An Information-Theoretic Analysis of Memory Bounds in a Distributed Resource Allocation Mechanism

Ricardo M. Araujo and **Luis C. Lamb**
Instituto de Informática, Universidade Federal do Rio Grande do Sul
91501-970, Porto Alegre, Brazil
rmaraujo@inf.ufrgs.br, LuisLamb@acm.org

Abstract

Multiagent distributed resource allocation requires that agents act on limited, localized information with minimum communication overhead in order to optimize the distribution of available resources. When requirements and constraints are dynamic, learning agents may be needed to allow for adaptation. One way of accomplishing learning is to observe past outcomes, using such information to improve future decisions. When limits in agents' memory or observation capabilities are assumed, one must decide on how large should the observation window be. We investigate how this decision influences both agents' and system's performance in the context of a special class of distributed resource allocation problems, namely dispersion games. We show by numerical experiments over a specific dispersion game (the Minority Game) that in such scenario an agent's performance is non-monotonically correlated with her memory size when all other agents are kept unchanged. We then provide an information-theoretic explanation for the observed behaviors, showing that a downward causation effect takes place.

1 Introduction

An important class of natural problems involving distributed resource allocation mechanisms may be modeled by *dispersion games* [9]. Problems in this class are of anti-coordination nature: agents prefer actions that are not taken by other agents. Stock exchange and load balancing in networks are instances of such problems [8]. In iterated versions of such games learning is an important component of the decision-making process. While each agent may try and learn individual behaviors of all agents, this is a complex task especially when the total number of agents in the system is unknown, not constant or very large. A possible solution is to consider other agents' actions as indistinguishable from the environment - an idea behind the so-called 0-level agents [17] which, for some scenarios, yields the best results [10]. In such cases, only the joint actions of all agents are considered and they may be represented as a single global outcome. The computational cost of analyzing one single global outcome is expected to be much smaller than that of analyzing multiple outcomes produced by individual agents. More importantly, such a cost does not depend on the number of agents, and its computation may be feasible even without knowledge of such number.

In repeated dispersion games, agents may learn from a series of past outcomes in order to improve their decisions. A natural issue is how large should the observation window be. While bounds in agents' memory or observation capabilities are often assumed [14], there is a paucity of explanation about how such limits are established. In this paper we tackle this problem in a specialized version of a dispersion game known as the Minority Game (MG) [4]. In the MG, a finite number of agents decide over a (also finite) set of actions and those who choose the action chosen by the smallest number of agents are rewarded. When the game is repeated, agents have the chance to learn from previous rounds and improve their performance by considering a finite window of past outcomes (they have finite memory) in order to try and learn temporal patterns.

Our contribution is twofold. First, we systematically simulate cases in the MG composed of agents with different memory sizes and compare their performances to a fixed class of (other) agents with fixed memory sizes, using two different learning algorithms. The results show a clear non-monotonic relation between access to information and agent's performance, unveiling cases where more information translates into higher gains and cases where it is harmful. Further, we relate those findings to the system's global efficiency state. Second, we provide an information-theoretic explanation to the observed behaviors. We show that the size of the generated outcome patterns does not always match the system's memory size, thus allowing agents with larger memory sizes to exploit this information.

The remainder of the paper is organized as follow. We start by discussing relevant related work. We then present the MG model and terminology. Next, we detail our methodology and the results of the experimental studies using two different learning algorithms. Finally, we analyze the results and point out directions for future research.

2 Related Work

While previous works have focused on problems relating homogeneous memory size to global system performance, little effort has been put into understanding the relation between individual memory size and individual agent's performance.

Homogeneous agents involved in self-play are typically assumed, with emphasis in system dynamics rather than individual agents. However, as argued in [15], a more appropriate agenda for multi-agent learning "asks what the best learning strategy is for a fixed class of the other agents in the game", as this allows the construction of effective agents. We follow this methodology here. In the context of the MG, little work so far has been directed to and concerned with individual agent's performance. An exception to that is [16], in which a genetic algorithmic-based agent is shown to outperform traditional ones, but little investigation about the conditions on why this happens is done. Other learning algorithms have been applied to the MG [1, 2, 5]; however, homogeneous agents are assumed with focus on global properties.

Heterogeneous memory sizes are studied in [11], where an evolutionary setup is used to search for the best memory size for the game; however, they do not fix a class of agents, allowing every agent to evolve independently. A similar setup is used in [5], but their concern is *average* memory size. We shall provide alternative explanation for some of their results. In [6] it is stated that agents with larger memories perform better than agents with smaller ones. Nonetheless, they consider a varying number of agents endowed with only one extra bit of memory; we will show that having larger memories may not lead to better performance.

3 A multiagent market scenario

The Minority Game (MG) was initially proposed in the context of *Econophysics*, as a formalization of the El Farol Bar Problem [3]. In these models, the main interest is the type of information structure created by multiple interacting bounded rational agents. In the MG, there are two groups and an odd number (N) of agents. At each round, agents must choose concurrently and without communication to be in only one of the two groups. At each round t, after all decisions are made, a global outcome is calculated: $A(t) = \sum_{i=0}^{N} a_i(t)$, where $a_i(t)$ is the decision of agent i in round t and $a_i(t) \in \{+1, -1\}$. Agents' wealth is distributed according to $w_i(t) = w_i(t-1) + a_i(t) sign(A(t-1))$, where $sign(\cdot)$ is $+1$ or -1 according to $A(t-1)$ being positive or negative (other functions of $A(t)$ may be used). Initial values are set to zero. $A(t)$ encodes which group had the *minority* of agents and only those in this group are rewarded, while those in the majority are penalized. There is no equilibrium point in which all agents profit, and the expectation of the majority is always frustrated due to the anti-coordination nature of the game: if many believe a particular group will be the minority one, then they will actually end up in the majority. In spite of its simplicity, the MG offers a rich and complex behavior that was shown to represent several scenarios of decentralized resource allocation, such as financial markets and traffic commuters scenarios [4]. In [9], this model was generalized and shown to belong to a broader class of *dispersion games*.

One of the main concerns in the MG is with the game's global efficiency, i.e. the number of agents winning the game. From the viewpoint of the system as a whole, one is interested in maximizing the number of winners, so that resources are better distributed. When the MG was originally introduced, agents were modeled using a simple inductive learning algorithm to make decisions, corresponding to a simplified model of how human decisions in complex environments are actually made [3]. Since this algorithm has been widely used and studied in other applications, it is of our interest to make use of it in a straightforward way (we refer to it as the *traditional learning algorithm*). In this algorithm, each agent has access to the m last outcomes of the game and she is given a set S of hypotheses, randomly selected from a fixed hypotheses space H. A hypothesis maps all possible patterns of past outcomes over m to a decision (i.e. what group to choose in the next round). m is known as the *memory size* of the game. A hypothesis is effectively represented by a binary string of length 2^m and $|H| = 2^{2^m}$. The learning algorithm keeps a measure $f_{i,j}(t)$ of the effectiveness of each hypothesis j held by agent i in time t, which is updated by $f_{i,j}(t) = f_{i,j}(t-1) + a_i sign(A(t-1))$. This measure is often called *virtual points*. An agent then uses her hypothesis with the maximum number of virtual points $argmax_j f_{i,j}(t)$ to commit to a decision in the next round of the game. Ties are broken by coin toss. This algorithm is very simple and provides more adaptation than learning, since agents are unable to better explore strategies outside their initial set. In order to overcome the limitations of this algorithm and to allow for comparisons with a different learning strategy, we shall introduce a new evolutionary-based learning algorithm.

4 Methodology and Experiments

Due to the nature of our interest in the model, we consider agents with different hypotheses space sizes (i.e. different memory sizes), in order to observe the effects of such differences in agents' performance[1]. We split the MG into two parts. The *environment* consists of a traditional MG with a set N of agents, as described above (with homogeneous agents). The *control group* G is a group of agents which sample hypotheses from a different space from that of the agents in the environment, differing only in its size $|H|$. Since hypotheses space size in our setup is defined only by each agent's memory size, we shall denote the environment's by m_e and the control group's by m_g.

In this work we assume $|G| = 1$. [6] investigates larger groups in similar fashion; however in their experiments the memory size is not systematically modified and their interest is on the effects of changes in group sizes. We will observe the performance for the resulting *control agent* when her hypotheses space size m_g varies in relation to m_e. We also want to verify how her performance (defined by the average wealth w_g) changes with m_g and compare it to the environment's performance (defined by the average wealth w_e of a random agent in the environment). In order to do so, we measure the control agent's *gain*: w_g/w_e. All results are averages over 200 randomly initialized games and they have been statistically tested for significance using a standard t-test with 95% confidence interval. Each game was run for 100,000 rounds and we have used the standard values found in the literature for all other parameters, namely: $|N| + |G| = 101$

[1] When dealing with homogeneous agents, hypotheses are drawn from the same hypotheses space.

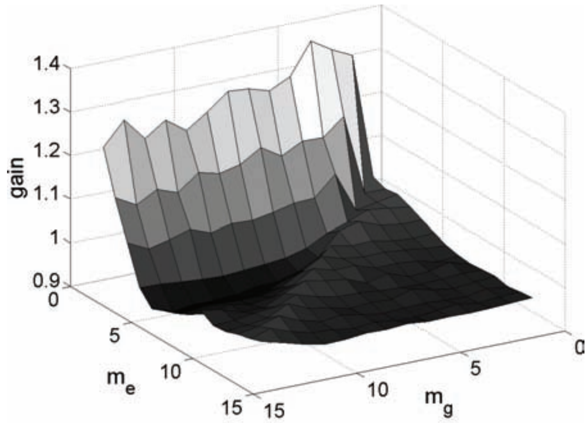

Figure 1: Control agent gains for combinations of m_e and m_g. Lighter shades denote higher gains.

and $|S| = 2$, so comparisons could be easily made [2]. We shall detail experiments using the traditional and the evolutionary learning algorithms.

4.1 Traditional Learning Algorithm

We start with the traditional learning algorithm in both the control agent and agents in the environment. In Fig. 1 we show the control agent's gain for every combination of $m_e \in [1, 15]$ and $m_g \in [1, 15]$. These ranges may seem somewhat limited, but are adequate for our purposes. The exponential increase of hypotheses space with m and the large number of agents often makes experimentation with larger memories intractable. In the resulting topography, we observe some interesting behaviors. For $m_e \leq 3$, higher than unit gains are obtained whenever $m_g > m_e$. On the other hand, for values immediately above $m_e = 3$ we can observe that whenever $m_g > m_e$, w_g falls below unit, showing that the target agent does *worse* by having a larger hypotheses space. For larger m_e, w_g is always below unit except where $m_g = m_e$, when the gain is, as expected, exactly unitary.

Let us now observe in detail what happens in specific points of the observed regions by sectioning the topography (see Fig. 2). For the first case, we take $m_e = 3$. We can see that our control agent benefits from a larger hypotheses space. Interestingly, having *smaller* spaces seem to cause no harm and the agent performs as if $m_g = m_e$. We can also observe that the transition from one of the above cases to another is quite sharp and further increases in m_g provide no additional gains. Instead, a logarithmic decrease of wealth may be observed with further increases in m_g. The highest gain occurs precisely where $m_g = m_e + 1$, where a spike may be observed. As for the second case, we detail the behavior for $m_e = 6$. We observe a logarithmic drop in w_g for $m_g > m_e$. For $m_g < m_e$, a small decrease of gains can be observed. Thus, in this region no m_g does better than unit and, interestingly, having access to a larger input window is harmful to the agent. For larger values of m_e, losses become larger whenever $m_g < m_e$. Having access to a larger information window is harmful to the agent, except when the environment has small memory sizes. To observe such phenomena, we plot the control agent's gain when m_e varies and we set $m_g = m_e + 1$. This is shown in Fig. 3, where it is made clear that there is a non-monotonic relation between memory size and agent's performance. We observe that gains are high for small values of m_e, then they become smaller than unit, reaching a minimum and later increasing again to finally converge to a value slightly below unit. Having worse performance when accessing more information may seem counter-intuitive. One could argue that this is due to the larger hypotheses space, which makes finding a good hypothesis harder. Even though this may be part of the explanation, it does not account for all of it since we observe a non-linear relationship between gains and hypotheses space size, as we shall further investigate.

[2]It is known that larger S does not change the qualitative behavior of the system [4]

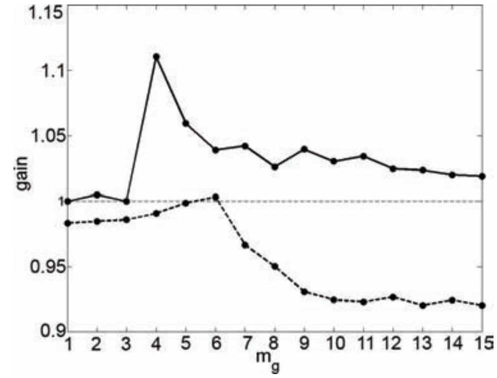

Figure 2: Control agent gain for $m_e = 3$ (solid line) and $m_e = 6$ (dashed line) with agents using the traditional learning algorithm

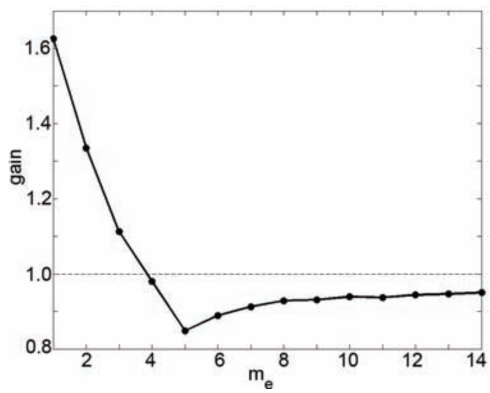

Figure 3: Control agent gain for varying m_e and $m_g = m_e + 1$

4.2 Evolutionary learning

It could be argued that the observed behavior is only but a peculiar effect of the learning algorithm used, whose main limitation is the inability to explore the hypotheses space during the game, i.e. an agent may only use the hypotheses given at the start of a run. In order to address this concern, we have repeated the experiments using an evolutionary-based learning algorithm so as to allow agents to further explore the hypotheses space. Some different evolutionary-based algorithms for the MG have been proposed [5, 11, 16]. We chose an adaptation of the one proposed in [2], due to its simplicity and good

```
Algorithm Evol-Learning
    w ← 0;
    S ← random hypotheses ∈ H;
    foreach s ∈ S do
    |   fitness(s) ← 0;
    end
    while not end of the game do
        obs ← window of size m of past outcomes;
        h ← arg max_s fitness(s);
        decision ← decision of h using observation obs;
        commit(decision);
        if decision = outcome then  w ← w + 1;
        foreach s ∈ S do
            if decision of s using obs = outcome then
            |   fitness(s) ← fitness(s) + 1;
            else
            |   fitness(s) ← fitness(s) - 1;
            end
        end
        with probability p_r do
            worst hypothesis s ← best hypothesis k;
            with probability p_m, flip bits in s;
        end
    end
end
```

Figure 4: Evolutionary Learning Algorithm

performance. While we could have applied a more elaborated algorithm used in similar games, such as [7], we have chosen to create an algorithm that is closely related to the traditional one, but presenting better learning characteristics. By doing so we are able to better understand and compare results obtained using both algorithms.

In this algorithm (depicted in Fig. 4), each agent starts with S hypotheses and, at every round, with probability p_r, she discards her worst performing hypothesis and replaces it with a copy of her best performing one. Each bit of this copy is then flipped with probability p_m. This allows agents to search for better hypotheses, continually introducing new ones to the game. It is interesting to note that, differently from other evolutionary learning algorithms such as [16], this one does not require a global ordering of agents based on their performances. Agents retain their autonomous characteristics by not relying on a central authority to decide which hypotheses among all agents are to be replaced, thus preserving the distributed nature of the game. Figure 5 shows $A(t)$ for a typical run when agents are using the new proposed learning algorithm and $p_r = 0.01$ and $p_m = 0.001$. Clearly, learning is taking place (when using the traditional algorithm, no decrease in oscillations is observed, even for very long runs). For all experiments using this algorithm we consider only results after 2,000 rounds, in order to observe the "steady" state of the system. Figure 6 shows the control agent's gain for every combination of $m_e \in [1, 15]$ and $m_g \in [1, 15]$ when agents use the evolutionary-based learning algorithm. We now observe a different behavior when compared to our previous case. There are no regions where larger memories are beneficial, each extra bit beyond the environment's memory size is harmful to the target agent. A logarithmic decrease with m_g is observed for $m_g > m_e$, for all tested values of m_e. In Fig. 7 we detail the behavior for some values of m_e.

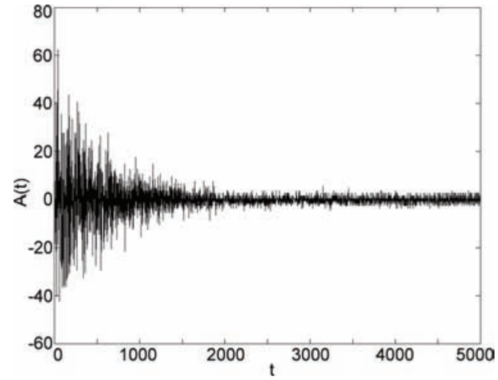

Figure 5: $A(t)$ for agents using evolutionary-based algorithm

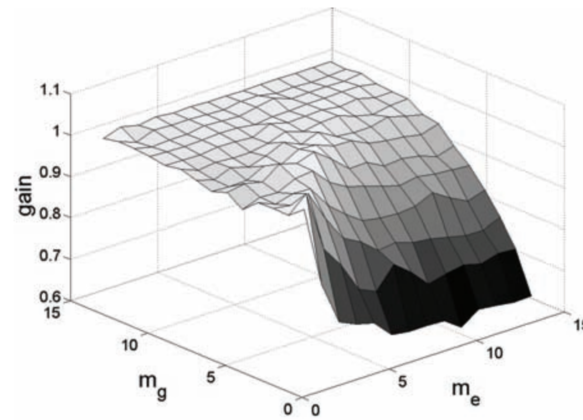

Figure 6: Control agent gains for combinations of m_e and m_g. Lighter shades denote higher gains.

As with the traditional learning algorithm, for $m_g < m_e$ no considerable losses or gains are observed.

5 Analyzing the Results

5.1 Dynamics of the game and its efficiency regions

As stated above, one of the main concerns with the MG is its global efficiency, i.e. how many agents are winning the game during a run. A typical way of measuring the temporal efficiency of resource distribution is by means of the statistical variance σ^2 of $A(t)$ [4]. The larger the variance is, the larger the waste of resources, making the system less efficient. The variance σ^2 is a function only of the number of agents in the game and their (homogeneous) memory size [13]. Since we keep the number of agents fixed and the control group is unitary, we consider σ^2 only as a function of m_e, the memory size of agents within the environment. Figure 8 shows σ^2 as a function of m_e when all agents use the traditional learning algorithm. The same plot is observed for any m_g. Three regions of efficiency can be observed. For small m_e, high variance is observed, thus low efficiency characterize the system. For large m_e, the system has precisely the variance expected if all agents were deciding randomly (i.e. the Random Case Game - RCG). Intermediate values of m_e are correlated with smaller, better than random, variances. This last case is often the main subject of interest in the MG, since it indicates that agents are able to self-organize to improve efficiency.

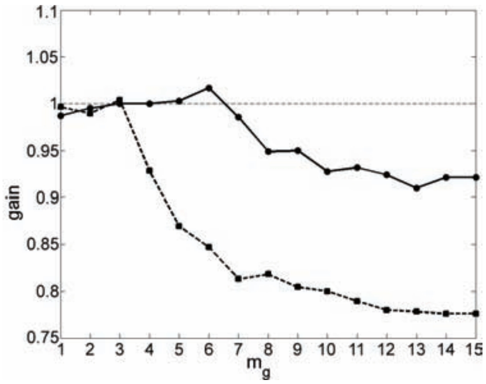

Figure 7: Target agent's gain versus m_g for $m_e = 3$ (solid line) and $m_e = 6$ (dashed line), using the evolutionary learning algorithm

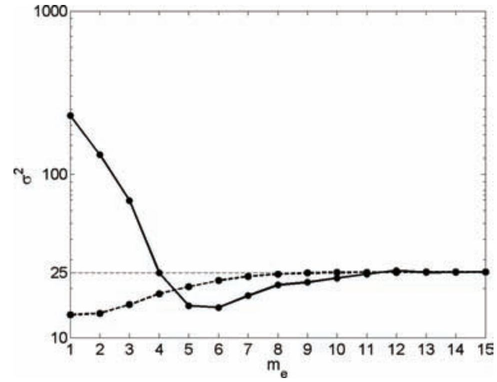

Figure 8: σ^2 as a function of m_e for the traditional (solid line) and the evolutionary algorithm (dashed line)

Comparing Fig. 8 with the plotted gain using the traditional algorithm (Fig. 3), we observe that higher than unit gains are associated with regions with high variances, gains below unit are associated with regions with small variance and increases in gains follow increases in variance. This indicates a correlation between system's global efficiency and exploitation possibilities of individual agents, where larger memories are beneficial when the system is behaving inefficiently (worse than random) whereas when it is efficient having larger memories becomes harmful. The same behavior is observed in the evolutionary setup. We can also observe in Fig. 8 that the variance curve is quite different from the one using the traditional algorithm - there is a smooth transition between a very low variance and the expected for the RCG. Since there is no inefficient region we would expect, following analogous reasoning, that no agent with $m_g > m_e$ would perform better than the agents in the environment and this was actually observed (see Fig. 6). Such results are interesting, since they relate the efficiency of the system as a whole with individual exploitation possibility (by means of larger memories) and indicate the existence of stability points in the system's efficiency regions. For instance, if we take an evolutionary version of the game where all agents start with a small memory and are allowed to increase or decrease their memory size during a run, we would expect that there would be an initial incentive towards larger memories, leading the system towards higher efficiency points. However, such incentive would stop when the system reaches an efficient point as larger memories then become harmful. Thus, this efficient point would be a stability point. Such experiment was conducted in [5] where initial memory growth was observed, halting at the predicted memory size. Such behavior was attributed to the simple nature of the game. Here we propose a more detailed, distinctive explanation relating individual agent gain to the efficiency region of the system. This is a case of *downward causation*, where game dynamics are initially fixed by agents but such dynamics end up fixing possible (or profitable) agents' behaviors.

5.2 Mutual Information

In [13] homogeneous memory sizes and the traditional learning algorithm are used to show that there is different *information* available to the agents for different memory sizes. They have measured this information by the conditional probability $P(1|\mu_k)$ - the probability of having a "1" following a binary sequence μ_k of length k. They have shown that the inefficient region is actually *informationally efficient*, in the sense that $P(1|\mu_k)$ is exactly 0.5 for all μ_k whenever all agents have memory of size k. On the other hand, in the efficient region there is an asymmetry in the probability distribution and there are μ_k that offer predictive power. The information analysis in [13] focused in cases where $k = m$, i.e. the measurement of information available to agents with the same memory size as their peers. We wish to access the information available within $k \neq m$, since it represents our target agent with different memory sizes. In order to do so, we have to define a more precise information measure. To measure the asymmetry in the distribution of the predictive power of each hypothesis, we use the concept of *mutual information* [12] between a string μ_k and its outcome, computing the average information contained in each string: $I(k) = \frac{1}{|H|}\sum_{\mu_k} p_1(\mu_k) log_2\left(\frac{p_1(\mu_k)}{p_u(\mu_k)p_1}\right) + p_0(\mu_k) log_2\left(\frac{p_0(\mu_k)}{p_u(\mu_k)p_0}\right)$, where p_1 and p_0 are the probabilities of a "1" or a "0" occurring, respectively; $p_1(\mu_k)$ and $p_0(\mu_k)$ are the probabilities of "1" or a "0" immediately occurring after a string $\mu_k \in H$, respectively; $p_u(\mu_k) = p_1(\mu_k)+p_0(\mu_k)$. In the above equation, $I(k)$ is zero whenever the probabilities of observing "1" or "0" are the same for all μ_k. The highest the value of $I(k)$, the highest the average asymmetry of probabilities, indicating the existence of hypotheses with predictive power. We have measured the information by executing runs of 200,000 rounds and recording each outcome, resulting in a binary string of 200,000 symbols. Mutual information is then measured over this string and averaged over 50 independent runs.

We are now in position to measure the information available for some hypotheses of different lengths when agents are using the traditional learning algorithm. For each value of m_e, we let k assume values below, equal to, and above m_e. Figure 9 shows results for $m_e = 3$ and $m_e = 6$ (the same values detailed in the previous sections). We observe that $I(k)$ for $m_e = 3$ remains close to zero for $k \leq m_e$. Information available to memory sizes below m_e is almost the same as for m_e, which explains why agents with smaller memories do not present lower gains in Fig. 2. For $k = m_e + 1$ we observe a substantial increase in the available information and further

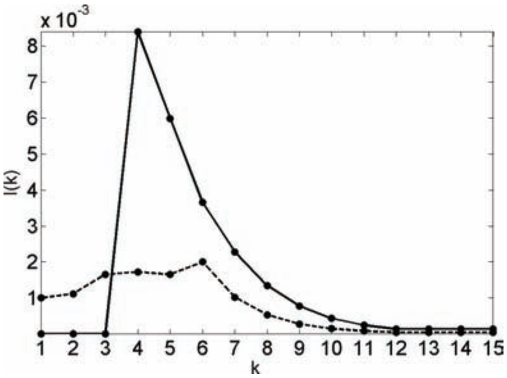

Figure 9: $I(k)$ versus k for $m_e = 3$ (solid line) and $m_e = 6$ (dashed line), using the traditional learning algorithm

increases in k lead to a logarithmic decrease in $I(k)$. This, again, is in accord with the observed behavior for the target agent's gains and is evidence that agent's interactions are creating patterns that are of greater length than their memory sizes. For $m_e = 6$, the mutual information plot also closely follows the target agent's gain plot (Fig. 2), where the highest information is present precisely at $k = m_e$. In both cases the highest value of $I(k)$ is followed by a logarithmic fall with further increases in k. This fall is expected: take k_p as the value of k associated with the highest value of $I(k)$; if we increase the hypotheses such that $k > k_p$, the informative hypotheses previously at k_p becomes more spread out through the hypotheses space. For example, if $k_p = 3$, we could have a hypothesis that detects the pattern "001" but, for $k = 4$, this pattern is found in two hypotheses - "1001" and "0001" - but only one of them may be actually happening. We conclude that mutual information provides a good explanation for the target agent's gains with different memory sizes. It is worth observing that for some cases using the traditional learning algorithm, patterns of lengths different from the agent's memory size are created - as can be inferred from the target agent's exploitation possibility when having $m_g > m_e$.

6 Conclusions

The number of game rounds considered by each agent is a central issue in several dispersion games, such as the MG. By means of a methodology including extensive simulations, we have analyzed emerging patterns resulting from the agents' interaction, relating them to the possibility of exploitation of certain setups. We have shown that having access to more information is not always an advantage and that it can actually be harmful. Experiments with the traditional learning algorithm have shown that there is a region, related to smaller memory sizes, where an agent with larger memory could exploit the system and obtain higher gains, whereas the same agent could have its gain reduced at another region. We have also measured the mutual information associated to different strings at the outcome history. We have related this information to the observed behavior, showing that agents in the system often generate patterns that are not of the same size of their memory sizes, but which can be exploited by an agent with larger memory. On the other hand, larger strings make information sparser in the space of possible patterns, resulting in decreasing agent performance when we increase her memory above an optimal size. The results presented here lead to a better understanding of the emergent patterns created from multiagent learning interactions using an information-theoretic analysis. In particular, we have provided arguments to help choosing the best memory size when designing an agent or strategy to play the MG. While it is well known that more memory is not necessarily better for collective performance, we showed that this is also true for *individual* agent performance. This result contributes for the construction of better algorithms, that may take into account the system's efficiency region when deciding between different hypotheses space to consider.

Acknowledgments: This work has been partly supported by CNPq and CAPES.

References

[1] M. Andrecut and M.K. Ali. Q learning in the minority game. *Physical Review E*, 64(067103), 2001.

[2] R. M. Araújo and L. C. Lamb. Towards understanding the role of learning models in the dynamics of the minority game. In *Proc. of 16th IEEE ICTAI 2004*, pages 727–731, 2004.

[3] W. B. Arthur. Inductive reasoning and bounded rationality. *American Economic Review*, 84:406–411, 1994.

[4] D. Challet, M. Marsili, and Y.-C. Zhang. *Minority Games*. Oxford University Press, 2004.

[5] D. Challet and Y.-C. Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246, 1997.

[6] D. Challet and Y.-C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514–532, 1998.

[7] Jacob W. Crandall and Michael A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proc. ICML 05*, pages 161 – 168. ACM Press, 2005.

[8] E. Gelenbe, E. Seref, and Z. Xu. Towards networks with intelligent packets. *Proc. IEEE-ICTAI*, 1999.

[9] Trond Grenager, Rob Powers, and Yoav Shoham. Dispersion games: general definitions and some specific learning results. In *Proc. AAAI 02*, pages 398 – 403. AAAI Press, 2002.

[10] J. Hu and M. Wellman. Online learning about other agents in a dynamic multiagent system. In *Proc. Agents-98*. ACM, 1998.

[11] Y. Li, R. Riolo, and R. Savit. Evolution in minority games ii: Games with variable strategy spaces. *Physica A*, 276, 2000.

[12] David J. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, 2003.

[13] R. Manuca, Y. Li, R. Riolo, and R. Savit. The structure of adaptive competition in minority games. *Physica A*, 282, 2000.

[14] Rob Powers and Yoav Shoham. Learning against opponents with bounded memory. In *IJCAI 05*, pages 817–822, 2005.

[15] Y. Shoham, R. Powers, and T. Grenager. On the agenda(s) of research on multi-agent learning. In *Proc. of AAAI Fall Symposium on Artificial Multi-Agent Learning*, 2004.

[16] M. Sysi-Aho, A. Chakraborti, and K. Kasti. Intelligent minority game with genetic-crossover strategies. *Eur. Phys. J. B*, 34(00234):373–377, 2003.

[17] J. M. Vidal and E. H. Durfee. Agents learning about agents: A framework and analysis. In *Multiagent Learning Workshop, AAAI97*, 1997.