Euro-Par 2012, LNCS 7484 (2012) pp. 451–463

DOI:10.1007/978-3-642-32820-6_45

Achieving Reliability in Master-Worker Computing via Evolutionary Dynamics

Evgenia Christoforou[1], Antonio Fernández Anta[2], Chryssis Georgiou[1], Miguel A. Mosteiro[3], and Angel (Anxo) Sánchez[4]

[1] University of Cyprus, Nicosia, Cyprus
[2] Institute IMDEA Network & Univ. Rey Juan Carlos, Madrid, Spain
[3] Rutgers University, Piscataway, NJ, USA & Univ. Rey Juan Carlos, Madrid, Spain
[4] Universidad Carlos III de Madrid, Madrid, Spain & BIFI Institute, Zaragoza, Spain

Abstract. This work considers Internet-based task computations in which a master process assigns tasks, over the Internet, to rational workers and collect their responses. The objective is for the master to obtain the correct task outcomes. For this purpose we formulate and study the dynamics of evolution of Internet-based master-worker computations through reinforcement learning.

1 Introduction

Motivation: As an alternative to expensive supercomputing parallel machines, Internet is a feasible computational platform for processing complex computational jobs. Several Internet-based applications operate on top of this global computation infrastructure. Examples are volunteer-based "@home" projects [2] such as SETI and profit-seeking computation platforms such as Amazon's Mechanical Turk.

Although the potential is great, the use of Internet-based computing is limited by the untrustworthy nature of the platform's components [2]. In SETI, for example, there is a machine, call it the *master*, that sends tasks, across the Internet, to volunteers' computers, call them *workers*, that execute and report back some result. However, these workers may not be trustworthy and it might be at their best interest to report incorrect results; that is, workers, or their owners, can be viewed as *rational* [1, 14]. 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 (i.e., redundant task allocation is employed [2]).

Prior work [8, 9, 18] has shown that it is possible to design algorithmic mechanisms with reward/punish schemes so that the master can reliably obtain correct task results. We view these mechanisms as one-shot in the following sense: In a round, the master sends a task to be computed to a collection of workers, and the mechanism, using auditing and reward/punish schemes guarantees (with high probability) that the master gets the correct task result. For another task to be computed, the process is repeated (with the same or different collection of workers) but without taking advantage of the knowledge gained.

Given a long running computation (such as SETI-like master-worker computations), it can be the case that the best interests, and hence the behavior of the workers, might change over time. So, one wonders: Would it be possible to design a mechanism for performing many tasks, over the course of a possibly infinite computation, that could positively exploit the repeated interaction between a master and the same collection of workers?

Our Approach: In this work we provide a positive answer to the above question. To do so, we introduce the concept of *evolutionary dynamics* under the biological and social perspective and relate them to Internet-based master-worker task computing. More specifically, we employ *reinforcement learning* [4, 15] to model how system entities or learners interact with the environment to decide upon a strategy, and use their experience to select or avoid actions according to the consequences observed. Positive payoffs increase the probability of the strategy just chosen, and negative payoffs reduce this probability. Payoffs are seen as parameterizations of players' responses to their experiences. Empirical evidence [3, 5] suggests that reinforcement learning is more plausible with players that have information only on the payoffs they receive; they do not have knowledge of the strategies involved. This model of learning fits nicely to our master-worker computation problem: the workers have no information about the master and the other workers' strategies and they don't know the set of strategies that led to the payoff they receive. The workers have only information about the strategies they choose at each round of the evolution of the system and their own received payoffs. The master also has minimal information about the workers and their intentions (to be truthful or not). Thus, we employ reinforcement learning for both the master and the workers in an attempt to build a reliable computational platform.

Our Contributions

1. We *initiate* the study of the evolutionary dynamics of Internet-based master-worker computations through reinforcement learning.
2. We develop and analyze a mechanism based on reinforcement learning to be used by the master and the workers. In particular, in each round, the master allocates a task to the workers and decides whether to audit or not their responses with a certain probability $p_{\mathcal{A}}$. Depending on whether it audits or not, it applies a different reward/punish scheme, and adjusts the probability $p_{\mathcal{A}}$ for the next round (a.k.a. the next task execution). Similarly, in a round, each worker i decides whether it will truthfully compute and report the correct task result, or it will report an incorrect result, with a certain probability p_{Ci}. Depending on the success or not of its decision, measured by the increase or the decrease of the worker's *utility*, the worker adjusts probability p_{Ci} for the next round.
3. We show necessary and sufficient conditions under which the mechanism ensures *eventual correctness*, that is, we show the conditions under which, after some finite number of rounds, the master obtains the correct task result in every round, with minimal auditing, while keeping the workers satisfied (w.r.t. their utility). Eventual correctness can be viewed as a form of *Evolutionary*

Stable Strategy [6, 10] as studied in Evolutionary Game Theory: even if a "mutant" worker decides to change its strategy to cheating, it will soon be brought back to an honest strategy.

4. Finally, we show that our mechanism, when adhering to the above-mentioned conditions, reaches eventual correctness quickly. In particular, we show analytically, probabilistic bounds on the convergence time, as well as bounds on the expected convergence time. Our analysis is complemented with simulations.

Background and Related Work: Evolutionary dynamics were first studied in evolutionary biology, as a tool to studying the mathematical principles according to which life is evolving. Many fields were inspired by the principles of evolutionary dynamics; our work is inspired by the dynamics of evolution as a mean to model workers' adaptation to a truthful behavior.

The dynamics of evolution have mainly been studied under the principles of Evolutionary Game Theory (EGT) [11]. In EGT the concept of evolutionarily stable strategy (ESS) is used [6, 10]. A strategy is called evolutionary stable if, when the whole population is using this strategy, any group of invaders (mutants) using a different strategy will eventually die off over multiple generations (evolutionary rounds). It is shown [10] that an ESS is a Nash Equilibrium, but the reverse is not true.

While evolution operates on the global distribution of strategies within a given population, reinforcement learning [15] operates on the individual level of distribution over strategies of each member of the population. There are several models of reinforcement learning. A well-known model is the Bush and Mosteller's model [4]. This is an aspiration-based reinforcement learning model where negative effects on the probability distribution over strategies are possible, and learning does not fade with time. The player's adapt by comparing their experience with an *aspiration* level. In our work we adapt this reinforcement learning model and we consider a simple aspiration scheme where aspiration is fixed by the workers and does not change during the evolutionary process.

Phelps, McBurney and Parsons [13] discusses the concept of Evolutionary Mechanism Design. The evolutionary mechanism has a continues interaction and feedback from the current mechanism, as opposed to classical mechanism design [12] than when the mechanism is introduced in the system, it remains in the same Nash Equilibrium forever. In some way, our mechanism can be seen as an evolutionary mechanism, since the probability of auditing of the master and the probability of cheating of the workers, change, which is similar to changing the mechanism.

An extended account on related work (discussing applications of game theory to distributed computing, the concept of combinatorial agencies, the BAR model, etc.) can be found in [17].

2 Model and Definitions

Master-Worker Framework: We consider a distributed system consisting of a master processor that assigns, over the Internet, computational tasks to a set

of n workers (w.l.o.g., we assume that n is odd). In particular, the computation is broken into rounds, and in each round the master sends a task to the workers to compute and return the task result. The master, based on the workers' replies, must decide on the value it believes is the correct outcome of the task in the same round. The tasks considered in this work are assumed to have a unique solution; although such limitation reduces the scope of application of the presented mechanism [16], there are plenty of computations where the correct solution is unique: e.g., any mathematical function.

Following Abraham et al. [1], and Shneidman and Parkes [14], we assume that workers are *rational*, that is, they are selfish in a game-theoretic sense and their aim is to maximize their benefit (utility) under the assumption that other workers do the same. In the context of this paper, a worker is *honest* in a round, when it truthfully computes and returns the task result, and it *cheats* when it returns some incorrect value. So, a worker decides to be honest or cheat depending on which strategy maximizes its utility. We denote by p^r_{Ci} the probability of a worker i cheating in round r. This probability is not fixed, the worker adjusts it over the course of the computation.

While it is assumed that workers make their decision individually and with no coordination, it is assumed that all the workers that cheat in a round return the same incorrect value (as done, for example, in [7] and [8]). This yields a worst case scenario (and hence analysis) for the master with respect to obtaining the correct result using mechanisms where the result is the outcome of voting; 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.)

Auditing, Payoffs, Rewards and Aspiration: To "persuade" workers to be honest, the master employs, when necessary, *auditing* and *reward/punish* schemes. The master, in a round, might decide to audit the response of the workers (at a cost). In this work, auditing means that the master computes the task by itself, and checks which workers have been honest. We denote by $p_{\mathcal{A}}$ the probability of the master auditing the responses of the workers. The master can change this auditing probability over the course of the computation. However, unless otherwise stated, we assume that there is a value $p^{min}_{\mathcal{A}} > 0$ so that at all times $p_{\mathcal{A}} \geq p^{min}_{\mathcal{A}}$.

Furthermore, the master can reward and punish workers, which can be used (possibly combined with auditing) to encourage workers to be honest. 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 it rewards only the majority. We refer to this as the $\mathcal{R}_{\text{m}}$ reward scheme.

The payoff parameters considered in this work are detailed in Table 1. 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. 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.

Table 1. Payoffs. The parameters are non-negative.

$WP_{\mathcal{C}}$	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
$MB_{\mathcal{R}}$	master's benefit from accepting the right answer

We assume that, in every round, a worker i has an *aspiration* a_i, that is, the minimum benefit it expects to obtain in a round. In order to motivate the worker to participate in the computation, the master must ensure that $WB_{\mathcal{Y}} \geq a_i$; in other words, the worker has the potential of its aspiration to be covered. We assume that the master knows the aspirations. This information can be included, for example, in a contract the master and the worker agree on, prior to the start of the computation.

Note that, 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 goal of eventual correctness. All other parameters can either be fixed because they are system parameters or may also be chosen by the master (except the aspiration, which is a parameter set by each worker).

Eventual Correctness: The goal of the master is to eventually obtain a reliable computational platform. In other words, after some finite number of rounds, the system must guarantee that the master obtains the correct task results in every round with probability 1. We call such property *eventual correctness*.

3 Algorithmic Mechanism

We now detail the algorithms run by the Master and the workers.

Master's Algorithm: The master's algorithm begins by choosing the initial probability of auditing. After that, at each round, the master sends a task to all workers and, when all answers are received (a reliable network is assumed), the master audits the answers with probability $p_{\mathcal{A}}$. In the case the answers are not audited, the master accepts the value contained in the majority of answers and continues to the next round with the same probability of auditing. In the case the answers are audited, the value $p_{\mathcal{A}}$ of the next round is reinforced (i.e., modified according to the outcome of the round). Then, the master rewards/penalizes the workers appropriately. The master initially has scarce or no information about the environment (e.g., workers initial p_C). The initial probability of auditing will be set according to the information the master possesses. For example if it has no information about the environment, a safe approach may be to initially set $p_{\mathcal{A}} = 0.5$.

Observe that, when the answers are not audited, the master has no information about the number of cheaters in the round. Thus, the probability $p_{\mathcal{A}}$ remains

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $r \leftarrow 1$ **to** ∞ **do**
  send *a task* T *to all workers in* W
  upon *receiving all answers* **do**
    audit the answers with probability $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
      accept the majority
    else
      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(cheaters(r)/n - \tau)$
      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W$: *pay/charge* Π_i*to worker* i

Algorithm 2 Algorithm for Worker i

$p_{Ci} \leftarrow y$, *where* $y \in [0, 1]$
for $r \leftarrow 1$ **to** ∞ **do**
  receive *a task* T *from the master*
  set $S_i \leftarrow -1$ *with probability* p_{Ci}, *and*
    $S_i \leftarrow 1$ *otherwise*
  if $S_i = 1$
    then $\sigma \leftarrow compute(T)$
  else $\sigma \leftarrow$ *arbitrary solution*
  send *response* σ *to the master*
  get payoff Π_i
  $p'_{Ci} \leftarrow p_{Ci} - \alpha_w(\Pi_i - a_i)S_i$
  $p_{Ci} \leftarrow \max\{0, \min\{1, p'_{Ci}\}\}$

the same as in the previous round. When the answers are audited, the master can safely extract the number of cheaters. Then, the master adapts the auditing probability $p_{\mathcal{A}}$ according to this number. (We denote by $cheaters(r)$ the number of cheaters in round r.) Observe that the algorithm guarantees $p_{\mathcal{A}} \geq p_{\mathcal{A}}^{min}$. This, combined with the property $p_{\mathcal{A}}^{min} > 0$ will prevent the system to fall in a permanent set of "bad" states where $p_{\mathcal{A}} = 0$ and $p_C > 0$. A discount factor, which we call *tolerance* and denote by τ, expresses the master's tolerable ratio of cheaters (typically, we will assume $\tau = 1/2$). Hence, if the proportion of cheaters is larger than τ, $p_{\mathcal{A}}$ will be increased, and otherwise, $p_{\mathcal{A}}$ will be decreased. The amount by which $p_{\mathcal{A}}$ changes depends on the difference between these values, modulated by a *learning rate* α_m. This latter value determines to what extent the newly acquired information will override the old information. (For example, if $\alpha_m = 0$ the master will never adjust $p_{\mathcal{A}}$.)

Workers' Algorithm: The workers' algorithm begins with each worker i deciding an initial probability of cheating p_{Ci}. At each round, each worker receives a task from the master and, with probability $1 - p_{Ci}$ calculates the task, and replies to the master with the correct answer. If the worker decides to cheat, it fabricates an answer, and sends the incorrect response to the master. We use a flag S_i to model the decision of a worker i to cheat or not. After receiving its payoff (detailed in the analysis section), each worker i changes its p_{Ci} according to the payoff Π_i received, the chosen strategy S_i, and its aspiration a_i. Observe that the workers' algorithm guarantees $0 \leq p_{Ci} \leq 1$. The workers have a learning rate α_w. We assume that all workers have the same learning rate, that is, they learn in the same manner (see also the discussion in [15]; the learning rate is called step-size there); note that our analysis can be adjusted to accommodate also workers with different learning rates.

4 Analysis

We now analyze the mechanism, which is composed of the Master's and the workers' algorithms presented in the previous section. We first model the evolution of the mechanism as a Markov Chain, and then we prove necessary and sufficient conditions for achieving eventual correctness. Then, we provide analytical evidence that convergence to eventual correctness can be reached rather quickly. Observe in Algorithms 1 and 2 that there are a number of variables that may change in each round. We will denote the value of a variable X after a round r with a superindex r as X^r.

4.1 The Mechanism as a Markov Chain

We analyze the evolution of the master-workers system as a Markov chain. To do so, we first define the set of states and the transition functions:

Let the state of the Markov chain be given by the vector of probabilities $(p_{\mathcal{A}}, p_{C1}, p_{C2}, \ldots, p_{Cn})$. Then, the state after round r is $(p^r_{\mathcal{A}}, p^r_{C1}, p^r_{C2}, \ldots, p^r_{Cn})$. Observe from Algorithms 1 and 2 that any state $(p_{\mathcal{A}}, p_{C1}, p_{C2}, \ldots, p_{Cn})$ in which $p_{\mathcal{A}} \in [p^{min}_{\mathcal{A}}, 1]$, and $p_{Ci} \in [0, 1]$ for each worker i, is a possible initial state of the Markov chain. The workers' decisions, the number of cheaters, and the payoffs in round r are the stochastic outcome of the probabilities used in round r. Then, restricted to $p^r_{\mathcal{A}} \in [p^{min}_{\mathcal{A}}, 1]$ and $p^r_{Ci} \in [0, 1]$, we can describe the transition function of the Markov chain in detail. For each subset of workers $F \subseteq W$, $P(F) = \prod_{j \in F} p^{r-1}_{Cj} \prod_{k \notin F} (1 - p^{r-1}_{Ck})$ is the probability that the set of cheaters is exactly F in round r. Then, we have the following.

- With probability $p^{r-1}_{\mathcal{A}} \cdot P(F)$, the master audits when the set of cheaters is F, and then, (0) the master updates $p_{\mathcal{A}}$ as $p^r_{\mathcal{A}} = p^{r-1}_{\mathcal{A}} + \alpha_m(|F|/n - \tau)$, and (1) each worker $i \in F$ updates p_{Ci} as $p^r_{Ci} = p^{r-1}_{Ci} - \alpha_w(a_i + WP_{\mathcal{C}})$, (2) each worker $i \notin F$ updates p_{Ci} as $p^r_{Ci} = p^{r-1}_{Ci} + \alpha_w(a_i - (WB_{\mathcal{Y}} - WC_{\mathcal{T}}))$.
- With probability $(1 - p^{r-1}_{\mathcal{A}})P(F)$, the master does not audit when F is the set of cheaters. Then, the master does not change $p_{\mathcal{A}}$ and the workers update p_{Ci} as follows. For each $i \in F$, (3) if $|F| > n/2$ then $p^r_{Ci} = p^{r-1}_{Ci} + \alpha_w(WB_{\mathcal{Y}} - a_i)$, (4) if $|F| < n/2$ then $p^r_{Ci} = p^{r-1}_{Ci} - \alpha_w \cdot a_i$, and for each $i \notin F$, (5) if $|F| > n/2$ then $p^r_{Ci} = p^{r-1}_{Ci} + \alpha_w(a_i + WC_{\mathcal{T}})$, (6) if $|F| < n/2$ then $p^r_{Ci} = p^{r-1}_{Ci} + \alpha_w(a_i - (WB_{\mathcal{Y}} - WC_{\mathcal{T}}))$.

The following terminology will be used throughout. Let a *covered worker* be one that is paid at least its aspiration a_i and the computing cost $WC_{\mathcal{T}}$. In any given round r, let an *honest worker* be one for which $p^{r-1}_C = 0$. Let an *honest state* be one where the *majority* of workers are honest. Let an *honest set* be any set of honest states. We refer to the opposite cases as *uncovered worker*, *cheater worker* ($p^{r-1}_C = 1$), *cheat state*, and *cheat set* respectively.

4.2 Conditions for Eventual Correctness

We show the conditions under which the system can guarantee eventual correctness. We begin with some terminology. Let a set of states S be called *closed* if,

once the chain is in any state $s \in S$, it will not move to any state $s' \notin S$. (A singleton closed set is called an *absorbing* state.) For any given set of states S, we say that the chain *reaches* (resp. *leaves*) the set S if the chain reaches some state $s \in S$ (resp. reaches some state $s \notin S$).

In order to show eventual correctness, we must show eventual convergence to a closed honest set. Thus, we need to show (i) that there exists at least one such closed honest set, (ii) that all closed sets are honest, and (iii) that one honest closed set is reachable from any initial state. *Omitted proofs are given in [17].*

Lemma 1. *Consider any set of workers $Z \subseteq W$ such that $\forall i \in Z : WB_{\mathcal{Y}} \geq a_i$. If $|Z| > n/2$, then the set of states*

$$S = \{(p_{\mathcal{A}}, p_{C1}, \ldots, p_{Cn}) | (p_{\mathcal{A}} = 0) \wedge (\forall w \in Z : p_{Cw} = 1)\},$$

is a closed cheat set.

Given (ii) above, the necessity of $p_A^{min} > 0$ is motivated by the above lemma. Hence, $p_A > 0$ is assumed for the rest of the analysis.

Lemma 2. *If there exists a set of workers $Z \subseteq W$ such that $|Z| > n/2$ and $\forall i \in Z : WB_{\mathcal{Y}} < a_i + WC_{\mathcal{T}}$, then no honest set is closed.*

Given (i) above, the necessity of a covered majority is motivated by Lemma 2. Hence, in the remainder we assume that the majority of workers are covered.

Lemma 3. *Consider any set of workers $Z \subseteq W$ such that $\forall i \in Z : WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}$ and $\forall j \notin Z : WB_{\mathcal{Y}} < a_j + WC_{\mathcal{T}}$. If $|Z| > n/2$, then the set of states*

$$S = \{(p_{\mathcal{A}}, p_{C1}, \ldots, p_{Cn}) | \forall w \in Z : p_{Cw} = 0\},$$

is a closed set.

Hence Lemma 3 proves (i) above. We continue with the proof of the other properties.

Lemma 4. *Consider any set of workers $Z \subseteq W$ such that $\forall i \in Z : WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}$ and $\forall j \notin Z : WB_{\mathcal{Y}} < a_j + WC_{\mathcal{T}}$. Then, for any set of states*

$$S = \{\ (p_{\mathcal{A}}, p_{C1}, \ldots, p_{Cn}) | \exists Y \subseteq W : (|Y| > n/2) \wedge (\forall w \in Y : p_{Cw} = 0) \wedge (Z \nsubseteq Y)\},$$

S is not a closed set.

Lemma 5. *Consider any set of workers $Z \subseteq W$ such that $\forall i \in Z : WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}$ and $\forall j \notin Z : WB_{\mathcal{Y}} < a_j + WC_{\mathcal{T}}$. If $|Z| > n/2$ and $p_{\mathcal{A}} > 0$, then for any set of states*

$$S = \{(p_{\mathcal{A}}, p_{C1}, \ldots, p_{Cn}) | \exists Y \subseteq W : (|Y| > n/2) \wedge (\forall w \in Y : p_{Cw} > 0)\},$$

S is not a closed set.

Together, Lemma 4 and 5 prove (ii), and also (iii) because, if only honest sets are closed, then there is a way of going from non-honest sets to one of them. Lemmas 3–5 give the overall result:

Theorem 1. *If $p_{\mathcal{A}} > 0$ then, in order to guarantee with positive probability that, after some finite number of rounds, the system achieves eventual correctness, it is* **necessary** *and* **sufficient** *to set* $\boxed{WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}}$ *for all $i \in Z$ in some set $Z \subseteq W$ such that $|Z| > n/2$.*

The above theorem shows that there is a positive probability of reaching some state after which correctness can be guaranteed, as long as for a chosen majority of workers, the payment is enough to cover their aspiration and cost of performing the task.

Remark: From Algorithm 1 it is easy to see that once the closed set $S = \{(p_{\mathcal{A}}, p_{C1}, \ldots, p_{Cn}) | \forall w \in Z : p_{Cw} = 0\}$ is reached, eventually $p_{\mathcal{A}} = p_{\mathcal{A}}^{min}$ and stays such forever.

4.3 Convergence Time

Theorem 1 shows necessary and sufficient conditions to achieve eventual correctness. However, in order to have a practical system, it is necessary to bound the time taken to achieve it, which we call the *convergence time*. In other words, starting from any initial state, we want to compute the number of rounds that takes to the Markov chain to reach an honest closed set. In this section, we show bounds on the convergence time. *Omitted proofs are given in [17].*

Expected Convergence Time: Let C be the set of all covered workers. We assume, as required by Theorem 1, that $|C| > n/2$. From transitions (1) and (2) in the Markov chain definition, it can be seen that it is enough to have a consecutive sequence of $1/(\alpha_w \min\{ WB_{\mathcal{Y}} - a_i - WC_{\mathcal{T}}, WP_{\mathcal{C}} + a_i\}), \forall i \in C$, audits to enforce $p_C = 0$ for all covered workers. Which gives the following upper bound on the convergence time.

Theorem 2. *The expected convergence time is at most $\rho/(p_A^{min})^{\rho}$, where $\rho = 1/(\alpha_w \min_{i \in C}\{ WB_{\mathcal{Y}} - a_i - WC_{\mathcal{T}}, WP_{\mathcal{C}} + a_i\})$ and C is the set of covered workers.*

The upper bound shown in Theorem 2 may be too pessimistic for certain values of the parameters. The following theorem provides a tighter bound under certain conditions.

Theorem 3. *Let us define, for each worker i, $dec_i \triangleq \alpha_w \min\{ WP_{\mathcal{C}} + a_i, WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i\}$, and $inc_i \triangleq \alpha_w \max\{ WB_{\mathcal{Y}} - a_i, WC_{\mathcal{T}} + a_i\}$. Let C be the set of covered workers. If $p_A^{min} = \max_{i \in C}\{inc_i/(inc_i + dec_i)\} + \varepsilon$, for some $0 < \varepsilon < 1 - \max_{i \in C}\{inc_i/(inc_i + dec_i)\}$, the expected convergence time is $1/(\varepsilon \min_{i \in C} dec_i)$.*

The following corollary is derived from the previous theorem for a suitable scenario.

Corollary 1. *If $WP_{\mathcal{C}} + a_i \geq WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i$ and $WB_{\mathcal{Y}} - a_i \leq WC_{\mathcal{T}} + a_i$, $\forall i \in C$, and if*

$$p_A^{min} = \frac{WC_{\mathcal{T}} + \max_{i \in C} a_i}{WB_{\mathcal{Y}}} + \varepsilon,$$

where C is the set of covered workers and $0 < \varepsilon < 1 - (WC_{\mathcal{T}} + \max_{i \in C} a_i)/WB_{\mathcal{Y}}$, then the expected convergence time is ρ/ε, where $\rho = 1/(\alpha_w(WB_{\mathcal{Y}} - WC_{\mathcal{T}} - \max_{i \in C} a_i))$.

Probabilistic Bound on the Number of Rounds for Convergence: We show now that, under certain conditions on the parameters of the system, it is possible to bound the probability to achieve convergence and the number of rounds to do so. Assume that $p^0_{\mathcal{A}} > 0$. Since $p_{\mathcal{A}}$ is not changed unless the master audits, we have the following.

Lemma 6. *Let $p^0_{\mathcal{A}} = p > 0$. Then, the master audits in the first $\rho = \ln(1/\varepsilon_1)/p$ rounds with probability at least $1 - \varepsilon_1$, for any $\varepsilon_1 \in (0, 1)$.*

Let us assume that the system parameters are such that, for all workers i, $\alpha_w(WP_{\mathcal{C}} + a_i) \in [0, 1]$ and $\alpha_w(WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i) \in (0, 1]$ (all workers are covered). Let us define $dec\_cheater \triangleq \alpha_w \min_i\{WP_{\mathcal{C}} + a_i\}$ and $dec\_honest \triangleq \alpha_w \min_i\{WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i\}$. From transitions (1) and (2) we derive the following lemma.

Lemma 7. *Let r be a round in which the master audits, and F be the set of cheaters in round r. Then,*

$$p^r_{Ci} \leq 1 - \alpha_w(WP_{\mathcal{C}} + a_i) \leq 1 - dec\_cheater, \forall i \in F$$
$$p^r_{Cj} \leq 1 - \alpha_w(WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_j) \leq 1 - dec\_honest, \forall j \notin F$$

Denoting the sum of all cheating probabilities before a round r as $P^{r-1} \triangleq \sum_i p^{r-1}_{Ci}$.

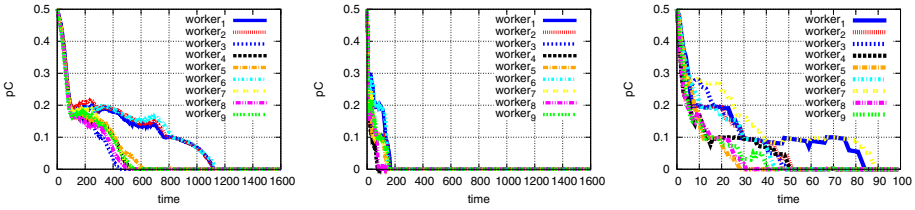
Lemma 8. *Let r be a round in which the master audits such that $P^{r-1} > n/3$. If $dec\_cheater \geq dec\_honest$ and $dec\_cheater + 3 \cdot dec\_honest \geq 8/3$, then $P^r \leq n/3$ with probability at least $1 - \exp(-n/96)$.*

Let us now define $dec_i \triangleq \alpha_w \min\{a_i, WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i\}$. Let, $dec \triangleq \min_i dec_i$. Assume $WP_{\mathcal{C}} \geq 0$ and $a_i \geq 0$, for all workers.

Lemma 9. *Consider a round r such that $P^{r-1} \leq n/3$. Then, with probability at least $1 - \exp(-n/36)$ each worker i has $p^r_{Ci} \leq \max\{0, p^{r-1}_{Ci} - dec\}$, and hence $P^r \leq n/3$.*

Lemmas 6–9 lead to the following result:

Theorem 4. *Assume $\alpha_w(WP_{\mathcal{C}} + a_i) \in [0, 1]$ and $\alpha_w(WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i) \in (0, 1]$ for all workers i. (Observe that all workers are covered.) Let $dec\_cheater \triangleq \alpha_w \min_i\{WP_{\mathcal{C}} + a_i\}$, $dec\_honest \triangleq \alpha_w \min_i\{WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i\}$, and $dec \triangleq \alpha_w \min_i\{a_i, WB_{\mathcal{Y}} - WC_{\mathcal{T}} - a_i\}$. If $p^0_{\mathcal{A}} = p > 0$, $dec\_cheater \geq dec\_honest$ and $dec\_cheater + 3 \cdot dec\_honest \geq 8/3$, then eventual convergence is reached in at most $\ln(1/\varepsilon_1)/p + 1/dec$ rounds, with probability at least $(1 - \varepsilon_1)(1 - \exp(-n/96))(1 - \exp(-n/36))^{1/dec}$, for any $\varepsilon_1 \in (0, 1)$.*



(a) (b) (c)

Fig. 1. Cheating probability for the workers as a function of time (number of rounds) for parameters $WP_{\mathcal{C}} = 0$, $WC_{\mathcal{T}} = 0.1$ and $a_i = 0.1$. (a) $\alpha = 0.01$, $WB_{\mathcal{Y}} = 1$; (b) $\alpha = 0.1$, $WB_{\mathcal{Y}} = 1$; (c) $\alpha = 0.1$, $WB_{\mathcal{Y}} = 2$.

5 Simulations

In this section we complement the theoretical analysis with simulations. Our analytical upper bounds on convergence time correspond to worst case scenarios. Here we present simulations for a variety of parameter combinations likely to occur in practice. We have created our own simulation setup by implementing our mechanism; technical details can be found in [17]. Each depicted plot value represents the average over 10 executions of the implementation.

We choose sensible parameter values, likely to be encountered in real applications. In particular, the number of workers has been set to nine (providing majority). Nine workers seems like an appropriate workforce, compared to Seti-like systems using three workers. The initial cheating probability of each worker i is not known, and therefore we have set it at $p_{Ci} = 0.5$ as a reasonable assumption. Similarly, we have set $p_{\mathcal{A}} = 0.5$ as the master's initial probability of auditing. The minimum probability of cheating is set to be $p_{\mathcal{A}}^{min} = 0.01$ and tolerance $\tau = 0.5$, hence the master will not tolerate a majority of cheaters.

The payoffs for the workers are set using $WB_{\mathcal{Y}} \in \{1, 2\}$ as our normalizing parameter and we take in analogy $WP_{\mathcal{C}} = 0$ and $WC_{\mathcal{T}} = 0.1$ as realistic values to explore the effects of these choices. The aspiration is a parameter defined by the workers in an idiosyncratic manner; for simplicity, here we consider all workers having the same aspiration level $a_i = 0.1$. The values of the aspiration and $WC_{\mathcal{T}}$ satisfy the necessary conditions of Theorem 1 and hence eventual convergence is reached. Finally, we consider the same learning rate for the master and the workers, i.e., $\alpha = \alpha_m = \alpha_w$. The learning rate, as discussed for example in [15] (called step-size there), for practical reasons it can be set to a small constant value; experimentally we notice that high values make the learning unstable. So we consider $\alpha \in \{0.1, 0.01\}$. A rich account of our results, on several scenarios under different parameter values (providing as well more intuition on system parameters e.g., tolerance) can be found in [17].

Figure 1 shows that convergence can be reached very quickly (in a few hundred rounds) even if no punishment is given to the workers caught cheating, and the number of workers and $WB_{\mathcal{Y}}$ are small. We also notice that a slightly higher value of α can make the convergence time shorter.

Comparing Figures 1(b) with 1(c) we observe that for a specific set of parameter values, a larger $WB_{\mathcal{Y}}$ leads to a shorter convergence time. Interestingly, this observation points out to a trade-off between convergence time and the cost the master has for reaching faster convergence and maintaining it. In this way, the master could choose between different protocols estimating the cost of the auditing during the whole interval to convergence: less auditing leads to larger convergence times, so it is not clear in principle what is going to be optimal.

6 Conclusions

This work applies reinforcement learning techniques to formulate the evolution of Internet-based master-worker computations. The mechanism developed is presented and analyzed. In particular we show that under necessary and sufficient conditions, the master reaches a state after which the correct task result is received at each round, with minimal cost. In addition we show that such state can be reached quickly. The convergence analysis is complemented with simulations; our simulation results suggest that when having a positive reinforcement learning (i.e., $WP_{\mathcal{C}} = 0$) the master can reach fast convergence, while applying negative reinforcement learning (i.e., $WP_{\mathcal{C}} = \{1, 2\}$) provides even faster convergence (see [17]). In fact, we may conclude that applying only negative reinforcement is enough to have fast convergence.

Acknowledgments. This work is supported by the Cyprus Research Promotion Foundation grant ΤΠΕ/ΠΛΗΡΟ/0609(BE)/05, NSF grants CCF-0937829, CCF-1114930, Comunidad de Madrid grant S2009TIC-1692, Spanish MOSAICO and RESINEE grants and MICINN grant TEC2011-29688-C02-01, and National Natural Science Foundation of China grant 61020106002. We thank Carlos Diuk for useful discussions.

References

[1] Abraham, I., Dolev, D., Goden, R., Halpern, J.Y.: Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In: Proc. of PODC 2006, pp. 53–62 (2006)
[2] Anderson, D.: BOINC: A system for public-resource computing and storage. In: Proc. of GRID 2004, pp. 4–10 (2004)
[3] Bendor, J., Mookherjee, D., Ray, D.: Aspiration-based reinforcement learning in repeated interaction games: An overview. International Game Theory Review 3(2-3), 159–174 (2001)
[4] Bush, R.R., Mosteller, F.: Stochastic Models for Learning. Wiley (1955)
[5] Camerer, C.F.: Behavioral game theory: Experiments in strategic interaction. Roundtable Series in Behavioral Economics (2003)
[6] Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press (2010)
[7] Fernández, A., Georgiou, C., Lopez, L., Santos, A.: Reliably executing tasks in the presence of untrusted processors. In: Proc. of SRDS 2006, pp. 39–50 (2006)

[8] Fernández Anta, A., Georgiou, C., Mosteiro, M.A.: Designing mechanisms for reliable Internet-based computing. In: Proc. of NCA 2008, pp. 315–324 (2008)
[9] Fernández Anta, A., Georgiou, C., Mosteiro, M.A.: Algorithmic Mechanisms for Internet-based Master-Worker Computing with Untrusted and Selfish Workers. In: Proc. of IPDPS 2010, pp. 1–11 (2010)
[10] Gintis, M.C.: Game Theory Evolving. Princeton University Press (2000)
[11] Maynard Smith, J.: Evolution and the Theory of Games. Cambridge U. Press (1982)
[12] Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior 35, 166–196 (2001)
[13] Phelps, S., McBurney, P., Parsons, S.: Evolutionary mechanism design: A review. Journal of Autonomous Agents and Multi-Agent Systems (2010)
[14] Shneidman, J., Parkes, D.C.: Rationality and Self-interest in P2P Networks. In: Kaashoek, F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 139–148. Springer, Heidelberg (2003)
[15] Szepesvári, C.: Algorithms for Reinforcement Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2010)
[16] Taufer, M., Anderson, D., Cicotti, P., Brooks, C.L.: Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing. In: Proc. of IPDPS 2005 (2005)
[17] Technical report of this work, TR-12-02, Dept. of Computer Science, University of Cyprus (February 2012), `http://www.cs.ucy.ac.cy/~chryssis/EvolMW-TR.pdf`
[18] Yurkewych, M., Levine, B.N., Rosenberg, A.L.: On the cost-ineffectiveness of redundancy in commercial P2P computing. In: Proc. of CCS 2005, pp. 280–288 (2005)