Noname manuscript No.
(will be inserted by the editor)

Evgenia Christoforou ·
Antonio Fernández Anta ·
Chryssis Georgiou ·
Miguel A. Mosteiro · Angel Sánchez

Crowd computing as a cooperation problem: an evolutionary approach

Received: date / Accepted: date

Abstract Cooperation is one of the socio-economic issues that has received more attention from the physics community. The problem has been mostly considered by studying games such as the Prisoner's Dilemma or the Public Goods Game. Here, we take a step forward by studying cooperation in the context of crowd computing. We introduce a model loosely based on Principal-agent theory in which people (workers) contribute to the solution of a distributed problem by computing answers and reporting to the problem

This work is supported by the Cyprus Research Promotion Foundation grant TE/HPO/0609(BE)/05, the National Science Foundation (CCF-0937829, CCF-1114930), Comunidad de Madrid grant S2009TIC-1692 and MODELICO-CM, Spanish MOSAICO, PRODIEVO and RESINEE grants and MICINN grant TEC2011-29688-C02-01, and National Natural Science Foundation of China grant 61020106002.

E. Christoforou
University of Cyprus, Nicosia, Cyprus
E-mail: evgenia.christoforou@gmail.com

A. Fernández Anta
Institute IMDEA Networks, Madrid, Spain
and Universidad Rey Juan Carlos, Madrid, Spain
E-mail: antonio.fernandez@imdea.org

C. Georgiou
University of Cyprus, Nicosia, Cyprus
E-mail: chryssis@cs.ucy.ac.cy

M. A. Mosteiro
Kean University, Union, NJ, USA
and Universidad Rey Juan Carlos, Madrid, Spain
E-mail: miguel.mosteiro@urjc.es

A. Sánchez
GISC/Matemáticas, Universidad Carlos III de Madrid, Spain
and BIFI, Universidad de Zaragoza, Spain
E-mail: anxo@math.uc3m.es

proposer (master). To go beyond classical approaches involving the concept of Nash equilibrium, we work on an evolutionary framework in which both the master and the workers update their behavior through reinforcement learning. Using a Markov chain approach, we show theoretically that under certain – not very restrictive – conditions, the master can ensure the reliability of the answer resulting of the process. Then, we study the model by numerical simulations, finding that convergence, meaning that the system reaches a point in which it always produces reliable answers, may in general be much faster than the upper bounds given by the theoretical calculation. We also discuss the effects of the master's level of tolerance to defectors, about which the theory does not provide information. The discussion shows that the system works even with very large tolerances. We conclude with a discussion of our results and possible directions to carry this research further.

Keywords Evolutionary game theory · cooperation · Markov chains · crowd computing · reinforcement learning

PACS 89.20.Ff · 89.65.-s · 89.75.-k · 02.50.Le · 02.50.Ga

1 Introduction

Physicists, and in particular statistical physicists, have long been interested in understanding social phenomena by using their own disciplinary toolbox. While early efforts in this direction go back hundreds of years [1], only in the last two decades has the application of physics to social systems grown to involve a respectable number of researchers and to reach results of true interest for practitioners in the field. Thus, the success in the early nineties of econophysics [2], which eventually led to important contributions to the field of quantitative finance, was subsequently followed by a rapid widening of the scope of this research, and nowadays many different socio-economic problems are investigated with physics-inspired approaches [3].

One of the physics paradigms that has proven most fruitful to make connections with social science questions is that of emergence [4], i.e., the realization that the properties of large aggregates of entities can not be simply deduced from the properties of the said entities: At each level of *complexity*, entirely new, *emergent* properties appear. Thus, water molecules are water molecules and behave as such no matter what form water takes, whether ice, liquid, or vapor. Statistical physics taught us that it is possible to understand these phenomena by considering not only the constituents of the system of interest, but also their interactions with each other [5]. Not many years after Anderson's seminal paper, the same epiphany was experienced by Schelling [6], who attempted, among other things, to explain the segregation of populations of different races on the basis of the individual motives of each person. In doing so, he came up with a model which was a specific version of another one already well-known to physicists, namely the Ising model [7]. However, there is a crucial difference between the two viewpoints: while physicists are interested in optimizing a global quantity (e.g., the Hamiltonian), the approach of the social scientist involves optimizing quantities at the individual level (e.g., utility functions [9]).

Aside from the local vs global perspectives, another important characteristic must be taken into account when dealing with social issues: the fact that interactions (which, let us keep in mind, are the responsible for emergent phenomena) are strategic [10], i.e., an individual's situation is fully dependent on the choices or actions of others, and all parts know this and have the wit to use this awareness to their advantage. When such cognitive capabilities must be considered, the indispensable tool to describe interactions is game theory [11]. In its original formulation, game theory was a mathematical tool to analyze situations of conflict or cooperation in economics, but its applicability has grown with the decades and now reaches many disciplines, ranging from biology [12] to the whole spectrum of the behavioral sciences [13]. Loosely speaking, a game consists of a number of players, a set of strategies for each of the players, and a payoff function that associates a payoff to each player with a choice of strategies by each player [14,15]. Within this setup, the key question is to find what are the strategies every player has to use to maximize her profits. The classical answer to this was given by Nash [16], through his concept of equilibrium: a Nash equilibrium is a set of strategies, one per player, such that no player has incentives to choose a different strategy unilaterally. In this formalism, given a game, Nash equilibrium is the behavior that should be observed, i.e., the game has been solved.

Unfortunately, this concept has a number of problems as soon as games with some degree of complication are considered. For instance, games may have more than one equilibrium, and in that case one needs to introduce refinements, more restrictive equilibria concepts, in order to select one of them as the solution of the game. On the other hand, a natural question arises when one thinks that players must be able to compute the Nash equilibria in order to play it, a task that is often very difficult. One possible solution to this problem arises from evolutionary game theory, introduced by Maynard-Smith [12] in order to apply game theoretical ideas to understand animal conflict. As Gintis [13] puts it, in evolutionary game theory, successful strategies diffuse across populations of players rather than being learned inductively by disembodied rational agents. This in turn leads to a dynamical approach to the evolution of strategies [18–20]. The dynamics that has received more attention is described by the replicator equation [21], which arises when one postulates that strategies appear in the next generation with probability proportional to the difference of their payoff with the mean payoff of all strategies in the population. Thus, payoff is assimilated to the biological concept of fitness, or reproductive success. In this framework, it can be proven that the stable points of the dynamics coincide with the Nash equilibria of the game under consideration. This is commonly taken as an explanation of how people learn to play the equilibrium of a game: Either they learn (and then evolution means here that people change the way they decide their actions as the game proceeds) or, in the case of irrational animals, their behavior is genetically determined and those who fare better have larger progenies.

Within the game theoretical formalism, one of the most important long unsolved puzzles is how cooperative behavior can emerge and be stable. Indeed, ever since Darwin's time [22], understanding how individuals that help others at their own expenses can survive has posed serious questions. This problem is very often formulated in terms of social dilemmas [23] such as the Prisoner's Dilemma [14,15]. For instance, in the Prisoner's Dilemma a cooperator provides its partner with a benefit b at a cost, c, for herself, with $c < b$; a defector receives the benefit, but does not cooperate. Therefore, two cooperators both obtain a payoff of $b - c$, two defectors receive nothing, and a defector facing a cooperator earns b, whereas the cooperator receives only $-c$. It would be better for both to cooperate, but on the other hand it is easy to see that the only Nash or evolutionary equilibrium of the game is to defect, which leads to both players getting nothing. Several mechanisms have been proposed to explain why is it that we actually observe cooperation at a large scale in human and other animal societies [24], but the question is far from being solved yet. In particular, a lot of research has been devoted to mechanisms leading to deviations from the replicator equation [25], which could turn cooperation into an equilibrium, with different degrees of success and agreement with experiments [26,27]. It is important to stress that this is one of the many aspects that have been studied within the statistical physics approach to evolutionary game theory on networks. Indeed, literally hundreds of papers have been published [28], including an increasing amount of literature on networks co-evolving with the strategies [29]. As it would be very lengthy to summarize all that work here, the reader is referred to these references for details and results on the many systems studied.

Here, we contribute to the literature of cooperation by moving away from well-known paradigms such as the one described in the preceding paragraph and turning instead to a situation closer to different real life applications, namely the Principal-agent problem [30,31], that is, the question of how to motivate one party (the agent) to act on behalf of another (the principal). Agency problems arise in a variety of different contexts: Thus, CEO's should act on behalf of shareholders, lawyers are supposed to act in the best interest of their clients, employees work for their employers, and so on. For the sake of concreteness, and also for its interest in many cases, here we focus on the specific case of crowd computing.

Crowd computing has arisen in the last few years as an alternative to expensive supercomputers. The idea is to use the Internet as a platform for processing complex computational jobs, where complexity may come because the task is in itself very complex, so it needs to be split in many different subtasks, or because it is simple but must be repeated many times. Already, some Internet-based applications are taking advantage of this global computation infrastructure, being either volunteer-based "@home" projects [32] such as SETI [33] and Ibercivis [34], or profit-seeking computation platforms such as Amazon's Mechanical Turk [35]. This type of distributed computation has already been considered from the traditional viewpoint in economics, namely the fully-rational agent [36–39]: 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. This is precisely the process involved in crowd computing: a central computer, the master, sends tasks to the volunteers' computers, the workers, who carry out the task and report back the result they obtain.

We are thus faced with a cooperation problem. Indeed, crowd computing relies on people providing either computing time on their computers or performing the computations themselves and reporting the result. However, participating individuals may in fact be untrustworthy [32,40,41], i.e., they may be tempted to defect by cheating. Reporting cheaply-obtained incorrect results is the rational choice [40,42,43] whenever the computation is costly (be it in terms of time or money). Therefore, the master must take actions that ensure the correctness of the result (in other words, that makes workers cooperate, not defect). Thus, for instance, SETI attempts to prevent obtaining bogus results through redundancy (i.e., by assigning the same task to several workers and comparing their outcomes [32,44]). This is actually the case we are going to consider in detail in this paper. To this end, we will make use of a game-theoretical model introduced in [45,46], which was understood in the classical sense, i.e., interaction was one-shot and assumed neither prior knowledge of the actions of the agent nor reactions to them or future interactions. This is unrealistic in many situations in which individuals participating in the crowd computing process do so on a regular basis. In fact, it is often the case that their payoff arises from the reputation they receive by being listed as top contributors by the master, such as in SETI or Ibercivis. Therefore, an evolutionary approach to the problem is needed to understand in depth this type of collaborative enterprises and the incentives involved in them.

To report our results, the paper is organized as follows. The next section presents the model in detail. Subsequently, Sec. 3 collects some analytical results we have obtained by using a Markov chain formalism on the convergence to well-behaved (i.e., cooperator) workers. Sec. 4 probes further on this issue by means of computer simulations, showing that convergence is often achieved in much shorter times than those predicted by theory, and discussing how it depends on the model parameters. Finally, Sec. 5 concludes the paper by summarizing our findings and discussing possible extensions.

2 Model

2.1 Baseline model

Let us consider a distributed system consisting of a master processor that assigns computational tasks to a set W of n workers. For reasons that will become apparent later, and that suppose no serious limitation to our model, we will assume n is odd. In every *round* of the process, the master sends one task, the same to all workers, who compute and return the obtained results. We will further assume that the correct answer to the task is unique; admittedly, this does restrict somewhat the validity of our results, but it simplifies the analysis and at the same time there are very many real-life cases in which this is actually the case. Based on the workers' answers, the master chooses what she believes is the correct answer to the task. Workers, who are rational players in a game-theoretical sense, can report the correct answer after having computed it (honest or cooperative workers), or can report a wrong answer that requires much less or no computing cost (cheaters or defectors). In this last case, we consider the worst case scenario, in which cheaters collude towards providing the same wrong answer (as, for example, in [44]), which makes it much more difficult to detect by the master. The decision on the best action will of course depend on the parameters of the model. In this scheme, we characterize workers' strategy by their individual cheating probability, p_{Ci}, for worker i. This probability is not a fixed quantity but rather it may change with the computing rounds, i.e., workers can readjust their strategy depending on the outcome of previous rounds (see below).

Clearly, the master's goal is to obtain reliable answers and, in fact, to build a reliable computational platform. The master has three tools at her disposal to achieve this goal: auditing, rewarding and punishing. In a given round, the master may audit the response of the workers (at a cost), i.e., she computes the task herself, thus finding which workers have been truthful or not. We call $p_{\mathcal{A}}$ the probability of the master auditing the responses of the workers and, as p_{Ci}, it may change in time. We assume that there is a value $p_{\mathcal{A}}^{min} > 0$ so that at all times $p_{\mathcal{A}} \geq p_{\mathcal{A}}^{min}$. This reflects the fact that, even if the master does not want to audit because it is costly, she will not stop auditing completely under any circumstances. This is a somewhat technical point and, given that $p_{\mathcal{A}}^{min}$ may be very small, it is not a restrictive assumption. Having the possibility of auditing in mind, the master proceeds in the following way: either she audits and therefore knows the correct answer, or she does not audit, and then takes as the correct answer that provided by the majority of workers. Whatever the case, an answer to the task is always obtained, allowing the master to achieve her computational objective and, importantly, to reward or punish the workers.

Rewarding and punishing are the other tools at the master's disposal to encourage cooperative behavior. As we have already stated, rewarding may not necessarily be of monetary type, and retribution in terms of reputation are very common in volunteering schemes. Obviously, in the Amazon Mechanical Turk, the only available reward is a payment for the person's work. We will not enter here on the nature of the payment and look at the question from an abstract viewpoint. On the other hand, there is abundant evidence that punishment promotes cooperation [47,48]. The effect of punishment will be one of the questions we will look into below. The general reward-punishment scheme is then as follows [45,46]: Workers providing the right answer (irrespective of it being declared right because of auditing or because of being in the majority) will receive a reward $WB_{\mathcal{Y}}$ (which may in principle cost a different amount $MC_{\mathcal{Y}}$ to the master). In case the master audits, cheaters will receive a punishment $WP_{\mathcal{C}}$ (assumed costless for the master, i.e., it should be understood as a fine). Computing the task is of course costly (otherwise the problem is trivial) and its cost is $WC_{\mathcal{T}}$. For the master, auditing has also a cost (arising from carrying out the computation or checking whether the answer is correct when input into the problem), which we denote by $MC_{\mathcal{A}}$. Accepting a wrong answer is also costly for the master, and the corresponding parameter is $MP_{\mathcal{W}}$. Finally, the right answer provides the master with a benefit $MB_{\mathcal{R}}$. These payoffs are summarized in

Table 1 and it is important to stress that they are all non-negative, so they must enter the calculation with their corresponding sign.

$WP_{\mathcal{C}}$	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

Table 1 Summary of the parameters of the baseline game of our model.

2.2 Evolutionary dynamics

In the preceding subsection, we have introduced the baseline game underlying our model. As was done in [45,46], this can be treated on its own from the viewpoint of classical game theory, finding the Nash equilibria for a one-shot game. However, our aim in this paper is to study how the system behaves when the interaction between master and workers takes place repeatedly, and they can actually change their actions and the corresponding governing strategies in view of previous exchanges. The only prior example of this perspective we are aware of is that of Rose and Willemain [49] who introduced both evolution and bounded rationality in the principal-agent model. Players' learning is simulated with a genetic algorithm that roughly simulates selection and mutations in biological evolution.

Here, we use a completely different approach, namely the approach of reinforcement learning [50–53]. Reinforcement learning models how the learner interacts with the environment to decide upon a strategy and use its experience to select or avoid actions according to the consequences observed. Empirical evidence [54] in general strategic contexts suggests that reinforcement learning is more plausible with players that have information only about the payoffs they receive. Even then, they may 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 evolution. Camerer [53] argues that "reinforcement learning is a reasonable theory for players with very imperfect reasoning ability (...) or human players who know absolutely nothing about the forgone or historical payoffs from strategies they did not choose". Thus, we make use of reinforcement as a learning method, since workers are rational, and have sufficient cognitive skills and no idea about the payoffs from strategies they did not choose. Moreover the algorithm assumes workers have no information about other workers' strategies, received payoffs, or master's $p_{\mathcal{A}}$. Within the specific context of cooperation models and social dilemmas, reinforcement learning has been used before [55–57], showing that such dynamics can indeed lead to different forms of cooperation in a manner that relates to what is observed in real situations.

To implement reinforcement learning, we add time evolution to the two probabilities we introduced earlier, namely the workers cheating probability, p_{Ci}, and the masters auditing probability, $p_{\mathcal{A}}$. Further, we assume that, in every round, worker i has an *aspiration* a_i, which stands for the minimum benefit she expects to obtain. Of course, for the worker to take part in the project, the master must make sure that $WB_{\mathcal{Y}} \geq a_i$, or, equivalently, that workers can have their aspirations fulfilled. We will assume that the master knows the workers' aspiration as, typically, master and workers have agreed on a contract prior to the start of the computation. Having included the possibility of time evolution, we can state the master's goal as follows: After some finite number of rounds, the system must guarantee correctness (the master obtains the correct task results in every round) with probability 1 forever. We will refer to this goal as *eventual correctness.*

Let us now specify how the two probabilities evolve in time, beginning with the master's probability of auditing. Clearly, the master can only assess the cooperation level of the workers when she audits, otherwise she does not have any reliable information on the truth of their answers. At every round in which the master has audited, she changes her auditing probability according to the following rule:

$$p_{\mathcal{A}}^{t+1} = \min\left\{1, \max\left\{p_{\mathcal{A}}^{min}, p_{\mathcal{A}}^{t} + \alpha_m\left(\frac{cheaters(t)}{n} - \tau\right)\right\}\right\}, \qquad (1)$$

where superindices t and $t+1$ correspond to consecutive instants of time, $cheaters(t)$ is the number of workers that provided an incorrect answer at time t, and τ stands for the tolerance of the master, i.e., the number of cheaters she is ready to accept. 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. As we will see, tolerance is an important parameter which we will study in detail. This rule has to be supplemented with an appropriate initial value $p_{\mathcal{A}}^0$. Assuming the master has no prior knowledge of her worker pool, a reasonable approach is to initially set $p_{\mathcal{A}} = 0.5$. In any event, we have checked that the particular choice of initial condition does not change qualitatively our results below. On the other hand, when the answers are not audited, the master has no information about the number of cheaters, hence the rationale behind not updating the auditing probability in that case. Finally, we note that with this choice for the evolution of the master's auditing probability, we need not consider anymore the master's parameters in the payoffs Table 1, as they play no role in Eq. (1). This is a reduction in the number of parameters which for a first study like this is very useful, and will allow us to gain insight on the behavior of the model.

As for the workers, they also proceed with a reinforcement learning update based on their aspiration level. When the master has decided on the correct answer and distributed the corresponding rewards or punishment, workers revise their probability to cheat, according to

$$p_{Ci}^{t+1} = \max\{0, \min\{1, p_{Ci}^{t} - \alpha_w(\Pi_i - a_i)S_i\}\}, \tag{2}$$

where Π_i stands for the payoff worker i receives from the master and S_i represents the previous action of the worker, namely $S_i = 1$ if the worker cooperated and provided the correct answer, and $S_i = -1$ if the worker defected, i.e., cheated. What this scheme models is that workers update their strategy based on the payoffs of the current round. The standard by which this experience is judged to be satisfactory is defined by the aspiration level a_i, against which achieved payoffs, and the actions leading to them, are evaluated. The worker's learning rate is denoted as α_w. For simplicity, we assume that all workers have the same learning rate, but our results can be generalized to the case that workers have different learning rates.

3 Analytical results

The model we have just described can be formulated in terms of a Markov chain approach for the probabilities of the master and the workers. Indeed, if we denote the state of the Markov chain by the vector of probabilities $(p_{\mathcal{A}}, p_{C1}, p_{C2}, \ldots, p_{Cn})$, one can write equations for their evolution and note that it only depends on the values at the previous round. This theoretical analysis was carried out in detail in [58], and therefore we will not repeat it here. However, for the sake of completeness, we find it appropriate to quote the main results that can be rigorously proved about the model.

The first result arising from our analytical work is that eventual correctness, i.e., the goal of reaching absolute certainty that the results provided by the system are correct in a finite number of rounds, is achieved. This is shown by the following theorem:

Theorem 1 *If $p_{\mathcal{A}} > 0$ then, after some finite number of rounds, the system achieves eventual correctness if and only if $WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}$ for all $i \in Z$ in some set $Z \subseteq W$ such that $|Z| > n/2$.*

In words, this theorem is a simple statement: if the master wants to reach a situation in which her crowd-computing scheme is providing the correct answer with probability 1, all she needs to do is make sure that a majority of workers have their aspirations (taking into account the cost of computing) satisfied. Note that the workers may have different aspirations and as these are revealed to the master she can optimize in choosing the value of $WB_{\mathcal{Y}}$, the worker's benefit from her answer being considered correct, just large enough to satisfy the majority of workers with lower aspiration levels. On a more technical note, results like this are those requiring the number of workers to be odd to be proven rigorously.

The question that immediately arises after this first result is what is the convergence time. That is, the theorem shows that eventual correctness is reached, but how long does it take? In [58] we were also able to answer this question by looking at the set of *covered workers*, i.e., those whose aspiration level and computing cost combined are smaller than the reward for providing the correct answer:

Theorem 2 *Let C be the set of all covered workers, with $|C| > n/2$, as required by Theorem 1. The expected convergence time is at most $\rho/(p_A^{min})^{\rho}$, where $\rho = 1/(\alpha_w \min_{i\in C}\{WB_{\mathcal{Y}} - a_i - WC_{\mathcal{T}}, WP_{\mathcal{C}} + a_i\})$.*[1]

As we can see, the time to reach eventual correctness depends basically on three things: the learning rate of the master, α_w, the benefit to the workers from providing a correct answer, $WB_{\mathcal{Y}} - a_i - WC_{\mathcal{T}}$, and the punishment $WP_{\mathcal{C}}$. Thus, if the master reacts slowly to the actions of the workers, then the convergence time may be very long. Therefore, it is important that the master reacts swiftly to the workers' input, albeit too large a value of α_w leads to problems because of overreactions, as we will discuss in the next section. It is also important to mention that the participants in the set of covered workers should be the same throughout the computation, which in general will always be the case if they keep their aspiration level constant. On the other hand, Theorem 1 states that covering workers' aspirations is enough, but this second theorem tells us that the master should not be too greedy, because if the net reward is very meager, then the convergence time increases enormously. Finally, the theorem also gives a somewhat unexpected role to the punishment, in such a way that no matter what the reward is, if there is no punishment it seems that the convergence time may be large. This is an issue that we will certainly need to look at in the simulations and will be discussed in the corresponding section. Nevertheless, from the analytical perspective, we found that we could remove the dependence on the punishment by being a little more restrictive:

Theorem 3 *Let C be the set of all covered workers, with $|C| > n/2$. 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 $0 < \varepsilon < 1 - (WC_{\mathcal{T}} + \max_{i\in C} a_i)/WB_{\mathcal{Y}}$, then the expected convergence time is ρ/ε, where $\rho = 1/(\alpha_w(WB_{\mathcal{Y}} - WC_{\mathcal{T}} - \max_{i\in C} a_i))$.

Thus, if a higher minimum auditing probability is affordable for the master (an increased number of audits yields a more costly computation) then she can reach eventual correctness in a reasonable time that is independent of any possible punishment.

As can be seen, the main conclusion of this section is, first, that the model does indeed allow to reach eventual correctness, and second, that the time to reach it can be bounded as above. Interestingly, the bound does not depend explicitly on the number of workers, albeit of course the more workers there are, the more the master needs to invest in rewards to ensure that half of them have their aspirations covered, but that does not affect the bound provided. This is quite a satisfactory result that arises from the lack of interactions between the workers, as all we are assuming is that those who cheat give the same wrong answer (which is a worst case scenario as we

[1] For technical reasons, α_w cannot be very large so the possible changes in p_{Ci} are bounded. This is not a problem because reinforcement learning with large values of α may show pathological behavior, so we will stay within this constraint.

have already said). Therefore, the master is actually playing a collection of independent principal-agent games. We will come back to this issue in the discussion. To conclude this section on analytical results, we note that it is possible to obtain more quantitative results if stochastic bounds are searched for, i.e., ensuring eventual correctness with a probability higher that some specified value, smaller than one. The reader is referred to [58] for details on this approach.

4 Simulation results

4.1 Parameters

Having presented what we learned from our theoretical analysis, it is now needed to go beyond that and to test parameters and scenarios not covered by the theoretical analysis. Of particular importance is the issue of the convergence of the system to providing correct answers and the range of parameters under which this takes place in practically useful time. On the other hand, one of the theoretical constraints is the very large number of workers to which the master has to send the same computational task. Having such a large number of workers affects the benefit the master obtains from the computation, especially in the case workers are profit-seeking ($MC_{\mathcal{Y}} = WB_{\mathcal{Y}}$). As we will see, even if we relax to a large extent the constraints of the theorems (e.g. number of workers), convergence can be fast and the system performs very well.

Going to numerical simulations requires fixing values for the parameters. Our model has quite a few of them: the number of workers, three values for payoffs, costs and punishments (seven in Table 1 of which for the current model we disregard the master's parameters as we have already discussed), two learning rates, one for the master and one for the workers, the master's tolerance of cheating, and the minimum auditing probability. Aspiration levels must also be considered; in the analysis we assumed that they could be different for each worker for the sake of generality, but for simplicity we will consider here an homogeneous population (which can always be taken as a worst case scenario for populations whose maximum aspiration level equals our uniform value). In addition, there is the question of the initial values of the auditing and cheating probabilities. Therefore, it is clear that we need to fix several of these parameters in order to be able to make a meaningful simulation study.

To begin with, the initial cheating probability of each worker i is not known, and therefore $p_{Ci} = 0.5$ seems to be a reasonable assumption. Similarly, workers are rational and have no information about the master's initial probability of auditing. The same applies for the master's initial probability of auditing, hence, we have taken $p_{\mathcal{A}} = 0.5$ as the initial value. The minimum probability of auditing will be set to be $p_{\mathcal{A}}^{min} = 0.01$. As for the tolerance, this will be one of our parameters of interest so we will sweep a range of values in what follows.

Let us now proceed with the payoffs for the workers. We will take the worker's reward for correct answers, $WB_{\mathcal{Y}} = 1$, by way of normalization and, within that framework, we will choose $WP_{\mathcal{C}} = 0, 1$ and $WC_{\mathcal{T}} = 0.1, 0.2$ as realistic values (within the same order of magnitude as $WB_{\mathcal{Y}}$) to explore the effects of these choices. As we will see, those values of punishment are enough to understand the system behavior, and, as for the computing cost, we believe that it is reasonable to assume that it is not too large compared with the reward. Regarding the aspiration level, as we have already said, this is a parameter defined by the workers in an idiosyncratic manner; for simplicity, in these simulations we fix a uniform aspiration level, $a_i = 0.1$. We have checked that when values are assigned randomly around some mean, the results are similar to those presented here, provided the variance is not very large. Of course, the values we have set for the aspiration and for the workers' cost for computing the task are such that the necessary conditions of Theorem 1 are satisfied and hence eventual convergence should be 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 [59] (called step-size there), for practical reasons can be set to a small constant value; so we consider $\alpha = 0.01, 0.1$.

4.2 Results

Let us begin the report on our simulation results by looking at a few typical runs. Examples of the simulation outcome are shown in Fig. 1. As can be seen from the plots, the average cheating probability of the workers decays rapidly to zero, and eventual correctness is easily reached. In fact, introducing the parameters of the plots in the bound given by Theorem 2, we find that for practical purposes the value of the bound is infinite: values of ρ with the parameters in Fig. 1 go from 12.5 to 1000, and then the factor $p_{\mathcal{A}}^{min} = 0.01$ becomes extremely large when raised to ρ. Numerical results show very clearly that convergence is much more faster than the theorem, and therefore such a system would be useful for practical purposes. However, it is important to stress from the start that while the results in Fig. 1 are typical, we have found instances in which convergence was not achieved in 10^5 rounds. To illustrate this point, we have collected two instances in Fig. 2, obtained for a very large tolerance value, $\tau = 0.9$. Indeed, tolerance turns out to be the key parameter to understand the model behavior, so we will discuss its role in detail below. For the time being, focusing on the differences between realizations, it is clear that while in general convergence takes place in a swift manner, it is not always granted to occur in a reasonable amount of time (as also the analysis shows). We note that the decay on the cheating probability is always accompanied by a corresponding decrease of the master's auditing probability, which reaches its minimum value and remains at it forever. In fact, the master could actually not monitor anymore their workers, as their cheating probability exactly vanishes, albeit one expects that in practice, random, non-frequent audits would be in place in any functioning system.

Looking again at both Figs. 1 and 2, we see that there is more important information about the behavior of the model that we can extract. Thus, focusing on the results for different values of the aspiration level, we observe that the higher the value of a_i, the faster the system evolves towards eventual

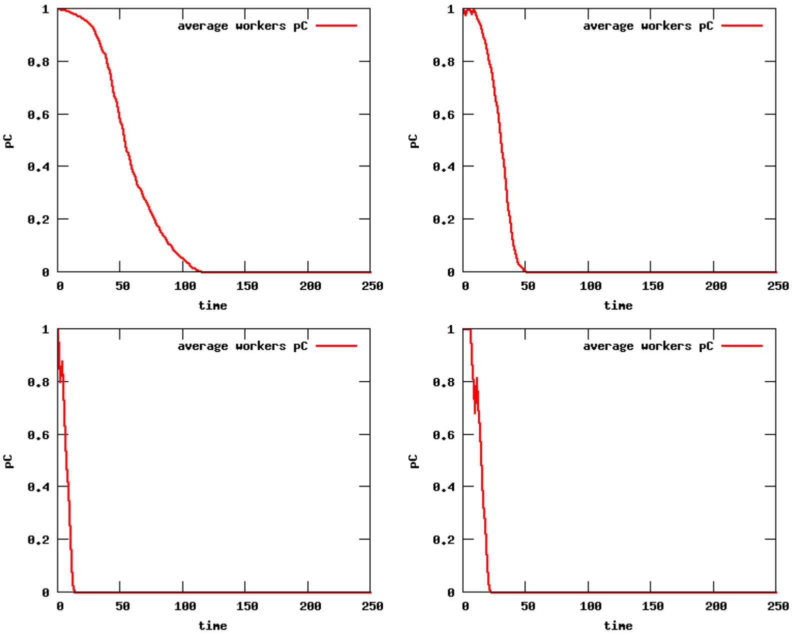

Fig. 1 Average cheating probability for the workers as a function of time (number of rounds) for a system with 100 workers. Top left: $WP_{\mathcal{C}} = 0$, $WC_{\mathcal{T}} = 0.1$, $a_i = 0.01$, $\alpha = 0.1$, $p_{\mathcal{A}}^{min} = 0.01$, and $\tau = 0.3$. Top right: as top left, but $a_i = 0.1$. Bottom left: as top left, but $WP_{\mathcal{C}} = 1$. Bottom right: as top left, but $a_i = 0.1$, $WP_{\mathcal{C}} = 1$ and $WC_{\mathcal{T}} = 0.2$.

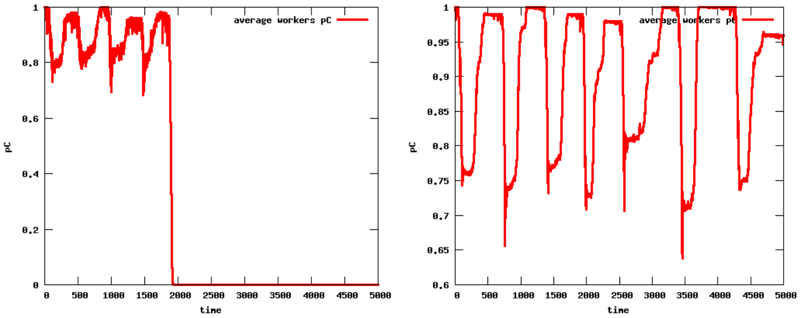

Fig. 2 Average cheating probability for the workers as a function of time (number of rounds) for a system with 100 workers. In both plots, $WP_{\mathcal{C}} = 0$, $WC_{\mathcal{T}} = 0.2$, $\alpha = 0.1$, $p_{\mathcal{A}}^{min} = 0.01$, and $\tau = 0.9$. Left, $a_i = 0.1$; right, $a_i = 0.01$.

correctness. In the example in Fig. 1, comparing the two upper panels we find that for an aspiration level one order of magnitude higher, convergence takes approximately half the time. On the contrary, in the lower panels the situation is reversed, and the system converges slightly faster for smaller aspiration rates (note that the bottom right panel corresponds to a different value of the cost, but we have checked that this is true keeping the cost constant). This result is in agreement with the bounds on convergence time of Theorem 2, which predict that when there is no punishment ($WP_{\mathcal{C}} = 0$) increasing aspiration levels improves the performance of the algorithm, while the opposite is true for large enough punishments (which is true with our choice $WP_{\mathcal{C}} = 1$). On the other hand, if the cost of computing is larger, convergence is slower, as shown by the lower plots in Fig. 1 where, to avoid having multiple panels, we simultaneously changed two parameters, both the aspiration level and the computing cost. The case with $WP_{\mathcal{C}} = 0$ is the most interesting one, as the results show that if the master does not want to punish the workers it is better to have workers with higher aspiration levels, contrary to what one can might expect. More evidence on the effect of the aspiration level arises from the extreme cases depicted in Fig. 2, where for the higher a_i (left panel) eventual correctness is achieved after a few oscillations of the average cheating probability, while for the lower a_i the system did not converge during the observed time. We thus conclude that aspiration level has effects which interact in a non trivial manner with the use of punishment, whereas the cost of computing goes in the foreseeable direction of increasing the convergence time.

Another factor we can discuss from the available data is the effect of punishment. Looking at the left panels of Fig. 1 it can be clearly noticed that convergence with punishment is much faster than for the case without punishment, by an order of magnitude approximately. Of course, we are showing results with a strong punishment, of the same order of the reward; we have checked that smaller values exhibit the same positive trend and that the improvement of the convergence is more or less proportional to the punishment. On the other hand, a large punishment would not make much sense, because it could take away from the worker the profit from multiple rounds of computation. Facing such scenario, workers might not perceive participating in the computation as beneficial. Furthermore, our reinforcement learning scheme only looks at the profit of the previous round. Thus, considering large punishments would not be consistent.

Having analyzed the effect of the different payoff-related parameters on the model behavior, i.e., payoffs (punishment and costs compared to reward, aspiration levels) we must now turn to the other key parameter of the system, the master's tolerance to cheating τ. As we have already seen, τ may have dramatic effects on the system performance. Fig. 3 collects our results on the influence of tolerance on convergence, showing the percentage of runs that converged to eventual correctness out of a batch of 50 realizations with a runtime of 1000 rounds for every tolerance value and four numbers of workers. The first conclusion we can draw from these plots is that for the case without punishment and with the lowest aspiration level (top left panel) all realizations yielded eventual correctness for tolerances as large as $\tau = 0.6$.

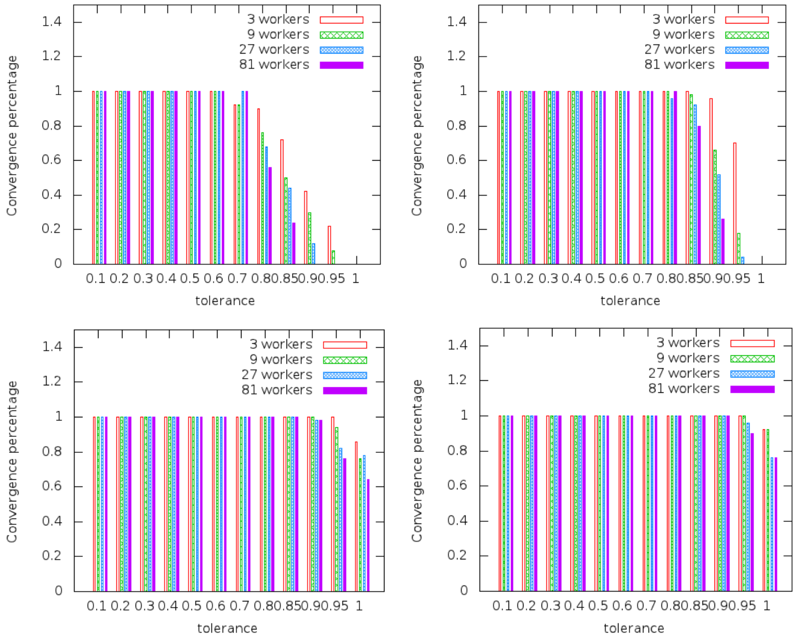

Fig. 3 Convergence percentage of 50 realizations in 1000 rounds. Initial condition in all cases was $p_c = 1$ for all workers, $p_A = 0.5$. Parameters in all panels, $WC_{\mathcal{T}} = 0.1$, $\alpha = 0.1$, $p_{\mathcal{A}}^{min} = 0.01$. Left panels: $a_i = 0.01$, right panels: $a_i = 0.1$. Top panels, $WP_{\mathcal{C}} = 0$, bottom panels, $WP_{\mathcal{C}} = 1$.

That is, the system remained well behaved (in the sense of providing correct answers) even if the master does not react to a majority of defectors. For higher tolerances, in agreement with what we advanced above, the system behavior worsens. Lack of convergence within our simulation time becomes more frequent. For those realizations that do not convergence, we cannot be sure that they will reach eventual correctness at some later time. Therefore, the percentage of realizations that converged would be a estimate for a lower bound to the probability of reaching eventual correctness. As we can see from the rest of the panels, increasing the aspiration level to $a_i = 0.1$ takes the maximum allowed tolerance that still has eventual correctness in every realization up to $\tau = 0.7$, whereas introducing punishment for defectors is also an improvement factor, leading to threshold tolerances of $\tau = 0.85$ for the low aspiration level and even to $\tau = 0.9$ for the largest one. We also checked that increasing the computing cost to $WC_{\mathcal{T}} = 0.2$ does not change the picture. A value such as $\tau = 0.9$ is certainly very large; the fact that the system ends up providing correct answers even when the master only updates her auditing probability in extreme cases shows the robustness of the design.

Interestingly, Fig. 3 also shows the relevance of the number of workers in the system performance. It appears from the plots that when the master takes her tolerance beyond the convergence threshold the probability to reach eventual correctness decreases if the number of workers is increased. Except for the results at the threshold ($\tau = 0.7$) in the case with no punishment and low aspiration (top left plot), increasing the number of workers always lead to worse results in terms of convergence for any value of the tolerance. We believe that this phenomenon arises because increasing the number of workers and keeping everything else the same, it is more probable that one of the workers defects. In this large tolerance region, the behavior would not change the master's auditing probability and, given that there is no punishment, reinforcement learning would not act so strongly on the corresponding worker, making it more difficult for the system to converge.

Further information on the dependence of the system on the number of workers is provided by Fig. 4, where we study one of the cases of Fig. 3 in more detail. For the lowest aspiration value, $a_i = 0.01$, we see that for tolerances $\tau = 0.75$ and higher, the convergence percentage decreases with the number of workers. The system sizes we are able to study at this time are not enough to ascertain whether it eventually reaches zero for sufficiently large number of workers, which would indicate an abrupt transition in the performance of the algorithm upon changing the tolerance. Increasing the aspiration level we observe, in agreement with our earlier remarks, that the system performs much better in terms of higher convergence rates, and also that the transition to the non-performing system seems somewhat more abrupt, going from full convergence (i.e., convergence percentage equal to 1) for 2217 workers at $\tau = 0.8$ to a very low value for $\tau = 0.9$. However, specific claims about the nature of this transition cannot be made without studying the whole parameter space. What we can certainly conclude is that the degrading of the performance becomes worse for larger number of workers, as hinted from previous plots, and that for tolerance values up to $\tau = 0.7$ the system always provides correct answers. We thus confirm the robustness of our design as far as the need for the master to update her auditing probability is concerned, as she can obtain good results just keeping the same value (and hence auditing less often and saving costs) unless an inordinate number of workers defect. In addition, having workers with high aspirations allows for an even lower rate of increase of the auditing probability.

5 Discussion

In this paper, we have presented a model for crowd computing, a collaborative enterprise with important applications both already working and under development. The model is based on the principal-agent models of economics but incorporates also elements of social dilemmas such as the temptation to defect or the punishment to promote cooperation. The other aspect of novelty of this work is that we bring dynamics into the literature on the problem of crowd computing by introducing evolution through reinforcement learning. This allows us to get insights on the problem beyond the assessment of equilibria which are relevant for practical purposes, such as typical convergence times. The formalism also allows us to look into behavioral parameters such as the tolerance about which a classical game-theoretical perspective provides no information at all.

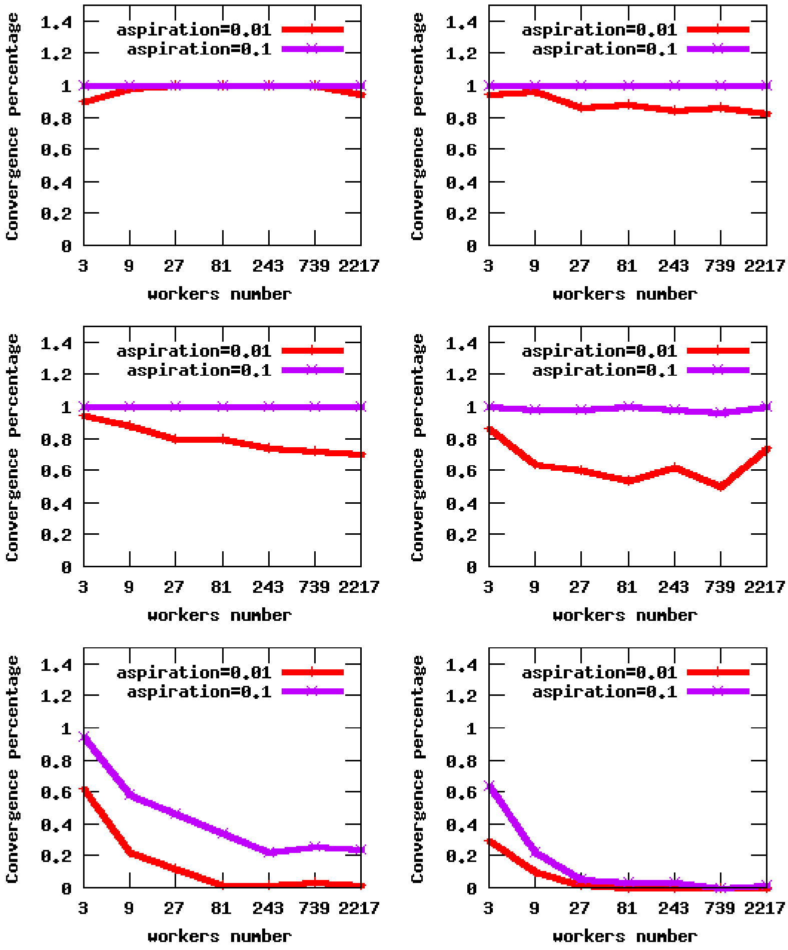

Fig. 4 Convergence percentage of 50 realizations in 1000 rounds as a function of the number of workers for the parameters of the top let panel of Fig. 3, i.e., $WP_{\mathcal{C}} = 0$, $WC_{\mathcal{T}} = 0.1$, $\alpha = 0.1$, $p_{\mathcal{A}}^{min} = 0.01$, with either $a_i = 0.01$ (red lines) or $a_i = 0.1$ (purple lines). From top to bottom and from left to right, tolerance values are $\tau = 0.7, 0.75, 0.775, 0.8, 0.9, 0.95$. Lines are only a guide to the eye.

From the viewpoint of the payoff related parameters, i.e., those more in the spirit of the principal-agent theory of studying how binding contracts should be made, we have found that quite generally the convergence of the system to eventual correctness, to providing reliable answers, is quite fast. One factor that improves this convergence is punishment. In turn, this bears an interesting influence on the effects of the aspiration level: Without punishment, having workers with higher aspiration levels leads to faster convergence, whereas the opposite is true when there is punishment. This is expected from the bounds found in Thm. 2 and confirmed by the simulations.

On the other hand, strictly speaking punishment is not necessary to ensure convergence, as the analytical results we summarized in Sec. 3 already hinted to. Therefore, it is left to the master's choice whether she would rather have slower convergence and no punishment or faster convergence with punishing. In the case in which punishing is not costly it seems that the rational thing to do is to punish, but if costs are incurred then one has to compare with those of auditing in order to make a decision. As for the computing costs for the worker, they have more natural consequences and we have seen that increasing costs would increase the convergence time in turn.

Simulations have also allowed to look in detail into the issue of tolerance, something about which our analytical results do not provide much information. As we have seen, the system is able to perform correctly and provide the right answer for high tolerances, implying that the master needs only to react to large majorities of defectors or cheaters. In fact, using punishment may push the tolerance levels for which convergence to eventual correctness is always guaranteed up to values as large as $\tau = 0.85$, meaning that the master would only need to increase her auditing probability when almost all workers are cheating. This is a very good result not only in terms of the convergence of the system but also as regards the cost, because lower auditing probabilities means less actual audits and therefore less incurred costs by the master. On the other hand, the tolerance behavior above the convergence threshold depends on the number of workers participating in the crowd computing process. As we have seen, the performance of the system decreases with the number of workers for both aspiration values studied, the decrease being slower (but starting at lower tolerances) for the lowest aspiration level. This observation points to the need of balancing the requirements of the task to be carried out in terms of computing power (number of workers) with the reliability of the process (convergence percentage). Keeping everything else constant, increasing the number of workers may lead to a serious degrading of the results, so if more computers are required it is probably necessary to lower the tolerance to keep the worker population in check. Finally, an interesting question arises regarding the nature of the transition that takes place in the convergence percentage as a function of the tolerance and the aspiration level. Our simulations correspond only to a small set of parameters and we cannot resolve the transition region in detail, and on top of that we would need more realizations and certainly a much larger range of sizes. This remains an interesting open problem which is most relevant for the applicability of the algorithm: Indeed, if the transition is abrupt one has to be careful and work with tolerances away from that region, as any deviation in the parameters may take the system into the region where it never works properly in practice. If, on the contrary, the transition is smooth, then the value of the tolerance at the transition is not that critical as small changes of the parameters would only lead to small decreases of the performance of the algorithm.

To conclude the paper, a few comments are in order regarding its present scope and its possible extensions. The way we have presented our model here, it is clear that we are proposing a scheme in terms of independent principal-agent relationships, with no interactions between workers whatsoever. However, when we introduce the dynamics, the dependence on the total number of cheaters introduces a global mean field type interaction among the workers that couples their behavior as in a majority model. In this context, we have also a new feature with respect to majority models, namely the fact that the master reacts to this mean field coupling by readjusting how often she looks into the results she receives from the workers. These two effects combine in a manner that adds quite some complication to the problem of independent agents, and features such as the interplay of the worker's aspirations and the master's possibility to punish are an example of this nontrivial dynamics. Notwithstanding, being already a rich model, what we have introduced is a baseline scenario in which there are no direct couplings between the workers.

It is interesting to note that, to some extent, the generalization to a situation where workers interact on a network is, in certain sense, going from pairwise interactions to group interactions. Therefore, understanding group interactions may be a central premise to then go on to crowds. In this regard, several relevant aspects have already been noted in previous works. Thus, possible noise-dependences (such as mistakes) change when going to groups depending on the network topology [60]. The size of groups may also have effects on the evolution of cooperation [61]. In addition, group heterogeneity may also affect the outcome of the process [62] and, as a matter of fact, a correct description of the group interactions turns out to be crucial for the formulation of the problem [63–65].

Another possible extension of our model has to do with the workers' aspiration level. In the current formulation, workers have a fixed aspiration level, but this needs not be the case, as it has been proposed for the Prisoner's Dilemma by Macy and Flache [55] to study the effects of habituation. They modeled habituation as an aspiration level that changes following repeated stimulus: If the payoff exceeds aspirations, the aspiration level increases, leading to satiation on reward and sensitization to punishment. If the payoff falls below aspirations, the aspiration level decreases, leading to sensitization on reward and desensitization to punishment. Macy and Flache found that such dynamic aspiration levels may lead to worse results in terms of promotion of cooperation. In our case, we would expect that, in agreement with our discussion above, increasing aspiration levels may lead to an improvement in the system performance; however, this may lead to trouble in the mid- to long-term because, on one hand, it will increase the cost for the master and, on the other hand, workers may eventually reach aspiration levels that are hard (if not impossible) to meet. When the master covers only a majority of workers this might self-regulate, as workers with large aspiration levels are replaced by others with smaller aspirations, but we cannot exclude that there may be instabilities that lead to a complete breakdown of the process, at least for certain habituation dynamics as observed in [55]. It is thus clear that this is an issue that deserves further investigation.

In many applied contexts in crowd computing and elsewhere, this needs not be the case and in fact workers may know each other or even, if we think of an organization, be connected through a social network. Therefore, the question of explicit collusions immediately arises, but beyond that, we should also consider different informational contexts. Indeed, if workers have some relationship, they may observe the actions of each other, and perhaps the received payments, and that could influence their own behavior. Aspiration levels could also be public information or at least inferred from the payments a player receive. On the other hand, the master can introduce bookkeeping, for instance, by assigning a reputation to each worker (that can in turn be public or private). In fact, preliminary work shows that, again as in the general problem of the emergence of cooperation, introducing reputation improves very much the performance of the system. In any event, it is apparent that while the results we are presenting here are certainly interesting, there is a lot more to do in order to bring the model closer to specific real life applications. In this respect, it is worth noting that our model assumes the dynamics of the master and the agents' strategy updates. As has been shown for the case of the Prisoner's Dilemma [27,26,66,67], it can perfectly be the case that the way people behave in this kind of situation is not that we are assuming. Therefore, following the usual physics approach to a problem, specifically designed experiments where the dynamics of actual human subjects could be observed are in order to assess the extent of the validity of our work. In the same manner, specific predictions regarding, for instance, the aspiration level, could be also tested. We hope that this work stimulates further research along these directions, some of which we are currently undertaking.

Acknowledgements We thank Carlos Diuk for useful discussions.

References

1. P. Ball, *Critical Mass: How One Thing Leads to Another.* Arrow Books, London (2005).
2. R. N. Mantegna, H. E. Stanley, *An Introduction to Econophysics: Correlations and Complexity in Finance.* Cambridge University Press, Cambridge (1999).
3. C. Castellano, S. Fortunato, and V. Loreto, Statistical physics of social dynamics, Rev. Mod. Phys. **81**, 591-646 (2009).
4. P. W. Anderson, More is different, Science **177**, 393–396 (1972).
5. N. Goldenfeld, *Lectures on Phase Transitions and the Renormalization Group.* Addison-Wesley, Boston (1992).
6. T. C. Schelling, *Micromotives and Macrobehaviour.* W. W. Norton, New York (1978).
7. D. Stauffer and S. Solomon, Ising, Schelling and self-organising segregation, Eur. Phys. J. B **57**, 473–479 (2007).
8. L. Blume, The statistical mechanics of strategic interaction, Games Econ. Behav. **5**, 387–424 (1993).
9. G. J. Stigler, The Development of Utility Theory: I, II, J. Polit. Econ., **58,** 307–327, 373–396 (1950).
10. E. Goffman, *Strategic Interaction.* University of Pennsilvania Press, Philadelphia (1969).
11. J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior.* Princeton University Press, Princeton (1944).
12. J. Maynard-Smith, *Evolution and the Theory of Games.* Cambridge University Press, New York/London (1982).
13. H. Gintis, *The Bounds of Reason: Game Theory and the Unification of the Behavioral Sciences.* Princeton University Press, Princeton (2009).
14. F. Vega-Redondo, *Game Theory and Economic Applications.* Cambridge University Press, Cambridge (2003).

15. H. Gintis, *Game Theory Evolving, 2nd edition.* Princeton University Press, Princeton (2009).
16. J. Nash, Equilibrium points in n-person games, Proc. Natl. Acad Sci. USA **36** 48–49(1950).
17. J. C. Harsanyi and R. Selten, *A General Theory of Equilibrium Selection in Games.* MIT Press, Cambridge (1988).
18. J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics.* Cambridge University Press, Cambridge (1988).
19. J. Hofbauer and K. Sigmund, Evolutionary Game Dynamics, Bull. Amer. Math. Soc. **40**, 479–519 (2003).
20. M. A. Nowak, *Evolutionary Dynamics.* Harvard University Press, Harvard (2006).
21. P. Taylor and L. Jonker, Evolutionarily stable strategies and game dynamics, Math. Biosci. **40**, 45–156 (1978).
22. C. Darwin, *The Descent of Man, and Selection in Relation to Sex.* J. Murray, London, 1871.
23. P. Kollock, Social dilemmas: The anatomy of cooperation, Annu. Rev. Sociol. **24**, 183–214 (1998).
24. M. A. Nowak, Five rules for the evolution of cooperation, Science **314**, 1560–1563 (2006).
25. C. P. Roca, J. A. Cuesta, and A. Sánchez, Evolutionary game theory: temporal and spatial effects beyond replicator dynamics, Phys. Life Rev. **6,** 208–249 (2009).
26. J. Grujić, C. Fosco, L. Araújo, J. A. Cuesta, and A. Sánchez, Social experiments in the mesoscale: Humans playing a spatial Prisoner's Dilemma, PLOS ONE **5**, e13749 (2010).
27. C. Gracia-Lázaro, A. Ferrer, G. Ruiz, A. Tarancón, J. A. Cuesta, A. Sánchez, and Y. Moreno, Heterogeneous networks do not promote cooperation when humans play a Prisoner's Dilemma, Proc. Natl. Acad. Sci. USA **109**, 12922–12926 (2012).
28. G. Szabó, and G. Fáth, Evolutionary games on graphs, Phys. Rep. **446**, 97–216 (2007).
29. M. Perc, and A. Szolnoki, Coevolutionary games — A mini review, Biosystems **99**, 109–125 (2010).
30. R. Rees, The Theory of Principal and Agent: I, II, Bull. Econ. Res. **37,** 3–26, 75–97 (1985).
31. A. Mass-Colell, M. Whinton, and J. Green, *Microeconomic Theory.* Oxford University Press, Oxford (1995).
32. D. Anderson, BOINC: A system for public-resource computing and storage. Proceedings of GRID 2004, 4–10 (2004).
33. E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky, SETI@home: Massively distributed computing for SETI. Comp. Sci. Eng. **3** 78–83 (2001).
34. Ibercivis Citizen Computing Platform, `http://www.ibercivis.net`.
35. Amazon's Mechanical Turk, `https://www.mturk.com`.
36. M. Babaioff, M. Feldman, and N. Nisan, Combinatorial agency. Proceedings of ACM EC 2006, 18–28 (2006).
37. M. Babaioff, M. Feldman, and N. Nisan, Mixed strategies in combinatorial agency. Proceedings of WINE 2006, 353–364 (2006).
38. M. Babaioff, M. Feldman, and N. Nisan, Free riding and free labor in combinatorial agency. Proceedings of SAGT 2009, 109–121 (2009).
39. R. Eidenbenz and S. Schmid. Combinatorial agency with audits. Proceedings of GameNets 2009, 374–383 (2009).
40. P. Golle and I. Mironov, Uncheatable distributed computations. Proceedings of CT-RSA 2001, 425–440 (2001).
41. E.M. Heien, D.P. Anderson, and K. Hagihara, Computing low latency batches with unreliable workers in volunteer computing environments. J. Grid Computing **7**, 501–518 (2009).
42. J. Shneidman and D.C. Parkes, Rationality and self-interest in P2P networks. Proceedings of IPTPS 2003, 139–148 (2003).

43. I. Abraham, D. Dolev, R. Goden, and J.Y. Halpern, Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. Proceedings of PODC 2006, 53–62 (2006).
44. L. Sarmenta, Sabotage-tolerance mechanisms for volunteer computing systems. Future Generation Computer Systems **18**, 561–572 (2002).
45. A. Fernández Anta, Ch. Georgiou, and M. A. Mosteiro, Designing mechanisms for reliable Internet-based computing. Proceedings of NCA 2008, 315–324 (2008).
46. A. Fernández Anta, Ch. Georgiou, and M. A. Mosteiro, Algorithmic Mechanisms for Internet-based Master-Worker Computing with Untrusted and Selfish Workers. Proceedings of IPDPS 2010, 1–11 (2010).
47. R. Boyd and P. J. Richerson, Punishment allows the evolution of cooperation (or anything else) on sizable groups. Ethol. and Sociobiol. **13**, 171–195 (1992).
48. E. Fehr and S. Gächter, Altruistic punishment in humans. Nature **415,** 137–140 (2002).
49. D. Rose, T. R. Willemain. The principal-agent problem with evolutionary learning. Computational and Mathematical Organization Theory, **2**, 139–162 (1996).
50. R. R. Bush and F. Mosteller, *Stochastic Models for Learning.* Wiley, New York (1955).
51. F. G. Cross. *A theory of Adaptive Economic Behavior.* Cambridge University Press, Cambridge (1983).
52. I. Erev and A. Roth, Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. Am. Econ. Rev. **88**, 848–881 (1998).
53. C. F. Camerer, *Behavioral game theory: Experiments in strategic interaction.* Princeton University Press, Princeton (2003).
54. J. Duffy, Agent-Based Models and Human Subject Experiments, In: Handbook of Computational Economics II: Agent-Based Computational Economics, edited by L. Tesfatsion and K. L. Judd, chapter 19, 949–1011. Elsevier, Amsterdam, 2006.
55. M. W. Macy and A. Flache. Learning dynamics in social dilemmas, Proc. Natl. Acad. Sci. USA **99**, 7229–7236 (2002).
56. S. S. Izquierdo, L. R. Izquierdo and N. M. Gotts, Reinforcement Learning Dynamics in Social Dilemmas. J. Artifi. Societ. Soc. Simul. **11**, 1 (2008).
57. L. R. Izquierdo and S. S. Izquierdo. Dynamics of the Bush-Mosteller learning algorithm in 2×2 games, in *Reinforcement Learning. Theory and Applications*, edited by C. Weber, M. Elshaw and N. M. Mayer, I-Tech Education and Publishing, Vienna (2008).
58. E. Christoforou, A. Fernández Anta, C. Georgiou, M. A. Mosteiro, and A. Sánchez, Achieving Reliability in Master-Worker Computing via Evolutionary Dynamics. Proceedings of Euro-Par 2012 (2012).
59. C. Szepesvári, *Algorithms for Reinforcement Learning.* Morgan & Claypool Publishers, San Rafael (2010).
60. A. Szolnoki, M. Perc, and G. Szabó, Topology-independent impact of noise on cooperation in spatial public goods games, Phys. Rev. E **80**, 056109 (2009).
61. A. Szolnoki, and M. Perc, Group-size effects on the evolution of cooperation in the spatial public goods game, Phys. Rev. E **84**, 047102 (2011).
62. M. Perc, Does strong heterogeneity promote cooperation by group interactions? New J. Phys. **13**, 123027 (2011).
63. J. Gómez-Gardeñes, M. Romance, R. Criado, D. Vilone, and A. Sánchez, Evolutionary games defined at the network mesoscale: The Public Goods Game, Chaos **21**, 016113 (2011).
64. J. Gómez-Gardeñes, D. Vilone, and A. Sánchez, Disentangling social and group heterogeneities: Public Goods games on complex networks, EPL **95**, 68003 (2011).
65. J. Peña, and Y. Rochat, Bipartite graphs as models of population structures in evolutionary multiplayer games, PLOS ONE **7**, e44514 (2012).
66. A. Traulsen, D. Semmann, R. D. Sommerfeld, H. J. Krambeck and M. Milinski, Human strategy updating in evolutionary games. Proc. Natl. Acad. Sci. USA **107,** 2962–2966 (2010).
67. D. Semmann, Conditional cooperation can hinder network reciprocity, Proceedings of the National Academy of the USA **109**, 12846–12847 (2012).