = dp_{\mathcal{A}}\, WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i - d_1\, WC_{\mathcal{T}} + d(1 - p_{\mathcal{A}})\, WB_{\mathcal{Y}}\sum_{i=k-1}^{n-1} r'_i h'_i \geq 0 \tag{11}$$

Notice that $\sum_{i=k-1}^{n-1} r'_i h'_i$ is the probability that at least $k-1$ other workers reply, and the majority of them is honest and $\sum_{i=k-1}^{n-1} r'_i c'_i$ is the probability that at least $k-1$ other workers reply, and the majority of them cheat. It can be seen that, when $p_{\mathcal{N}}$ is fixed, the equilibria condition 10 for this model is non-increasing on $p_{\mathcal{C}} \in [0, 1-p_{\mathcal{N}}]$ as follows. Only $\sum_{i=k-1}^{n-1} r'_i(h'_i - c'_i)$ depends on $p_{\mathcal{C}}$ in this condition. When $p_{\mathcal{C}}$ increases and $p_{\mathcal{N}}$ is fixed, the probability that the majority of repliers is honest decreases. On the other hand, the probability that the majority cheats increases with $p_{\mathcal{C}}$, but given that it is negated the slope is negative. Likewise, it can be seen that, when $p_{\mathcal{C}}$ is fixed, the equilibria condition 11 for this model is non-increasing on $p_{\mathcal{N}} \in [0, 1-p_{\mathcal{C}}]$ as follows. Only $\sum_{i=k-1}^{n-1} r'_i h'_i$ depends on $p_{\mathcal{N}}$ in this condition. When $p_{\mathcal{N}}$ increases and $p_{\mathcal{C}}$ is fixed, the probability that the majority of repliers is honest decreases. Therefore, replacing in the above conditions for $\Delta U_{\mathcal{HC}}(p_{\mathcal{C}} = 1) \geq 0$ and $\Delta U_{\mathcal{HN}}(p_{\mathcal{N}} = 1) \geq 0$ the claim follows. □

3.3 Correctness and Optimality

The following theorem proves the correctness of the mechanisms presented in Section 3.1. Its proof is the simple aggregation of the results presented in Section 3.2.

Theorem 5. *For any given system parameters, the values of $p_{\mathcal{A}}$ chosen after running the protocol depicted in Figure 3 satisfy that $P_{succ} \geq 1 - \varepsilon$.*

We now argue that only two approaches are feasible to bound the probability of accepting an incorrect value. In this respect, the strategy enforced by the mechanisms we designed is optimal.

Theorem 6. *In order to achieve $P_{succ} \geq 1 - \varepsilon$, the only feasible approaches are either to enforce a NE where $p_{\mathcal{H}} = 1$ or to use a $p_{\mathcal{A}}$ as shown in Lemma 2.*

Proof. It can be seen as in Lemma 4 that $\Delta U_{\mathcal{HC}}$ is non-increasing for $p_{\mathcal{C}} \in [0, 1-p_{\mathcal{N}}]$ and $\Delta U_{\mathcal{HN}}$ is non-increasing for $p_{\mathcal{N}} \in [0, 1-p_{\mathcal{C}}]$. Then, the only NE that can be made unique corresponds to $p_{\mathcal{H}} = 1$. Consider any other NE where $p_{\mathcal{H}} < 1$ (which is not unique). Then $p_{\mathcal{C}} = 1$ and $p_{\mathcal{N}} = 1$ are also both NE. In face of more than one equilibrium to choose from, different players might choose different ones. Thus, for the purpose of a worst case analysis with respect to the probability of correctness, it has to be assumed the worst case, i.e. $p_{\mathcal{A}}$ has to be set as in Lemma 2. □

3.4 Computational Issues

In Sections 3.1 and 3.2.3 we discussed a protocol for the master to choose appropriate values of $p_{\mathcal{A}}$ for different scenarios. A natural question is what is the computational cost of this protocol. In addition to simple arithmetical calculations, there are two kinds of relevant computations required: binomial probabilities and verification of conditions for Nash equilibria. Both computations are n-th degree polynomial evaluations and can be carried out using any of the well-known numerical tools [32] with polynomial asymptotic cost. These numerical methods yield only approximations, but all these calculations are performed either to decide in which case the parameters fit in, or to assign a value to $p_{\mathcal{A}}$, or to compare utilities. Given that these evaluations and assignments were obtained in the design as inequalities or restricted only to lower bounds, it is enough to choose the appropriate side of the approximation in each case.

Regarding the computational resources that rational workers require to carry out these calculations, notice that the choice of $p_{\mathcal{A}}$ in the mechanisms either yields a unique NE in $p_{\mathcal{H}} = 1$ or does not take advantage of the behavior of rational workers (Theorem 6). Furthermore, $p_{\mathcal{C}} = 1$ was assumed as a worst case (wrt probability of success). Notice from Tables 4–7 and the equilibrium conditions (eq. (1)) that setting $WP_{\mathcal{C}} = WB_{\mathcal{Y}} = 0$ for the cases where we do not use the behavior of the rational workers, $p_{\mathcal{C}} = 1$ is a dominant strategy. (Recall that $WB_{\mathcal{Y}}$ and $WP_{\mathcal{C}}$ can be chosen by the master.) Thus, the mechanisms are enriched so that rational workers are enforced to use always a unique NE, either $p_{\mathcal{C}} = 0$ or $p_{\mathcal{C}} = 1$. In order to make the computation feasible to the workers, the master sends together with the task a certificate proving such equilibrium. The certificate includes the strategy that the workers must play to achieve the unique NE together with the appropriate data to demonstrate this fact. These data include the system parameters/payoff values, the reward model and the values of $p_{\mathcal{A}}$, which is enough to verify uniqueness (recall the analysis in Section 3.2.3).

4 Putting the Mechanisms into Action

In this section two realistic scenarios in which the master-worker model considered could be naturally applicable are proposed. For these scenarios, we determine how to choose $p_{\mathcal{A}}$ and n in the case where the behavior of rational workers is enforced, i.e., under the conditions of Lemma 4. Again, for succinctness, the analysis of both mechanisms is presented for a number of replies k.

4.1 SETI-like Scenario

The first scenario considered is a volunteering computing system such as SETI@home, where users accept to donate part of their processors idle time to collaborate in the computation of large tasks. In this case, we assume that workers incur in no cost to perform the task, but they obtain a benefit by being recognized as having performed it (possibly in the form of prestige, e.g., by being included on SETI's top contributors list). Hence, we assume that $WB_{\mathcal{Y}} > WC_{\mathcal{T}} = 0$. The master incurs in a (possibly small) cost $MC_{\mathcal{Y}}$ when rewarding a worker (e.g., by advertising its participation in the project). As assumed in the general model, in this model the master may audit the values returned by the workers, at a cost $MC_{\mathcal{A}} > 0$. We also assume that the master obtains a benefit $MB_{\mathcal{R}} > MC_{\mathcal{Y}}$ if it accepts the correct result of the task, and suffers a cost $MP_{\mathcal{W}} > MC_{\mathcal{A}}$ if it accepts an incorrect value. Also it is assumed, as stressed before, that $d > 0$ (there is always a chance that the master will receive a reply from the worker).

Plugging $WC_{\mathcal{T}} = 0$ in the lower bounds of Lemma 4 it can be seen that, for this scenario and conditions, in order to achieve the desired P_{succ}, it is enough to set $p_{\mathcal{A}}$ arbitrarily close to 0 for all three models. So, we want to choose $\delta \leq p_{\mathcal{A}} \leq 1$, with $\delta \to 0$, so that the utility of the master is maximized. Using calculus, it can be seen that U_M is monotonic in such range, but the growth of such function depends on the specific instance of the master-payoff parameters. Thus, it is enough to choose one of the extreme values of $p_{\mathcal{A}}$. Replacing in Equation 3, we get

$$U_M \approx -\sum_{i=0}^{k-1} r_i MC_{\mathcal{S}} + \sum_{i=k}^{n} r_i \max\{\alpha_i, \beta_i\} \tag{12}$$

where $p_{\mathcal{N}} = 0$ and α_i, β_i as in Equation (3). The approximation given in Equation (12) provides a mechanism to choose $p_{\mathcal{A}}$ and n so that U_M is maximized for $P_{succ} \geq 1 - \varepsilon$ for any given worker-type distribution, reward model, and set of payoff parameters in the SETI scenario.

4.2 Contractor Scenario

The second scenario considered is a company that buys computational power from Internet users and sells it to computation-hungry costumers, such as Amazon's Mechanical Turk [4]. In this case the company pays the users an amount $S = WB_{\mathcal{Y}} = MC_{\mathcal{Y}}$ for using their computing capabilities, and charges the consumers another amount $MB_{\mathcal{R}} > MC_{\mathcal{Y}}$ for the provided service. Since the users are not volunteers in this scenario, we assume that computing a task is not free for them (i.e., $WC_{\mathcal{T}} > 0$), and that rational workers must have incentives to participate (i.e., $U > 0$). As in the previous case, we assume that the master verifies and has a cost for accepting a wrong value, such that $MP_{\mathcal{W}} > MC_{\mathcal{A}} > 0$. Also as before we assume that $d > 0$ and $p_N = 0$.

As mentioned before, using calculus it can be seen that U_M is monotonic on $p_{\mathcal{A}}$ but the growth depends on the specific instance of master-payoff parameters. Thus, the maximum expected utility can be obtained for one of the extreme values. Trivially, 1 is an upper bound for $p_{\mathcal{A}}$. For the lower bound, $p_{\mathcal{A}}$ must be appropriately bounded so that the utility of rational workers is positive and $P_{succ} \geq 1 - \varepsilon$. For example, for the $\mathcal{R}_{\emptyset}$ model, using Lemma 4 and conditioning $U > 0$, we get,

$$U_M = -\sum_{i=0}^{k-1} r_i MC_{\mathcal{S}} + \sum_{i=k}^{n} r_i \max\left\{\alpha_i, \beta_i + (\alpha_i - \beta_i)\frac{WC_{\mathcal{T}}}{d_2\, WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r'_i}\right\} \tag{13}$$

As in the previous section, the approximation given in Equation (13), and similar equations for the other reward models which are omitted for clarity, provide a mechanism to choose $p_{\mathcal{A}}$ and n so that U_M is maximized for $P_{succ} \geq 1 - \varepsilon$ for any given worker-type distribution, reward model, and set of payoff parameters in the contractor scenario.

4.3 Graphical Characterization of Master's Utility

In this section, in order to provide a better insight of the usability of our mechanisms, and to illustrate interesting trade-offs between reliability and cost, we provide a graphical characterization of the master's utility. Specifically we present and analyze various scenarios for the time-based and reply-based mechanisms, including the special case of reliable network (recall that in this case the two mechanisms converge), both in the SETI-like and the Contractor settings.

4.3.1 SETI-like Scenario

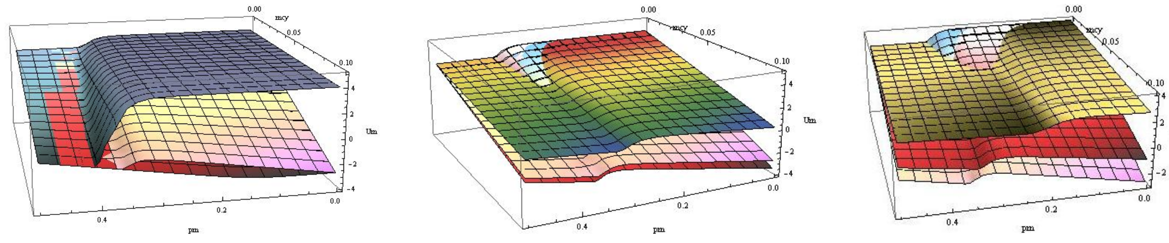
We begin by considering the timed-based mechanism, then the reply-based one, and then the special case of reliable communication where the two mechanisms converge (cf., Section 3.1).

Timed-based Mechanism. For this mechanism, we consider $MC_{\mathcal{A}} = 1$ as our normalizing parameter and we take $MP_{\mathcal{W}} = 100$, $MC_{\mathcal{S}} = 10$ and $MB_{\mathcal{R}} = 4$ as realistically large enough values (with respect to $MC_{\mathcal{A}} = 1$). Using other values for these parameters will not change qualitatively the results. We choose $p_\mu \in [0, 0.5]$ as we believe this is a reasonable interval. As it can be seen from the empirical evaluations of SETI-like systems reported in [15] and [18], p_μ is less than 0.1. So we took a larger range on p_μ to examine its general impact on the utility of the master. We choose $[0, 0.1]$ as the range of $MC_{\mathcal{Y}}$, to reflect the small cost incurred by the master for maintaining a workers contribution list.

We consider three plot scenarios were we vary p_μ and $MC_{\mathcal{Y}}$ as discussed above:
(a) We fix $d = 0.9$ and $n = 75$ and compute the master's utility for all three reward models. The results are depicted in Figure 4(a).
(b) We fix $n = 75$, we consider the $\mathcal{R}_{\mathrm{m}}$ model and compute the master's utility over $d = 0.5, 0.9, 0.99$. See Figure 4(b).
(c) We fix $d = 0.9$, we consider the $\mathcal{R}_{\mathrm{m}}$ model and we compute the master's utility over $n = 15, 55, 75$. The results are depicted in Figure 4(c).

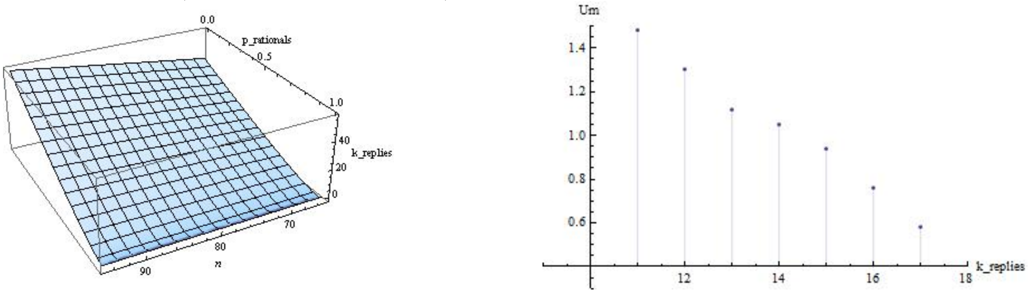
In all plots we can notice a threshold where the behavior of the utility changes. The threshold depicts the transition point in which the master changes its strategy from non-auditing to auditing.

In Figure 4(a) we can notice that for all the reward models, the master does not audit until p_μ gets around 0.35. This behavior is reasonable, since in the presence of more malicious workers the master must audit to ensure correctness. Once auditing, the utility of the master becomes the same in all three reward model, since now the same reward/penalize scheme is deployed. As expected, when the master does not audit, it gets its higher utility from $\mathcal{R}_{\emptyset}$ and its lower utility from $\mathcal{R}_{\mathrm{a}}$. The utility of the master



(a) (b) (c)

Figure 4: Time-based Mechanism in the SETI-like scenario: Master's utility for the three plot scenarios: (a) The upper plane corresponds to $\mathcal{R}_{\emptyset}$, the middle to $\mathcal{R}_{\mathrm{m}}$, and the third to $\mathcal{R}_{\mathrm{a}}$. (b) The upper plane corresponds to $d = 0.5$, the middle to $d = 0.9$, and the third to $d = 0.99$. (c) The upper plane corresponds to $n = 15$, the middle to $n = 55$, and the third to $n = 75$.



(a) (b)

Figure 5: Plots of the SETI-like Scenario for the Reply-based Mechanism

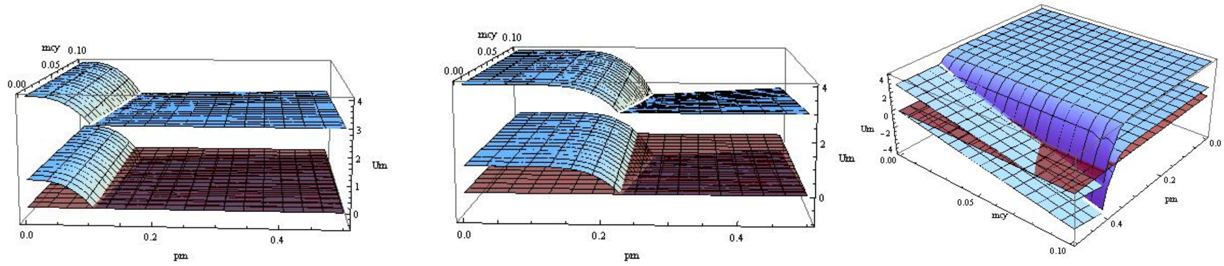
for the $\mathcal{R}_{\mathrm{m}}$ seems to balance nicely between the other two reward models. This perhaps suggests that the $\mathcal{R}_{\mathrm{m}}$ reward model is the most stable among the three. A final observation is that as $MC_{\mathcal{Y}}$ gets bigger, for $\mathcal{R}_{\mathrm{m}}$ and $\mathcal{R}_{\mathrm{a}}$ models, the utility of the master gets smaller; this is natural, since by increasing the payment to the workers the master is decreasing is own benefit.

In Fig 4(b) we can notice that for smaller values of d we get a higher utility for the master. This is due to fact that the master receives fewer replies, and hence it rewards a smaller number of workers. As with the previous plot scenario, for any d, as $MC_{\mathcal{Y}}$ is increasing, U_M is dropping. An important observation is that for $d = \{0.9, 0.99\}$ and for large values of $MC_{\mathcal{Y}}$, the utility of the master is higher as it audits. This is because the cost of rewarding the workers increases so much, that it is better for the master to audit.

In Figure 4(c) we notice that the utility of the master decreases as the number of workers increases; this is again due to the reward it must provide to the workers. Observer that for $n = 15$, the master chooses to change it's strategy to auditing for a smaller value of p_{μ}; this is due to the fact that as the master gets fewer replies, the probability of having a majority of incorrect replies gets bigger for smaller values of p_{μ}.

Reply-based Mechanism. We now provide a graphical characterization of the master's utility for the reply-based mechanism. Our aim is to observe how the minimum number of replies k will be affected by the number of workers selected by the master n, and by the probability distribution of rational workers p_{ρ}. Furthermore, we depict how k is affecting the utility of the master. As with the previous mechanism, we set $MC_{\mathcal{A}} = 1$, $MP_{\mathcal{W}} = 100$, $MC_{\mathcal{S}} = 10$ and $MB_{\mathcal{R}} = 4$.

We consider two plot scenarios:
(a) We vary n from 65 to 95, p_{ρ} for 0 to 1, and we compute the appropriate k that the master should choose for each n. The results are depicted in Figure 5(a).
(b) We use the $\mathcal{R}_{\mathrm{m}}$, we fix $p_{\rho} = 0.6$, $d = 0.9$, $MC_{\mathcal{Y}} = 0.05$, we vary k and we compute the utility of the master. See Figure 5(b).



(a) (b) (c)

Figure 6: Plots of the SETI-like scenario for $d = 1$. The upper plane corresponds to $MB_{\mathcal{R}} = 4$ the lower plane to $MB_{\mathcal{R}} = 1$ and the red flat plane to $U_M = 0$. (a) $n = 5$. (b) $n = 15$. (c) $n = 75$.

In Figure 5(a) we observe that as n increases, naturally, k increases as well. An interesting observation is that as p_ρ increases, k decreases. This is explained as follows: k is computed based on the number of malicious and altruistic workers that exist (since they always reply). Therefore, as these become fewer, k is naturally reduced.

In Figure 5(b) we observe how the utility of the master is affected by k; as k increases, the utility of the master decreases. This follows from the fact that as the master gets more replies, it has to reward more workers.

Reliable Network (Convergence of Mechanisms). We also provide the graphical characterization for the master's utility for the case that a reliable network exists ($d = 1$). From this simple case we can better study the trade-offs between reliability and cost without the complications of an unreliable network and workers not replying. By setting $d = 1$ we have the analysis for the SETI-like scenario for a reliable network; time-based and reply-based mechanisms converge to a single mechanisms where the master receives all replies from the workers. As before we set $MC_{\mathcal{A}} = 1$ and $MP_{\mathcal{W}} = 100$. Notice that in the reliable network case $MC_{\mathcal{S}}$ is not applicable and the probability of having this value is zero. We plot for values $p_\mu \in [0, 0.5]$ and $MC_{\mathcal{Y}} \in [0, 0.1]$. Recall that by plotting on the parameters the best strategy of the master is $p_A = 0$ or $p_A = 1$.

We consider three scenarios, applying the $R_\emptyset$ model and varying p_μ and $MC_{\mathcal{Y}}$ as discussed above. In particular:
(a) We fix n=5 and compute the utility of the master for $MB_{\mathcal{R}} = \{1, 4\}$; the results are depicted in Figure 6(a).
(b) We fix n=15 and compute the utility of the master for $MB_{\mathcal{R}} = \{1, 4\}$; the results are shown in Figure 6(b).
(c) We fix n=75 for both values of $MB_{\mathcal{R}}$ mentioned earlier; in Figure 6(c) are depicted the corresponding results.

All plots include a reference surface plane $U_M = 0$. Here we have only presented the $R_\emptyset$ model because it is the simplest one. However, for the other reward models the plots depict more or less the same behavior, with the difference that before the threshold point (where the master does not audit) the utility of the master also depends on $MC_{\mathcal{Y}}$ (e.g. Figure 4(c)).

A natural and expected observation in Figure 6, is the fact that the higher the value of $MB_{\mathcal{R}}$ the higher the utility of the master without this affecting the shape of the plot. In all plots we can notice a threshold where the behavior of the utility changes. The threshold depicts the transition point in which the master changes its strategy from non-auditing to auditing. For all three plots in Figure 6, we generally observe a smaller utility when the master audits than when it does not. Recall that we apply the $R_\emptyset$ model when the master follows a non-auditing strategy; thus the master rewards the honest workers only when it audits and this decreases its own utility proportionally to the value of payment to the workers ($MC_{\mathcal{Y}}$). Another interesting observation about the plots in Figure 6 is the sharp declining curve before the threshold (the master follows a non-auditing strategy). This curve is due to the fact that as p_μ increases the probability of the master getting an incorrect reply increases, and thus the utility of the master decreases accepting an incorrect reply. Notice that this declining curve is much sharper in Figure 6(c), since the larger the number of workers the more acute the impact of a high p_μ.

A significant difference between the number of chosen workers, is the threshold value of p_μ where the master changes its strategy to auditing. The larger the number of workers, the bigger the transition value (p_μ value) that the master starts to audit. This is due to the large reward it must provide when it audits, combined with the fact that having more workers increases the probability of getting the correct reply. We also notice that U_M increases slightly after the threshold, as p_μ increases. Although this behavior is not expected, we believe it is due to the fact that the master has resolved to auditing in order to guarantee getting the correct value, and thus the fewer honest workers it has to reward, the greater its benefit.

4.3.2 Contractor Scenario

We now consider the contractor scenario (e.g., Amazon's Mechanical Turk). Recall that in this setting $WC_{\mathcal{T}} > 0$, and the workers are willing to participate only if their utility if positive (they are not volunteers as in the SETI-like setting). For this scenario we focus on the special case of reliable communication (where the two mechanisms converge) to illustrate how the cost for computing the task ($WC_{\mathcal{T}}$) affects the trade-offs between reliability and cost (which we could not study in the SETI-like setting).

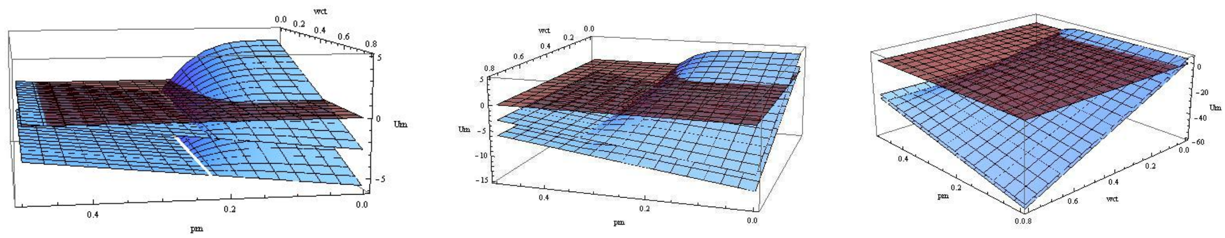
Figure 7 illustrates the utility of the master for the $R_\emptyset$ model and for a fix value of $S = 0.8$; we vary $p_\mu \in [0, 0.5]$ and $WC_{\mathcal{T}} \in [0, S]$. In Figure 7(a) we fix n=7, in Figure 7(b) we fix n=15 and in Figure 7(c) we fix n=75. For each of these plots we have two planes, one for each value of $MB_{\mathcal{R}} = \{1, 4\}$ and a reference surface plane $U_M = 0$ (similarly to the plots for the reliable communication case in the SETI-like setting).

Observe that a threshold point exists where the master changes its strategy from auditing with some probability (that guaranties the utility of the rational workers is positive) to auditing. We generally observe that (not surprisingly) for values of p_μ and $WC_{\mathcal{T}}$ close to zero we get the highest utility.

In all plot in Figure 7 when the master audits with some probability (before the threshold point) observe that as $WC_{\mathcal{T}}$ increases, the utility of the master decreases for every p_μ. This is a classical example of the trade-off between reliability and cost. The larger $WC_{\mathcal{T}}$ is, the higher the probability of $p_{\mathcal{A}}$ should be to guarantee correctness, thus the utility of the master decreases.

Another observation (especially in Figure 7(c)), is that before the threshold value, as p_μ increases, the utility of the master increases, and then decreases for every value of $WC_{\mathcal{T}}$ (except when close to $WC_{\mathcal{T}} = 0$ and $WC_{\mathcal{T}} = S$)! When p_μ is increasing, the number of truthful workers decreases thus the master has to reward less honest workers and so its utility increases; recall that the master audits the answers with some probability. On the other hand, when the value of p_μ increases even more, the probability of having a majority of incorrect answers is very large. So it is quite probable since the master audits with some probability to get an incorrect result; thus its utility decreases.

Naturally when the master audits, for every value of $WC_{\mathcal{T}}$, as p_μ increases so does the utility of the master. The higher the p_μ, fewer the honest workers, and thus the smaller the total payment of the master to the workers. Notice again that having larger $MB_{\mathcal{R}}$ does not affect the shape of the plots; the utility of the master increases uniformly. For similar reasons as in the reliable network SETI-like setting, the threshold value (p_μ value) increases for larger number of workers. Finally, observe the big decrease in the master's utility as the number of workers grows. This is due to the large payments that the master has to give to large groups of workers to guarantee reliability.



(a) (b) (c)

Figure 7: Contractor Scenario plots for fixed S and $d = 1$. The upper plane corresponds to $MB_{\mathcal{R}} = 4$ the lower plane to $MB_{\mathcal{R}} = 1$ and the red flat plane to $U_M = 0$. (a) $n = 7$. (b) $n = 15$. (c) $n = 75$.

5 Discussion

In this paper we have combined a classical distributed computing approach (voting) with a game-theoretic one (cost-based incentives and payoffs). This has lead to designing and analyzing two mechanisms that enable a master process to reliably obtain a task result despite the co-existence of malicious, altruistic and rational workers, and the underlying network's unreliability.

Several future directions emanate from this work. For example, in this work we have considered a cost-free, weak version of worker collusion (all rational cheaters and malicious workers return the same incorrect task result). It would be interesting to study more involved collusions, as the ones studied in [2] or [12]. In this work, we have considered a single-task one-shot protocol, in which the master decides which task result to accept in one round of message exchange with the workers. It would be interesting to consider several task waves over multiple rounds, that is, view the computation as an *Evolutionary Game* [31, 56]. The master could use the knowledge gained in the previous rounds to increase its utility and its probability of success in future rounds. Issues such as worker *aspiration level* [9] could be taken into account.

Acknowledgments. We thank Alexander A. Shvartsman for insightful discussions.

References

[1] I. Abraham, L. Alvisi, and J.Y. Halpern. Distributed computing meets game theory: Combining insights from two fields. *ACM SIGACT News: Distributed Computing Column*, 42(2):69–76, 2011.

[2] I. Abraham, D. Dolev, R. Goden, and J.Y. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *proc. of PODC 2006*, pp. 53–62.

[3] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J. Martin, and C. Porth. BAR fault tolerance for cooperative services. In *proc. of SOSP 2005*, pp. 45–58.

[4] Amazon's Mechanical Turk, `https://www.mturk.com`.

[5] D. Anderson. BOINC: A system for public-resource computing and storage. In *proc. of GRID 2004*, pp. 4–10.

[6] M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency. In *proc. of ACM EC 2006*, pp. 18–28.

[7] M. Babaioff, M. Feldman, and N. Nisan. Free riding and free labor in combinatorial agency. In *proc. of SAGT 2009*.

[8] M. Babaioff, M. Feldman, and N. Nisan. Mixed Strategies in Combinatorial Agency. In *proc. of WINE 2006*, pp. 353–364.

[9] J. Bendor, D. Mookherjee and D. Ray. Aspiration-based reinforcement learning in repeated interaction games: An overview. *International Game Theory Review*, 3:159–174, 2001.

[10] S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. In *proc. of SODA 2007*, pp. 169–178.

[11] G. Christodoulou and E. Koutsoupias. Mechanism design for scheduling. *Bulletin of the EATCS*, 97:39–59, 2009.

[12] J. R. Douceur. The Sybil attack. In *proc. of IPTPS 2002*, pp. 251–260.

[13] W. Du, J. Jia, M. Mangal, and M. Murugesan. Uncheatable grid computing. In *proc. of ICDCS 2004*, pp. 4–11.

[14] R. Eidenbenz and S. Schmid. Combinatorial agency with audits. In *proc. of GameNets 2009*.

[15] "Einstein@home", `http://einstein.phys.uwm.edu`.

[16] "Enabling Grids for E-sciencE", `http://www.eu-egee.org`.

[17] K. Eliaz. Fault tolerant implementation. *Review of Economic Studies*, 69:589–610, 2002.

[18] T. Estrada, M. Taufer, and D. P. Anderson. Performance prediction and analysis of BOINC projects: An empirical study with EmBOINC. *Journal of Grid Computing,* 7(4):537–554, 2009.

[19] W. Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 3rd edition, 1968.

[20] A. Fernández, Ch. Georgiou, L. Lopez, and A. Santos. Reliably executing tasks in the presence of untrusted processors. In *proc. of SRDS 2006*, pp. 39–50.

[21] A. Fernández Anta, Ch. Georgiou, and M. A. Mosteiro. Designing mechanisms for reliable Internet-based computing. In *proc. of NCA 2008*, pp. 315–324.

[22] I.T. Foster and A. Iamnitchi. On death, taxes, and the convergence of P2P and grid computing. In *proc. of IPTPS 2003*, pp. 118–128.

[23] D. Fotakis. Memoryless facility location in one pass. In *proc. of STACS 2006*, pp. 608–620.

[24] M. Gairing. Malicious Bayesian congestion games. In *proc. of WAOA 2008*, pp. 119–132.

[25] P. Golle and I. Mironov. Uncheatable distributed computations. In *proc. of CT-RSA 2001*, pp. 425–440.

[26] M. Halldorsson, J.Y. Halpern, L. Li, and V. Mirrokni. On spectrum sharing games. In *proc. of PODC 2004*, pp. 107—114.

[27] J.Y. Halpern. Computer science and game theory: A brief survey. *Palgrave Dictionary of Economics*, 2007.

[28] J.Y. Halpern and V. Teague. Rational secret sharing and multiparty computation. In *proc. of STOC 2004*, pp. 623–632.

[29] J. C. Harsanyi. Games with incomplete information played by Bayesian players, I, II, III. *Management Science*, 14:159–182, 320–332, 468–502, 1967.

[30] E.M. Heien, D.P. Anderson, and K. Hagihara. Computing low latency batches with unreliable workers in volunteer computing environments. *Journal of Grid Computing*, 7:501–518, 2009.

[31] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[32] W. G. Horner. A new method of solving numerical equations of all orders by continuous approximation. *Philos. Trans. Roy. Soc. London* 109:308–335, 1819.

[33] D. Kondo, F. Araujo, P. Malecot, P. Domingues, L. Silva, G. Fedak, and F. Cappello. Characterizing result errors in Internet desktop grids. In *proc. of Euro-Par 2007*, pp. 361–371.

[34] D. Kondo, H. Casanova, E. Wing, and F. Berman. Models and scheduling mechanisms for global computing applications. In *proc. of IPDPS 2002*, pp. 79–86.

[35] K.M. Konwar, S. Rajasekaran, and A.A. Shvartsman. Robust network supercomputing with malicious processes. In *proc. of DISC 2006*, pp. 474–488.

[36] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky. SETI@home: Massively distributed computing for SETI. *Computing in Science and Engineering*, 3(1):78–83, 2001.

[37] E. Koutsoupias and Ch. Papadimitriou. Worst-case equilibria. In *proc. of STACS 1999*, pp. 404–413.

[38] H. C. Li, A. Clement, M. Marchetti, M. Kapritsos, L. Robison, L. Alvisi, and M. Dahlin. FlightPath: Obedience vs Choice in Cooperative Services. In *proc. of USENIX OSDI 2008*, pp. 355–368.

[39] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR gossip. In *proc. of OSDI 2006*, pp. 191–204.

[40] G. Mailath and L. Samuelson. *Repeated Games and Reputations: Long-run Relationships*. Oxford University Press, 2006.

[41] A. Mass-Colell, M. Whinton, and J. Green. *Microeconomic Theory*, Oxford University Press, 1995.

[42] M. Mavronicolas and P. Spirakis. The price of selfish routing. *Algorithmica*, 48(1):91–126, 2007.

[43] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[44] T. Moscibroda, S. Schmid, and R. Wattenhofer. When selfish meets evil: byzantine players in a virus inoculation game. In *proc. of PODC 2006*, pp. 35–44.

[45] J.F. Nash. Equilibrium points in n-person games. *National Academy of Sciences*, 36(1):48–49, 1950.

[46] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[47] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[48] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2003.

[49] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of ACM*, 49(2):236–259, 2002.

[50] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[51] L. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, 18(4):561–572, 2002.

[52] J. Shneidman and D.C. Parkes. Rationality and self-interest in P2P networks. In *proc. of IPTPS 2003*, pp. 139–148.

[53] *Tables of Probability Functions*, Volume 2, National Bureau of Standards, 1942.

[54] M. Taufer, D. Anderson, P. Cicotti, and C. L. Brooks. Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing. In *proc. of IPDPS 2005*.

[55] "TeraGrid", `http://www.teragrid.org`.

[56] J.W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, 1995.

[57] S. Wong. An authentication protocol in web-computing. In *proc. of IPDPS 2006*.

[58] M. Yurkewych, B.N. Levine, and A.L. Rosenberg. On the cost-ineffectiveness of redundancy in commercial P2P computing. In *proc. of CCS 2005*, pp. 280–288.