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. Building on prior work we consider a framework where altruistic, malicious and rational workers co-exist. 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.

The master must obtain the correct task result while maximizing its benefit. Adding on that work, we consider the possibility that the communication between the master and the workers is not reliable, and that workers could be unavailable; assumptions that are very realistic for Internet-based master-worker computations. Within this framework 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. This analysis identifies trade-offs between reliability and cost, under different system parameters.

ALGORITHMIC MECHANISMS FOR RELIABLE MASTER-WORKER INTERNET-BASED COMPUTING UNDER COMMUNICATION UNCERTAINTY

Evgenia Christoforou

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Cyprus

Recommended for Acceptance

by the Department of Computer Science

June, 2012

APPROVAL PAGE

Master of Science Thesis

ALGORITHMIC MECHANISMS FOR RELIABLE MASTER-WORKER INTERNET-BASED COMPUTING UNDER COMMUNICATION UNCERTAINTY

Presented by

Evgenia Christoforou

Research Supervisor _____
Dr. Chryssis Georgiou

Committee Member _____
Dr. Anna Philippou

Committee Member _____
Dr. George Pallis

University of Cyprus

June, 2012

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Chryssis Georgiou for his unwavering support and trust and his valuable guidance. A special thanks goes to Antonio Fernández Anta and Miguel A. Mosteiro for their helpful collaboration and share of expertise. I would also like to thank Angel (Anxo) Sánchez for inspiring discussion on future work. Last but not least, I want to thank my family for their support and encouragement.

This work is supported by the Cyprus Research Promotion Foundation Grant ΤΠΕ/ΠΛΗΡΟ/0609(ΒΕ)/05.

CREDITS

This work is composed of results of the following published articles:

- Evgenia Christoforou, Antonio Fernandez Anta, Chryssis Georgiou and Miguel Mosteiro, Algorithmic Mechanisms for Internet Supercomputing under Unreliable Communication, 10th IEEE International Symposium on Network Computing and Applications (NCA 2011), Cambridge, MA, pp. 275 - 280, Aug, 2011.

- Evgenia Christoforou, Antonio Fernandez Anta, Chryssis Georgiou and Miguel Mosteiro, Brief Announcement: Algorithmic Mechanisms for Internet-Based Computing under Unreliable Communication, 25th International Symposium on Distributed Computing (DISC 2011), Rome, Italy, pp. 147-149, Sept, 2011.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 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 [21,68], the "@home" projects [6], such as SETI [44], Amazon's Mechanical Turk [5], and peer-to-peer computing–P2PC [28, 71]. Although the potential is great, the use of Internet-based computing is limited by the untrustworthy nature of the platform's components [6, 31, 36]. Let us take SETI as an example. In SETI, searching for extraterrestrial intelligence demands the analysis of gigabytes of raw data. Using expensive supercomputing parallel machines would not be efficient, thus 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 [6]) but there are also other methods [17, 42, 70].

The same group at U.C. Berkeley Spaces Sciences Laboratory that developed the original SETI@home has also developed BOINC (Berkeley Open Infrastructure for Network Computing) [6]. BOINC is an open source middleware system for volunteer computing. The goal is to provide researchers with enormous processing power from participating personal computers around the world. Among other projects supported by the BOINC platform are the Folding@home, the Einstein@home and the AIDS@home. With BOINC an application can submit to the system a task to be executed. Instances of the task are going to be sent to several workers (redundant task allocation is applied) and a validation process is going to decide upon the correct answer. The system is configured as to which validation process to follow and how many instances of a task to create. Usually systems like BOINC are referred to as desktop grids or computational grids in the bibliography.

Another popular master-worker Internet-based application is Amazon's Mechanical Turk [5]. Here the master and the workers can be in fact humans that contribute time for solving problems in exchange to economic rewards. A person who wishes to have a problem (task) solved can act as a master processor and hire worker processors (other persons) through the Mechanical Turk platform and have its task computed.

The problem of having untrustworthy workers has been studied under two different views: from a "classical" distributed computing view [25,43,63] and from a game-theoretic view [26,71]. 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, 31, 64]. 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, 14, 58] 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) [57], i.e., a strategy choice by each worker such that none of them has an 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 [59, 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 work proposed by Fernández et al. [27] considers the co-existence of all three types of workers, since in a massive computation platform such as the Internet this co-existence is very probable. An Internet-based master-worker framework was considered where a master processor assigns, over the Internet, a computational task to a set of untrusted worker processors and collects their responses, a reliable network was considered. Under this framework, a game-theoretic

mechanism was designed that provided necessary incentives to the rational workers to compute and report the correct task result despite the malicious workers actions. The objective of the mechanism was to maximize the probability of the master of obtaining the correct task result while minimizing its cost (or alternatively, increasing its benefit). The utility of the mechanism was demonstrated by applying it to two paradigmatic applications: a volunteer computing system (such as SETI) and a contractor-based system (such as Amazons mechanical turk).

This work extends the master-worker framework of [27] by additionally considering the possibility that the communication between the master and the workers is not reliable, an assumption that is reasonable when considering Internet-based master-worker computations. 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. [36] 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 [41], 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 now possible for a worker's reply not to reach the master, we additionally extend the framework of [27] by allowing workers to abstain from the computation. In [27] workers did not have the choice of abstaining. 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 (Chapter 3), 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. This unreliability of the network and workers abstaining, inquires often a large time to run tasks, something that justifies the use of single-round mechanisms (single-round mechanism was also used in [27]).

1.2 Contributions

In this work, building on the work in [27], we identify, with provable analytical guarantees, the tradeoffs between the master obtaining the correct task result, the cost of doing so, and the reliability of the underlying communication network. In particular:

- We extend the framework of [27] by considering network unreliability, modeled by a parametric probability. Furthermore, we extend the strategic space of rational workers, and besides the choice of being honest or cheaters, workers can also choose to abstain from the computation. Also a new parameter is added to the set of realistic payoff parameters considered in [27] (Chapter 3). The reward model used in [27], where the master rewards the majority and punishes the minority is no longer used is this work (the reward models are presented in Chapter 3) due to the unreliability of the network; when no reply from the worker is received no assumptions can be made as to why.

- 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 (Chapter 4). The algorithms are parametrized in terms of a probability of auditing $p_{\mathcal{A}}$ (defined in Chapter 3) as in [27] and d, a parametric probability modeling networks unreliability. Each of the algorithms implement an instance of the Bayesian game [35]. 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.

Following the same analysis pattern as in [27], 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 under the network's unreliability is also designed (Chapter 4). 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.

- Following [27], under the constraint of the bounded probability of success, it is shown how to maximize the master utility in two real-world scenarios (Chapter 5), when having an unreliable network and workers abstaining from the computation. The first scenario abstracts a system of volunteering computing like SETI [44]. The second scenario abstracts a contractor-based application where a company buys computational power from Internet users and sells it to computation-hungry consumers, such as Amazon's Mechanical Turk.

  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 [20] and [23].

1.3 Document Organization

Summarizing, in Chapter 2 a literature review of previous works and how they relate with each other and our work is presented. In Chapter 3 the model and definitions are presented. 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 as in [27]. Chapter 4 presents the algorithms and protocol followed by the master, comprising the mechanism that provides rational worker with the appropriate incentives for the master to achieve its goal. Chapter 5 shows how the mechanism is applied to two realistic scenarios and presents plots derived from those scenarios characterizing the utility of the master. Finally in Chapter 6 we discuss the presented work and what are possible future directions.

Chapter 2

Literature Review

This work uses a game-theoretic approach combined with a classical distributed computing approach to achieve a reliable master-worker computation in the presence of communication failures. In this section we provide a literature review of prior existing work to better indicate the contribution of our work.

2.1 Traditional Distributed Computing Approach

The work of Sarmenta [63] address the problem of protecting volunteer systems like SETI@ home from malicious workers, called saboteurs. Traditional techniques like checksums and cryptographic techniques will not prevent the actions of malicious workers because the errors created are intentional. This work assumes that workers are malicious with probability f, and that a malicious worker returns a wrong answer with probability s. This work considered a collection of tasks. The objective is to bound the expected number of wrong results accepted by the master, and the amount of work performed. Several mechanisms to achieve this objective are compared, the traditional voting technique taking the majority for each task and also they develop a new technique called spot-checking. With spot-checking workers are given tasks whose result is known to

identify the workers that are not truthful. This mechanism can be combined with blacklisting or credibility techniques.

In the work of Kondo et al. [41] the error detection mechanisms presented in the work of Sarmenta are evaluated with real data gathered from the XtreamLab project that uses the BOINC infrastructure. Through this work the errors generated in desktop grid applications are characterized. They conclude that a large fraction of errors is coming from a very small portion of workers. Also they show that little correlation between simultaneous malicious workers exist. Additionally, there is a large variability of the set of malicious workers over time, with the exception of a few frequent offenders. Also care has to be taken if blacklisting or credibility is used. They derive the conclusion that a large number of task and time are required to achieve low error rates with spot-checking and that, in general, to achieve low error rates it is better to use majority something used in our model as well.

In the work of Fernández et al. [25] an Internet-based master-worker computing that considers the presence of malicious workers is presented. An asynchronous distributed system is considered where the master processor sends tasks to a collection of n workers and a worker may deliberately return an incorrect result in an effort to harm the master. The authors model with an explicit parameter d the probability that the master will receive the reply from a worker on time. The master in an attempt to receive the correct result sends the same task to several workers and decides upon the correct result based on the received replies. In order to analyze a worst case scenario the assumption that malicious workers reply and return the same incorrect reply is made (full collusion). For each task assigned to a worker the master is charged with one work unit. The goal of the master is to accept the correct task result with high probability of success $1 - \varepsilon$, where $\varepsilon \ll 1$ and with the smallest possible amount of work. A probabilistic bound on the number of malicious workers is considered, with a probability $p < 1/2$ of any worker processor being

faulty. Lower bounds on the minimum amount of (expected) work required are given, so that any algorithm accepts the correct reply with probability of success $1 - \varepsilon$.

A single round protocol is used, the master decides upon the correct reply by the end of the round. Two algorithms for the master using different decision strategy are developed, a majority based and a threshold based algorithm. The majority based algorithm send the task to a subset of the workers and after time T decides upon the correct result taking the majority of received replies. The threshold based algorithm is an early termination algorithm. If the master receives a certain number of replies with the same value it makes a decision, otherwise it decides on the majority of received responses by time T (like the majority based algorithm). The authors show that both algorithms obtain the same probability of success $1 - \varepsilon$ and derive similar upper bounds on the (expected) work required in doing so. Also under certain conditions they show that these upper bounds are asymptotically optimal with respect to the lower bounds. So the authors have managed to show that it is possible, with provable analytical guarantees, to execute tasks reliably with high probability and low cost in the presence of malicious worker processors and unreliable communication.

The work proposed by Fernández et al. [25] takes into account the conclusions derived by the work of Kondo et al., that in general achieving low errors it is better to use majority. Also considers the unreliability of the network and unavailability of the workers something that is not considered in the work of Sarmenta. Our work is inspired by the way the unreliability of the network is introduced and uses the same parametric probability d for modeling the unreliability of the network. Although in our work we explicitly consider this probability d to be the product of two separate probabilities that model the unreliability of communication between master and worker. Our work takes a game theoretic approach considering the presence of rational workers that do not have a predefined behavior and we model the unavailability of the worker as a strategic

choice given to the rational workers. In our work the performance measure is not minimizing the number of workers to whom the master assigns the task, but maximizing the utility of the master. Due to the very different objectives and mechanisms deployed in [25] the results of this work with those of traditional distributed computing, cannot be compared even if no rational workers are assumed.

The work of Konwar et al. [43] studies an extension of the problem considered in the work of Fernández et al. [25] in which there are n workers and n tasks to be performed. But a synchronous system is assumed in which the result of a task assigned to a non-faulty worker is always received by the master. This makes their computation model somewhat stronger in comparison with [25] where a more realistic model of unreliable communication is assumed. But having a synchronous system allows them to obtain efficient algorithms even if the failure parameter p (faulty workers) is unknown.

In the work of Kuhn et al. [46] a classical distributed approach is taken considering the workers on a computational grid to either be altruistic or malicious in the sense that they intensionally cheat for their own benefit. They develop a distributed checking mechanism that detects wrong results, excludes malicious workers and thus prevents malicious workers from obtaining rewards. The authors have incorporated their distributed checking mechanism in the BOINC server software. The goal of their mechanism is to give rewards only to altruist and not flood the system with wrong results, also all malicious workers is assumed that form a single coalition.

While this line of work has a partially similar goal of the master receiving the correct result and considers the same a desktop grid framework the approach that we take differs completely. We have a central auditing mechanism and we are not concerned with cheater receiving a reward, as long as the master receives the correct result while maximizing its utility. We take a game theoretic approach and assume also that the network is unreliable something that this work does

not consider. Also in the work of Kuhn et al. [46] an upper bound of 20% of malicious worker has to be assumed for the checking mechanism to verify the correctness of a result. This bound on the malicious workers is smaller than the one assumed in [25]. In our work such limitation does not apply due to the auditing technique we use.

2.2 Algorithmic Game Theoretic Approach

As we already mentioned in Section 1.1, this is not the first work to consider master-worker computations in a game-theoretic model; this approach has been studied before [26, 27, 71]. The work by Yurkewych et al. [71] considers computational grids where clients are financially compensated for their work. The authors assume that they deal with rational workers, that are seeking only to maximize their expected profit; that is, are not the workers malicious. Workers follow a cheating strategy only if that increases their expected profit compared to an honest strategy, they are not considered risk-seeking. Because workers can not be trusted the authors use redundant task allocation, send the same task to several workers and collect their replies. Also they assume that workers can form colluding team returning the same incorrect result. Since workers are rational in the game theoretic sense, the master can audit there replies, giving reason to the workers to follow a truth telling strategy. This work considers the presence of a reliable communication, in our work we take a realistic approach considering an unreliable network and also the presence of malicious and altruistic workers. These assumptions demand that our mechanism can cope by giving the appropriate incentives that will force the good behavior of the rational workers.

In their work [71], the authors develop a game which they call the Internet Auditing Game. The master chooses a set of workers, announces the probability by which it will audit and send the task to the workers. If the master audits then it rewards honest workers and penalizes cheaters. If it does not audit and there is a more than half of the workers that gave an identical reply (consensus),

it rewards the members of the majority and penalizes the rest. If there is no consensus the master chooses again randomly a set of workers and re-initiates the same process with the same task. That is, not always a single-round protocol is followed, while our work follows a single round protocol. As pointed out experimentally in the work of Kondo et al. [41] task may take more than one day of CPU time to complete. This make the model of Yurkewych et al. vulnerable to long execution time.

The master in [71] has a fixed budget for computing a task, that includes the auditing cost and the rewards to the workers cost. The goal of the server is to guarantee that it will get the correct reply and within its budget. Bounds for the audit probability are computed to guarantee that workers have incentives to be honest in three scenarios: redundant allocation with and without collusion, and single-worker allocation. They authors conclude, that single-worker allocation is a cost-effective mechanism specially in presence of collusion. In our work we do not restrict the budget of the master and we are able to show useful tradeoffs between reliability and cost.

A later work by Fernández et al. [26] also considers Internet-based master-worker computations from a game-theoretic point of view. These computations are modeled as games where each worker is assumed to be rational and can choose to cheat, fabricate a reply and return it to the master, or be honest compute and return the correct result. A general single-round protocol is followed where the master assigns the task to n workers. Each worker processor cheats with a probability and the master verifies the answer with some probability. If the master verifies it rewards and punishes workers appropriately. If the master does not verify then it rewards the workers according to one of three reward models: (a) a reward majority model $\mathcal{R}_{\mathrm{m}}$, where the majority of workers is rewarded, (b) a reward all model $\mathcal{R}_{\mathrm{a}}$ where the master rewards all workers and (c) a reward none model $\mathcal{R}_{\emptyset}$ where the master does not reward at all.

Cost-sensitive mechanisms that provide the necessary incentives for the workers to truthfully compute and return the correct result are designed. The objective is to maximize the probability that the master will obtain the correct task result while minimizing its cost. For this purpose the authors consider a set of realistic payoff parameters that can model the environment considered in a game-theoretic sense. Four different games are considered: (a) a game between the master and a single worker ($1:1$ game), (b) a game between the master and a workers played n times, each with a different worker ($1:1^n$ game), (c) a game with a master and n workers ($1:n$ game) and finally, (d) a game with n worker and the master participating indirectly ($0:n$ game). Combined with the three reward models the authors have considered twelve games. The authors analyze the conditions under which, with general payoff parameters, unique NE is reach for each of the twelve games. Thus the analysis leads to mechanisms where the master can choose the game conditions that guarantee a unique NE that best fits its goal. Finally they have identified and proposed specific mechanism for two realistic scenarios, a volunteer computing scenario (e.g. SETI) and a company that buy computational cycles from Internet computers and sells them to customers in the form of a task-computation service.

The authors consider a weak form of collusion, we assume the same in our work, where the all cheaters return the same incorrect result. This assumption is not made in the work of Yurkewych et al. where a general form of colluding workers is assumed, this is also the reason why their protocol may become a multiple-rounds protocol. Also they make the assumption that if a worker does not perform the task, then it is almost impossible to guess the correct answer. In our work no such assumption is made, since we use auditing the master computes the task by its self. This work [26] uses a verification technique, since it considers that verifying is more efficient than computing the task (i.e. auditing). The master by verifying does not necessarily obtain the correct result, for example when all workers cheat. A survey contacted on this problem [15] for the

model considered in the work of Fernández et al. [26] compared the verification technique used with an auditing technique. The advantage of auditing is that the master will receive the benefit from getting the correct answer even if all workers cheated, on the other hand auditing can be costly for the master. In [15] a thorough study was contacted comparing the two method for different auditing/verification costs over the different games considered. The general result was that auditing had a better master utility except in the 0:n game.

Compared with the work of Yurkewych et al., the work of Fernández et al. [26] studies more algorithms and games, considers richer payoffs, probabilistic cheating and shows a reacher trade-offs between reliability and cost. But as we mentioned above the use of verification instead of auditing makes it weaker. Also a weaker form of collusion is assumed but the results reached are similar with the one of Yurkewych et al.; under certain conditions non redundant task allocation is best.

In their work Fernández et al. [26] did not considered the presence of non-intensional errors produced by hardware or software problems. In a later published work Fernández et al. [27] consider the presence not only of rational but also of malicious and altruistic workers in an Internet-based master-worker computation. Malicious workers have a predefined ‘‘bad’’ behavior that is due to hardware or software errors or a deliberate malicious behavior. Altruistic workers always compute and return the correct result, that is they exhibit a ‘‘good’’ behavior. Rational workers are defined in the same way as in their previous work, they do not have a priori established behavior. They act based on their own self-interest, decide to be honest or cheat based on the strategy that will increase their benefit. By considering all the three types of workers, a rich combination of game-theoretic and classical distributed computing approaches to the design of mechanisms for reliable Internet-based master-worker computing, is given.

A probability distribution of workers among the worker types is considered. The authors assume that the master and the workers do not know the type of other workers, only the probability distribution. The rational workers will play a game looking for a NE, while malicious and altruistic workers have a predefined strategy to cheat or be honest, respectively. The master does not participate in the game, rather it designs the game to be played. The game played is formalized as a game with imperfect information (i.e. Bayesian game). A collection of realistic payoff parameters and reward models are identified.

The goal of the master is to guarantee that it will obtain the correct task result with probability at least $1-\varepsilon$ (where$0 \leq \varepsilon < 1$), and having achieved this the master wants to maximize its benefit. The mechanism designed gives weapons to the master to achieve its goal, it can compute the task its self and reveal the cheaters (i.e. use audit) and reward/ punish the workers appropriately or it will not audit and reward/punish according to a reward model, thus encouraging the rational workers to be honest.

A general voting algorithm is run by the master to implement the game mentioned. The algorithm is parametrized by the auditing probability $p_{\mathcal{A}}$. The master sends the task, its audit probability and a certificate to all workers. The assumption is made that the communication between master and workers is reliable and that all workers reply to the master. Thus upon the master receiving all answers from the workers, it audits the answers with a probability $p_{\mathcal{A}}$. If the master does not audit it accepts the majority of answers and follows a chosen reward model. As in their previous work and as in this work, the assumption is made that all cheater return the same incorrect reply. If the master audits, it rewards the honest workers and penalized the cheaters. The master can follow one of four reward models defined. A reward model that rewards the majority and penalizes the minority (this model was not included in the authors previous work), a model

that rewards only the majority, a model that rewards all workers or finally a reward model that does not reward any worker.

In that work [27], and under a general type probability distribution, the authors analyze the master's utility and probability of error and identify the conditions under which the game has NE. Also they identify an algorithmic mechanism through which the master chooses the values of the auditing probability to guarantee its goal. The mechanism to choose $p_{\mathcal{A}}$ is designed taking into account two scenarios, a free rationals scenario and a guided rationals scenario, as they call it. In the guided rationals scenario, the type distribution is such that the rational workers have to be enforced to follow an honest strategy. In the free rational scenario, due to the type distribution again, the behavior of the rational workers does not need to be enforced. The designed mechanism is general in that reward models can either be fixed exogenously or be chosen by the master. The authors show that this mechanism is the only feasible approach for the master to achieve a given bound on the probability of error. As in their previous work, Fernández et al. apply their mechanism to two realistic scenarios, a volunteer-like scenario and a contractor scenario. Under the constrain of the bounded probability of error, they illustrate how to maximize the utility of the master.

In that work [27] as well as in the previous ones [26, 71] a reliable network is assumed. Our work compliments the work of Fernández et al. [27] by considering an unreliable network and also allows rational workers to abstain the computation. As mentioned before making these realistic assumptions makes the task of the master even more challenging, since the incentives provided by the mechanism must aid towards rational workers replying and also replying truthfully despite malicious workers and the unreliability of the network.

2.3 Combinatorial Agencies

Distributed computation in presence of selfishness was studied within the scope of combinatorial agencies in Economics [7–9, 19]. The basic model considered is a combinatorial variant of the classical principal-agent problem [51]: A master (principal) must motivate a collection of workers (agents) to exert costly effort on the master's 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 [7], 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 [9] 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 [7]). The work in [19] 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 by auditing.

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 under an unreliable network (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 worker's 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 worker's 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).

2.4 Other Related Work

The use of game theory in distributed systems is rapidly expanding in an effort to give solutions to traditional distributed computing problems. Prior examples of game theory in distributed computing include work on Internet routing [30, 45, 52, 61], resource/facility location and sharing [29, 32], containment of viruses spreading [56], secret sharing [2, 34], P2P services [3, 48, 49] and task computations [26, 71]. For more discussion on the connection between game theory and distributed computing we refer the reader to the surveys by Halpern [33] and by Abraham, Alvisi and Halpern [1], and the book by Nisan et al [59].

Eliaz [22] 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.

Another interesting work is the one by Karakostas and Viglas [40] that consider the presence of selfish players and only one malicious player in a routing application setting. The malicious player uses its flow through the network in an effort to cause the maximum possible damage. The impact of such malicious behavior is evaluated in the article. In our work through designing a mechanism that achieves reliability with a high probability the impact of malicious behavior can be inferred and is also illustrated through plots.

Gairing [30] introduced and studied *malicious Bayesian congestion games*. These games extend congestion games [62] 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 [56] to measure the influence of malicious behavior for a virus inoculation game involving both rational (selfish) and malicious nodes. Also in the work of Babaioff et al. [10] the Price of Malice

is studied on congestion games and it is measured directly, by comparing the outcome of games with only rational players to the outcome of games with both rational and malicious players.

In an article by Alon et al. [4] the notion of Bayesian ignorance is presented. Bayesian ignorance is quantified by comparing the social cost obtained by players that have local views in a Bayesian game to the expected social cost of players with global views. The authors assume the existence of both altruistic and rational players and present their derived results on a specific congestion game. The main result reached is that having rational agents bear a local view is best for the social cost. Relating to our model all workers and the master have the same view of the system, having workers with different views is something that can not be applied to our framework.

Besides investigating the co-existence of malicious and rational workers, also the co-existence of altruistic and rational workers has been studied. Meier et al. [53] study the virus inoculation game on a social graph and describe the degree of friendship by a factor F. This factor describes how must players care about their adjacent players in a social network, by $F = 1$ a player values the welfare of it's neighbor the same as it's own. In our model altruistic players can be considered as valuing only the welfare of the system. Also Hoefer and Skopalik [37] study congestion games with altruists, assuming a level of altruism for each player, $\beta_i = 0$ being a pure selfish and $\beta_i = 1$ being a pure altruist. Again in our work in that sense we consider only pure altruists and pure selfish players.

In the work of Kuznetsov and Schmid [47] the notion of a *social range matrix* and is effects on the equilibria in a network game are presented. This social range matrix describes arbitrary social relationships between players, how much a player cares about every other player. Malicious and altruistic behaviors of a player towards the other players are modeled, combining the malicious and altruistic behavior discussed above. In our line of work, workers have only social relationships with the master and not with each other. Also we assume that a worker, based on the literature's

phraseology, can only be pure selfish, pure altruistic or malicious towards the master. Considering different degrees of rationality or altruism will make our analysis loose it's generality and also having the percentage of different degrees of a behavior is not trivial.

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. (In this respect, one might say that our algorithmic mechanisms designed in this work, dealing also with the networks unreliability, 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 [49] 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 [48] 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* [13], 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.

Monderer and Tennenholtz [55] assume a reliable party that can not modify game rules, for example provide protocols, but wishes to influence the behavior of the players in a game. They introduce the concept of k-implementation, as a way to influence the outcome of the game and hence complement previous works in mechanism design. A k actual monetary payment is given

to provide the desirable outcomes; if k is large enough any specific outcome is possible. A player following a desired behavior will not necessarily take the monetary payment. k-implementation is addressed in the concept of games with complete and incomplete informations and pure and mixed strategies.

We provide a protocol that defines the different cases under which different payments/punishments are given to the workers. In our work, payments also will not necessarily be given when the worker follows a desired strategy. Our model's primal goal though, is for the master to get the correct reply thus rewarding less workers is a second order priority. Having a set of desirable outcomes, instead of forcing a unique equilibrium, it is possible to minimize the decrease the master's cost. But having to deal with many equilibria and at the same time rigorously analyzing the possibilities seems unmanageable.

A somewhat related work is [18] in which they face the problem of bootstrapping a P2P computing system, in the presence of rational peers. The goal is to incentive peers to join the system, for which they propose a scheme that mixes lottery psychology and multilevel marketing. In our setting, the master could use their scheme to recruit workers. We assume in this work that enough workers are willing to participate in the computation.

Chapter 3

Model and Definitions

In this section we present our model and the conventions that comes with it and give the parameter definitions followed in the rest of this work.

3.1 Master-Workers Framework and Worker Types

This work extends the framework presented in [27] to include the unreliability of the network and the rational workers strategic choice not to reply. 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 [67], 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 type of any worker w is known only by w. That is, neither the master nor the other workers know

the type of worker w. Furthermore, the number of workers of each type is unknown to everyone. With respect to the worker types, the only knowledge available is a probability distribution over those types. Specifically, it is known that each worker is independently of one of the three types with probabilities p_ρ, p_μ, p_α, respectively, where $p_\rho + p_\mu + p_\alpha = 1$. The knowledge of the distribution over types could be obtained, for example, statistically from existing master-worker applications. If such information is inaccurate it is enough to overestimate $p_\mu$and underestimate p_α (as we do in Section 4.3.1 from SETI-like systems [15,18]) to achieve correctness, although at a bigger expense. Malicious and altruistic workers always cheat and are honest, respectively, independently of how such a behavior impacts their utilities. In the context of this work, 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.

The above implies that all rational workers share the same probability distribution over the possible strategies (cheat, be honest, abstain), i.e., all rational workers are of the same type. Otherwise, in order to model the individuality of the non-monetary part of each rational worker's 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.

3.2 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. [36]. 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 Chapter 4, 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 sends a task assigned and receives the reply from a given worker; that is, d represents the network's reliability. 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. $d > 0$.

3.3 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 [63], [25], [26] and [27], 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 (as in [27]), 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.

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

$\mathcal{R}_{\mathrm{m}}$	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

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. In comparison with the model presented in [27], the model where the master rewards the majority and punishes the minority is not used, for the reason mentioned above.

The payoff parameters considered in this work are detailed in Table 2. 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 a "sufficient" number of replies
$MB_{\mathcal{R}}$	master's benefit from accepting the right answer

Table 2: Payoffs. All parameters are non-negative.

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. For

example, in scenarios such as SETI, workers carry out the computation for free. Nevertheless, the master may still incur in some costs for processing the replies, posting a list of participants, etc. Although workers are not penalized for not replying, our model allows the possibility for the master to be penalized for not getting enough replies (parameter $MC_{\mathcal{S}}$) (the actual number of "enough" replies is quantified in Section 3). 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. As usual in algorithmic mechanism design, we include a punishment in addition to the incentive. This is an implementation of a "carrot and stick" incentive-based mechanism when dealing with rational workers. Such mechanism is possible when the workers actions are contractible and verifiable as in our model (unlike the case of combinatorial agencies). Nevertheless, observe that, if needed, the punishment may be disabled setting $WP_{\mathcal{C}} = 0$ (as some instances here). 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.

3.4 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* [35].

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 network's 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 [60], 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,

$$U_i(s_i, \sigma_{-i}) = U_i(s_i', \sigma_{-i}), \forall s_i, s_i' \in supp(\sigma_i),$$

$$U_i(s_i, \sigma_{-i}) \geq U_i(s_i', \sigma_{-i}),$$

$$\forall s_i, s_i' : s_i \in supp(\sigma_i), s_i' \notin supp(\sigma_i),$$

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.

The above definition applies to our setting as follows. First notice that there is no NE where some players choose a pure strategy and others do not, because the game is symmetric for all rational players. (Should many types of rational players be considered, then we would have to consider such a NE.) Assume first that there is a NE with mixed-strategies (that is, a NE where no strategy is chosen with probability 1). Then, the expected utility of a worker is the same for each pure strategy that such worker can choose with positive probability, and it is not less than the expected utility of a pure strategy with probability zero of being chosen (if there is any). On the other hand, if only pure strategies are included in a NE (that is, there is only one strategy that can be chosen), that means that the expected utility of a worker is not less than the expected utility on the remaining pure strategies. Let us illustrate with an example. If $p_{\mathcal{C}} = 1/2, p_{\mathcal{H}} = 1/2, p_{\mathcal{N}} = 0$ is a NE, that means that the expected utility of a worker is the same if it cheats or is honest, and it is not less than the utility if it does not reply. On the other hand, if $p_{\mathcal{C}} = 0, p_{\mathcal{H}} = 1, p_{\mathcal{N}} = 0$ is a NE, then the expected utility of an honest worker is not less than the expected utility of cheating or not replying.

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 4. If we show conditions such that $\Delta U_{\mathcal{HC}} = 0$ and $\Delta U_{\mathcal{HN}} = 0$, 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

$W = \{1, 2, \ldots, n\}$	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

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 this work is summarized in Table 3.

Chapter 4

Algorithmic Mechanisms

In this section we present the mechanisms we design and analyze them. 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.

4.1 Algorithms

As discussed in Section 3.2, 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

```
1  send(task, p_A, certificate) to n workers
2  wait time T for replies
3  upon expire of time T do
4  audit the answers with probability p_A
5  if the answers were not audited then
6      accept the majority
7  end if
8  apply the reward model
```

Figure 1: Master Algorithm for the Time-based Mechanism

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 $MC_{\mathcal{S}}$.

```
1  send(task, p_A, certificate) to n workers
2  if at least k replies are received then
3      audit the answers with probability p_A
4      if the answers were not audited then
5          accept the majority
6      end if
7      apply the reward model
8  end if
```

Figure 2: Master Algorithm for the Reply-based Mechanism

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. [41] that have demonstrated

experimentally that there are common tasks that 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 3.4. 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., [7, 58]. (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.) As we show in the proof of Theorem 6, if multiple NE were allowed, choosing deterministically to cheat would be also an equilibrium strategy. Thus, for the purpose of a worst-case analysis with respect to the probability of correctness, it would have to be assumed that rational players choose to cheat, yielding the presence of rationals irrelevant. The reason to aim for a NE at all is that, although it is known that equilibria do not always yield optimal solutions, it is a "safe" way for the rational players to obtain high utility satisfaction [59, Chapter 1]. More importantly, a NE is *stable*, that is, once proposed, it is against the interest of the players to individually deviate.

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 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 4.4.

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.

if *Pr*[*majority honest* | *all rationals honest*] $< 1 - \varepsilon$ **then** /* P_{succ} is small, even if $p_{\mathcal{H}} = 1$ */
  $p_{\mathcal{C}} \leftarrow 1$; $p_{\mathcal{N}} \leftarrow 0$; $p_{\mathcal{A}} \leftarrow 1 - \varepsilon \big/ \sum_{i=k}^{n} r_i c_i$; /* 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**
  /* $p_{\mathcal{H}} = 1$, even if $p_{\mathcal{A}} = 0$ */
    $\Delta U_{\mathcal{HC}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **and** $\Delta U_{\mathcal{HN}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **then**
  $p_{\mathcal{C}} \leftarrow 0$; $p_{\mathcal{N}} \leftarrow 0$; $p_{\mathcal{A}} \leftarrow 0$; /* 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 $U_M\left(p_{\mathcal{A}}, p_{\mathcal{N}}, p_{\mathcal{C}}\right) < U_M\left(p_{\mathcal{A}} = (1-\varepsilon) \big/ \sum_{i=k}^{n} r_i, p_{\mathcal{N}} = 1, p_{\mathcal{C}} = 0\right)$ **then**
  $p_{\mathcal{N}} \leftarrow 1$; $p_{\mathcal{A}} \leftarrow (1-\varepsilon) \big/ \sum_{i=k}^{n} r_i$; /* 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 4.2

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 $p_{\mathcal{A}}$ so 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 3.2, 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 ($d = 1$), all replies will be received within a certain time frame. Thus the assumptions made by the two algorithms designed to provide termination in the presence of an unreliable network, seize to exist. The master's algorithms will fall into an algorithm where upon receiving all replies the master will decide whether to audit or not the received replies, in an abstract sense they become the same as the algorithm presented in the work of Fernández et al. [27]. 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.

4.2 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., $\zeta = 1/n$).

4.2.1 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,

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

4.2.2 General Equilibria Conditions

Recall from Section 3.4 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.

4.2.3 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 5.

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 4.1.

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 $p_{\mathcal{A}} = (1 - \varepsilon) \big/ \sum_{i=k}^{n} r_i$, 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,

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

Proof. Conditioning Equation 2 to be $\geq 1-\varepsilon$, $p_\mathcal{A} \geq 1-\varepsilon \big/ \sum_{i=k}^{n} r_i c_i$. 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

$$r'_i = \tbinom{n-1}{i} r^i \overline{r}^{n-1-i},$$

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

$$c'_i = \textstyle\sum_{j=\lceil i/2 \rceil}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil i/2 \rceil - \lfloor i/2 \rfloor\right) \frac{1}{2} \binom{i}{\lfloor i/2 \rfloor} q^{\lfloor i/2 \rfloor} \overline{q}^{\lceil i/2 \rceil},$$

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. □

4.3 Correctness and Optimality

The following theorem proves the correctness of the mechanisms presented in Section 4.1. Its proof is the simple aggregation of the results presented in Section 4.2.

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. □

4.4 Computational Issues

In Sections 4.1 and 4.2.3 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 [39] 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 4.2.3).

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

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

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

Chapter 5

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.

5.1 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.

5.2 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 [5]. 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.

5.3 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 (complementing the work of Fernández et al. [27]), both in the SETI-like and the Contractor settings.

5.3.1 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 receive all replies (cf., Section 4.1). Recall that the only knowledge available about the workers type is a probability distribution. Such knowledge could be obtained statistically from existing master-worker applications

such as [20, 23]. To err on the safe side, we overestimate p_μ and underestimate p_α with respect to those statistics.

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 [20] and [23], 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.

(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 5.

(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 6.

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 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 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.

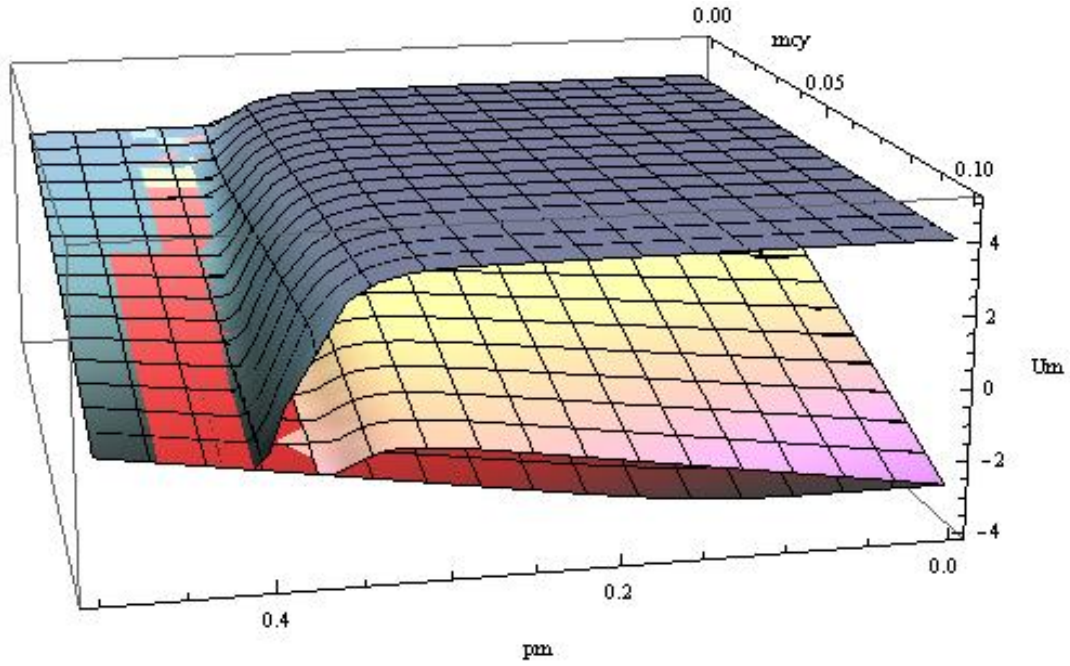


Figure 4: Time-based Mechanism in the SETI-like scenario: Master's utility for the three plot scenarios: The upper plane corresponds to $\mathcal{R}_\emptyset$, the middle to $\mathcal{R}_\mathrm{m}$, and the third to $\mathcal{R}_\mathrm{a}$.

In Figure 5 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.

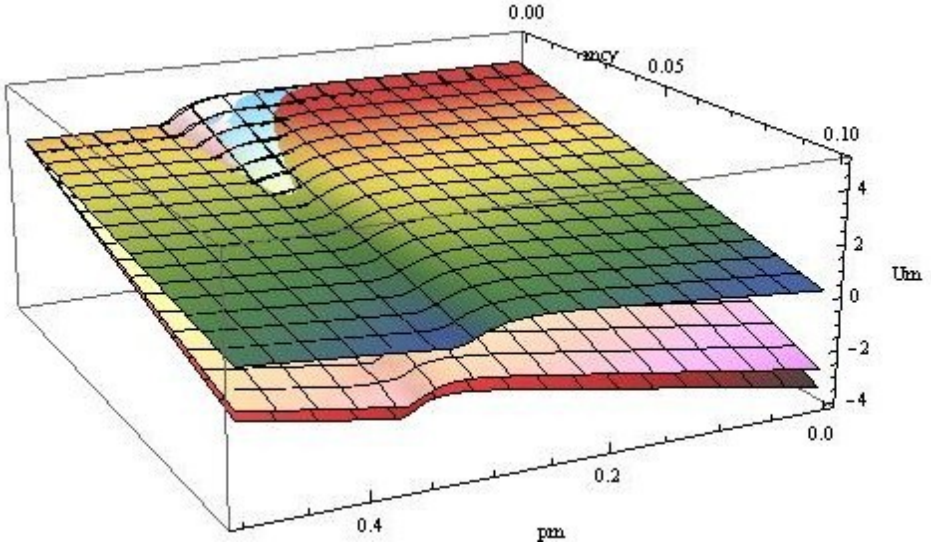


Figure 5: Time-based Mechanism in the SETI-like scenario: Master's utility for the three plot scenarios: The upper plane corresponds to $d = 0.5$, the middle to $d = 0.9$, and the third to $d = 0.99$.

In Figure 6 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

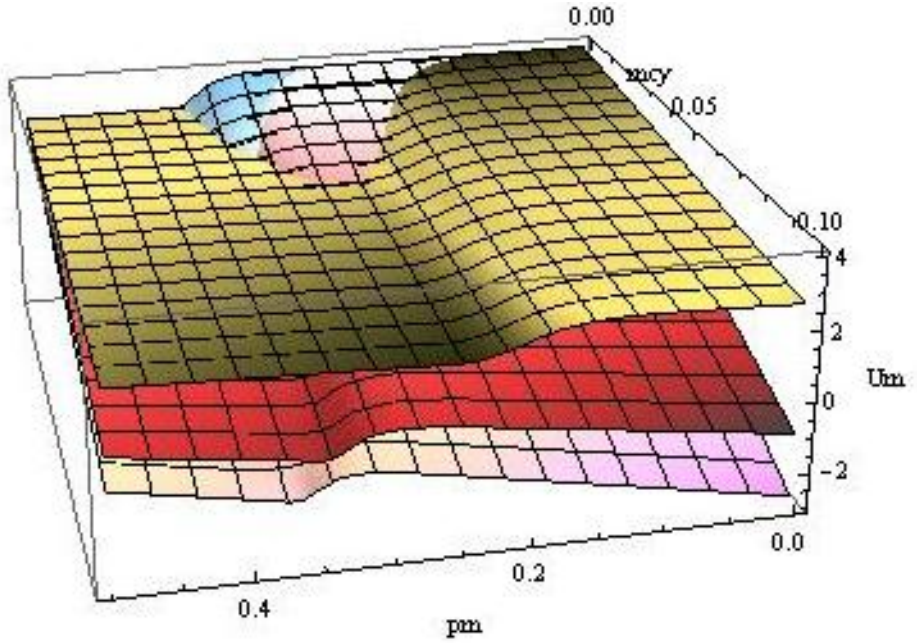


Figure 6: Time-based Mechanism in the SETI-like scenario: Master's utility for the three plot scenarios: The upper plane corresponds to $n = 15$, the middle to $n = 55$, and the third to $n = 75$.

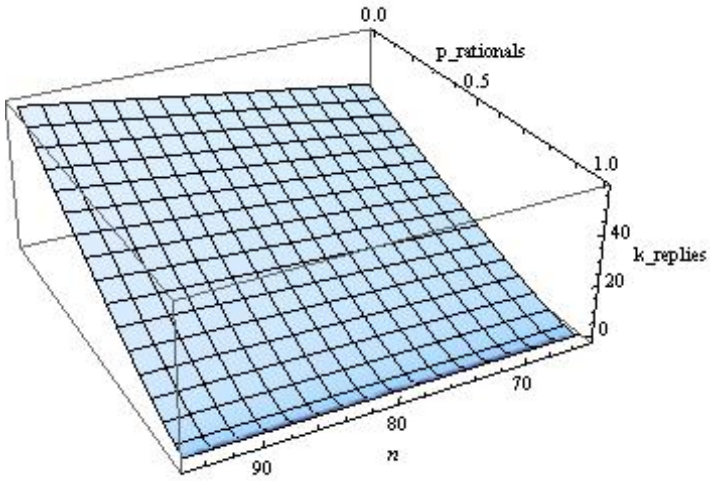
workers selected by the master n, and by the probability distribution of rational workers p_ρ. Furthermore, we depict how k is affecting the utility of the master. As with the previous mechanism, we set $MC_{\mathcal{A}} = 1$, $MP_{\mathcal{W}} = 100$, $MC_{\mathcal{S}} = 10$ and $MB_{\mathcal{R}} = 4$.

We consider two plot scenarios:

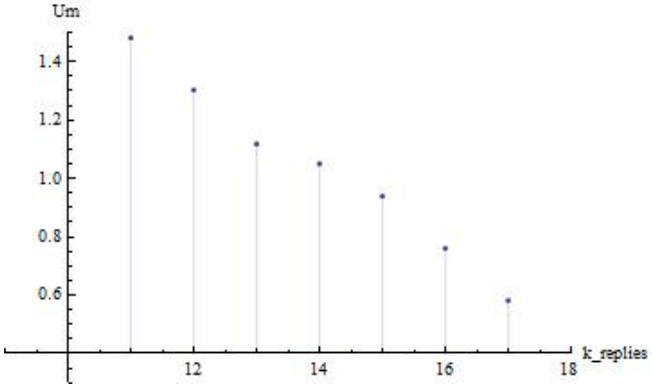
(a) We vary n from 65 to 95, p_ρ for 0 to 1, and we compute the appropriate k that the master should choose for each n. The results are depicted in Figure 7(a).

(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 7(b).

In Figure 7(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.



(a)



(b)

Figure 7: Plots of the SETI-like Scenario for the Reply-based Mechanism

In Figure 7(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:

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; for both time-based and reply-based mechanisms the master receives all replies from the workers. Hence the two mechanisms are essentially becoming the same mechanism. 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 8(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 8(b).

(c) We fix n=75 for both values of $MB_{\mathcal{R}}$ mentioned earlier; in Figure 8(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 6).

A natural and expected observation in Figure 8, 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 8, 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 8 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 8(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.

5.3.2 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 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 9 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 9(a) we fix n=7, in Figure 9(b) we fix n=15

and in Figure 9(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).

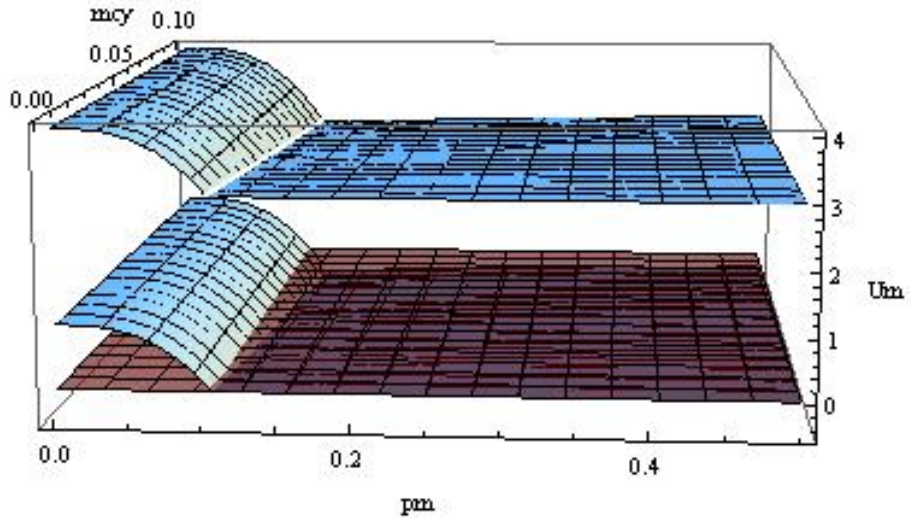
Observe that a threshold point exists where the master changes its strategy from auditing with some probability (that guaranties the utility of the rational workers is positive) to auditing. We generally observe that (not surprisingly) for values of p_μ and $WC_{\mathcal{T}}$ close to zero we get the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

In all plot in Figure 9 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.

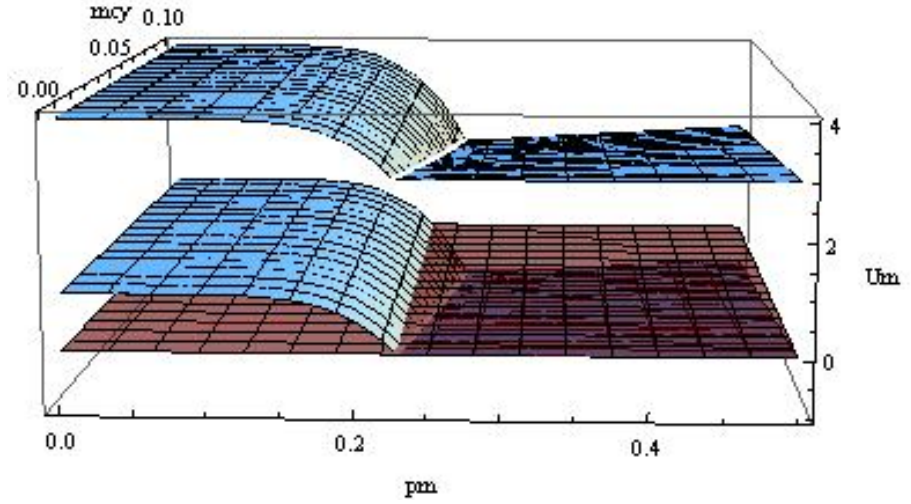
Another observation (especially in Figure 9(c)), is that before the threshold value, as p_μ increases, the utility of the master increases, and then decreases for every value of $WC_{\mathcal{T}}$ (except when close to $WC_{\mathcal{T}} = 0$ and $WC_{\mathcal{T}} = S$)! When p_μ is increasing, the number 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,

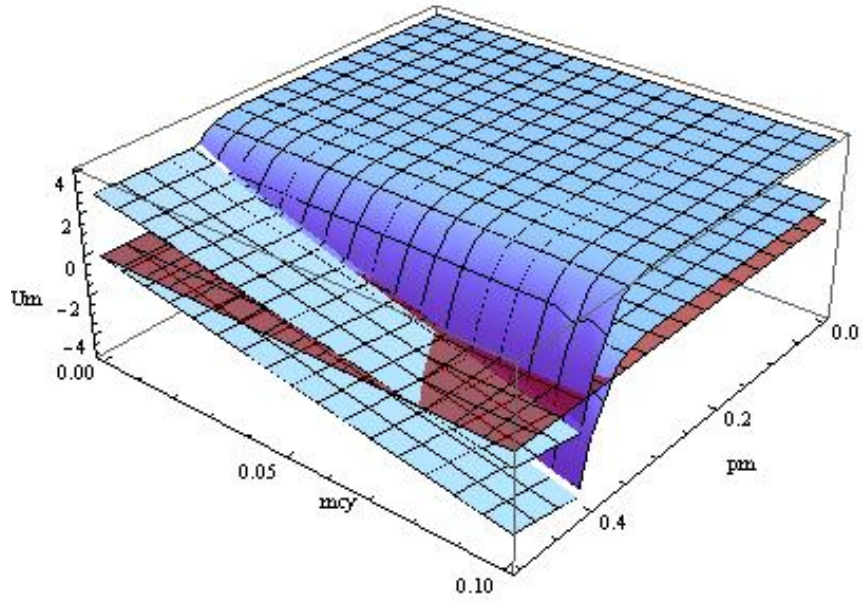
observe the big decrease in the master's utility as the number of workers grows. This is due to the large payments that the master has to give to large groups of workers to guarantee reliability.



(a)

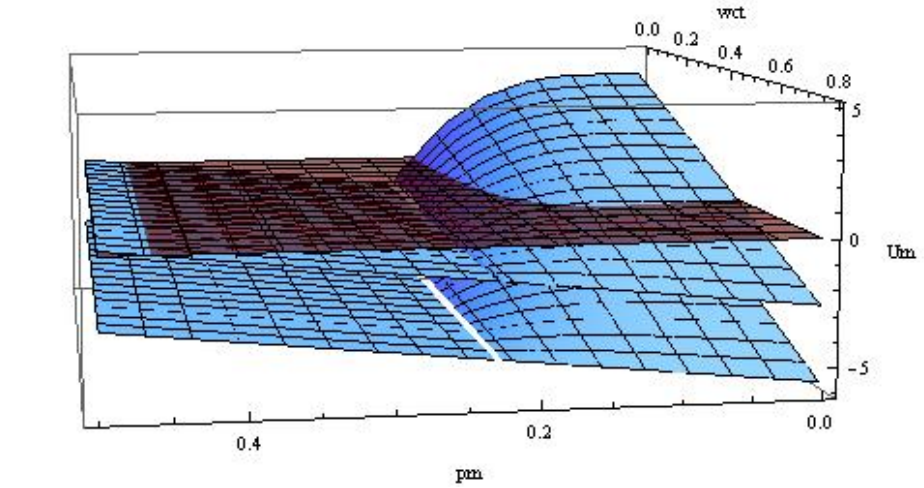


(b)

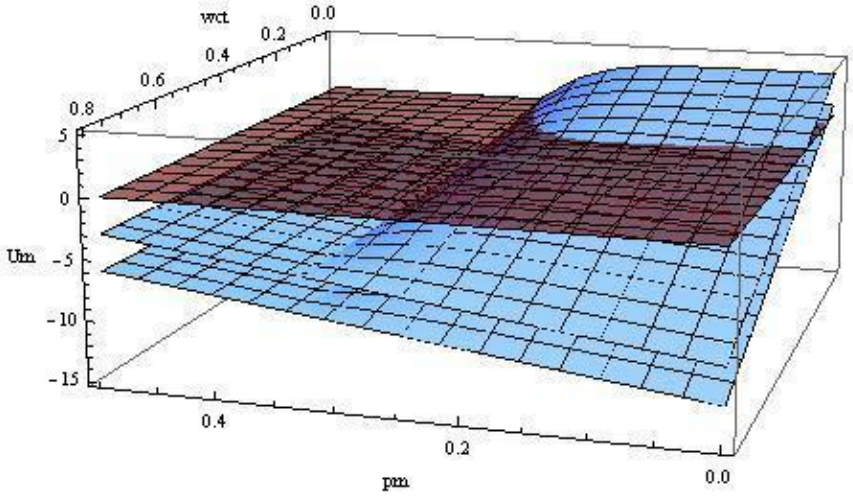


(c)

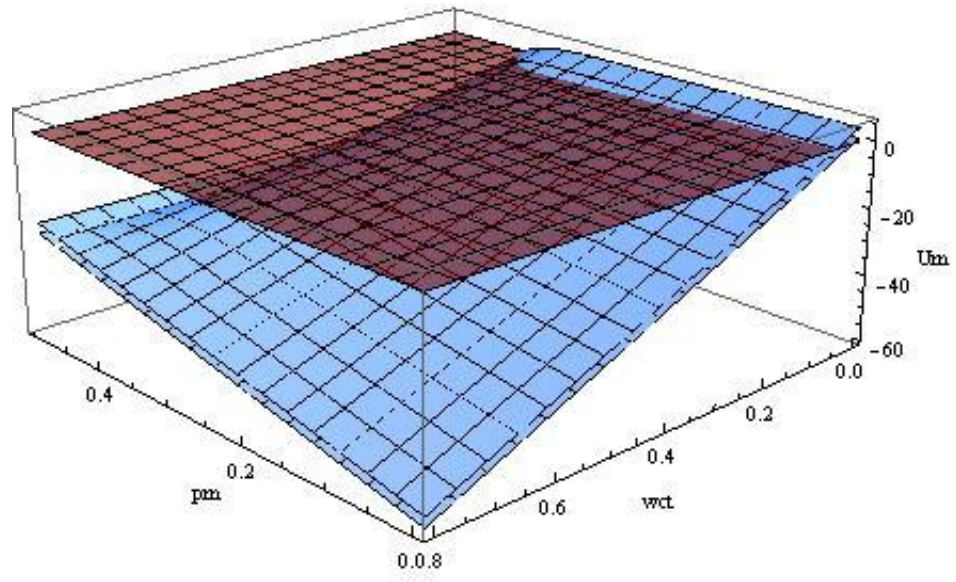
Figure 8: Plots of the SETI-like scenario for $d = 1$. The upper plane corresponds to $MB_{\mathcal{R}} = 4$ the lower plane to $MB_{\mathcal{R}} = 1$ and the red flat plane to $U_M = 0$. (a) $n = 5$. (b) $n = 15$. (c) $n = 75$.



(a)



(b)



(c)

Figure 9: 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) $n = 75$.

Chapter 6

Discussion and Future Work

In this work we have expanded the classical distributed computing approach (voting) with the game-theoretic one (cost-based incentives and payoffs) that is used in [27], by considering the unreliability of the network and workers abstaining the computation. This has led 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 (able to abstain) workers, and the underlying network's unreliability. In the case of a reliable communication the two mechanisms basically are becoming one and in fact we are led back to the mechanism proposed in [27]. Hence our the two mechanisms are purely more general.

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 [16].

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 [38,69]. The master could use the knowledge gained in the previous rounds to increase its utility and its probability of success in future rounds. The workers could benefit from the constant interaction with the master and reinforce their strategy with the goal of increasing their utility. This is an evolutionary process of learning that could be modeled with reinforcement learning [12, 65]. Reinforcement learning could help us model how system entities interact with their environment to decide upon a strategy, and use their experience to select or avoid actions according to the observed consequences. Also with reinforcement learning the only information that players need is the payoffs they receive. Issues such as worker *aspiration level* [11], the benefit that the worker wishes to receive from the computation, could be taken into account.

Considering a repeated interaction with the same workers over the course of multiple task assignments, could betray information about the workers behavior. Over the course of time the master could gather information about the workers using a reputation mechanism. Using reputation as an additional mechanism to enforce the good behavior of workers we could consider several implementation approaches. Choosing a number of workers from a pool according to their reputation, considering workers are aware that a reputation mechanism is used against them (something that would change the equilibrium dynamics), etc.

Bibliography

[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] N. Alon, Y. Emek, M. Feldman and M. Tennenholtz. Bayesian Ignorance In proc. ACM Symposium on Principles of Distributed Computing , 2010.

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

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

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

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

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

[10] M. Babaioff, R. Kleinberg and C. Papadimitriou. Congestion Games with Malicious Players. *In proc. of ACM Conference on Electronic Commerce (EC*, San Diego, CA, USA,2007.

[11] 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.

[12] R. R. Bush and F. Mosteller. *Stochastic Models for Learning*, Wiley, 1955.

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

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

[15] E. Christoforou. Studding Mechanisms for a Reliable Master-Worker Computation. Bachelor Thesis submitted at the Department of Computer Science, University of Cyprus, 2009.

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

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

[18] J. R. Douceur and T. Moscibroda. Lottery trees: motivational deployment of networked systems In *proc. of SIGCOMM 2007*, pp. 121132, 2007.-.1em

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

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

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

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

[23] 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.

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

[25] A. Fernández Anta, Ch. Georgiou, L. Lopez, and A. Santos. Reliably executing tasks in the presence of untrusted processors. In *proc. of PPL March 2012*, vol 2.

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

[27] A. Fernández Anta, Ch. Georgiou, and M. A. Mosteiro. Algorithmic Mechanisms for Internet-based Master-Worker Computing with Untrusted and Selfish Workers. In *proc. of IPDPS 2010*, PP.378-388.

[28] 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.

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

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

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

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

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

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

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

[36] 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.

[37] M. Hoefer and A. Skopalik. Altruism in Atomic Congestion Games. *In proc. of the European Symposium on Algorithms (ESA*, 2009.

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

[39] 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.

[40] G. Karakostas and A. Viglas. Equilibria for Networks with Malicious Users. *Mathematical Programming*, 110(3), 591613, 2007.

[41] 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.

[42] 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.

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

[44] 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.

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

[46] M. Kuhn, S. Schmid and R. Wattenhofer. Distributed asymmetric verification in computational grids. *In proc. 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 1-10, 2008.

[47] P. Kuznetsov and S. Schmid. Towards Network Games with Social Preferences. *Structural Information and Communication Complexity*, pages 1428, 2010.

[48] 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.

[49] 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.

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

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

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

[53] D. Meier, Y. A. Oswald, S. Schmid and R. Wattenhofer. On the Windfall of Friendship: Inoculation Strategies on Social Networks. *In proc. 9th ACM Conference on Electronic Commerce (EC)* , 2008.

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

[55] D. Monderer and M. Tennenholtz. k-Implementation. *In proc. of the 4th ACM conference on Electronic commerce*, 19-28, 2003.

[56] 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.

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

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

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

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

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

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

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

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

[65] C. Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool publishers, 2010.

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

[67] 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*.

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

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

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

[71] 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.