Measuring Universal Intelligence in Agent-Based Systems Using the Anytime Intelligence Test

Nader Chmait[1], David L. Dowe[1], David G. Green[1], Yuan-Fang Li[1], and Javier Insa-Cabrera[2]

[1] Faculty of Information Technology, Monash University, Clayton, Australia
[2] Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Spain
nader.chmait@monash.edu,
david.dowe@monash.edu,
david.green@monash.edu,
yuanfang.li@monash.edu,
jinsa@dsic.upv.es

Abstract. This paper aims to quantify and analyze the intelligence of artificial agent collectives. A universal metric is proposed and used to empirically measure intelligence for several different agent decision controllers. Accordingly, the effectiveness of various algorithms is evaluated on a per-agent basis over a selection of abstracted, canonical tasks of different algorithmic complexities. Results reflect the different settings over which cooperative multiagent systems can be significantly more intelligent per agent than others. We identify and discuss some of the factors influencing the collective performance of these systems.

1 Introduction

Numerous studies on multiagent systems are forward analysis or reverse design approaches in the goal of maximizing a utility function. However, no formal/mathematical intelligence tests were applied in order to quantify and compare the performance of groups of collaborative agents against isolated agents - which is one of the motivations behind this paper. Furthermore, successful multiagent solutions are often the result of the agents' aggregated rewards after being evaluated over a specific setting or environment. Despite the practicality of such systems, they may fall short of being categorized as more "intelligent" than isolated agents. The main reason being that successful multiagent solutions are often the result of the agents' aggregated (sum of) rewards over a given problem, resulting in an unfair comparison with single agent solutions. Moreover, the environment in which they were evaluated is often not examined for being unbiased towards some types of agents or some levels of intelligence, nor examined for being balanced (in the sens that it has a symmetric rewarding system). The notion of collective intelligence [25] in AI is usually used to describe the intelligence emerging from collaborative multiagent systems. One of the main goals of this paper is to quantify and analyze the intelligence of artificial agent collectives in a general setting. A metric based on an information-theoretical intelligence test is proposed and then used to empirically measure intelligence across a space of grid-world problems for several different collaborative and isolated agent decision controllers. For instance, we will evaluate the performance of various algorithms on a per-agent basis over selected problems consisting of abstracted, canonical tasks of different algorithmic complexities. The outcome from our experiments might account for a range of problems in the literature as it is an abstraction of: searching for a moving target (while avoiding injury), nest selection when there is one and only one best nest, as well as pattern recognition.

The remainder of this paper is organized as follows. The next section gives a brief history on machine intelligence tests since their birth in the 1950s, and an overview on the more recent research on computational collective intelligence. We introduce our methodology in Section 3 and then describe the different agent behaviors to be evaluated in Section 4. We present our experiments and discuss our results in Section 5 by making observations on how the collective behavior of the evaluated agents influenced their scores. We conclude in Section 6 by a brief summary of what was covered in this paper and also give some ideas for future work.

2 Background

A good start to understand the history of machine intelligence would be to take a look back at the imitation game proposed by Turing in the 1950s. While this has once been accepted to be an intelligence test for machines, it has many limitations [19] as it is a test of humanity instead. The machine intelligence quotient using fuzzy integrals was presented in [3]. However determining a universal intelligence quotient for ranking artificial systems is not very practical and almost unmeasurable due to the vast non-uniformity in the performances of different types of artificial systems. Several studies [5,7] investigated the relevance of compression, pattern recognition, and inductive inference to intelligence. This was later reflected by the development of the C-test [13], which was used as an intelligence test. Inspired by the above, [15] proposed a novel definition of *universal* (machine) intelligence and three years later, a test - based on this novel definition - for evaluating (anytime universal) intelligence was put forward by [14].

Independently, a great deal of research [2,11,26,6,24,21,22,18,9] was being conducted to investigate cooperative multiagent systems. Others tried to quantify and measure the performance of these systems. For instance, [10] conducted several simulations of artificial performance tests using ideas from multi-classifiers systems in order to investigate the influence of task allocation and joint decision-making on the agents' collective performance. [12] also described tools and techniques for measuring the performance of large distributed multiagent systems.

3 Methodology

A common setting in most approaches to measuring intelligence is to evaluate a subject over a series of different problems and return a quantitative measure or score reflecting the subject's performance over these problems [14]. In artificial systems, the agent-environment framework [15] (Fig. 1a) is an appropriate representation for this matter.

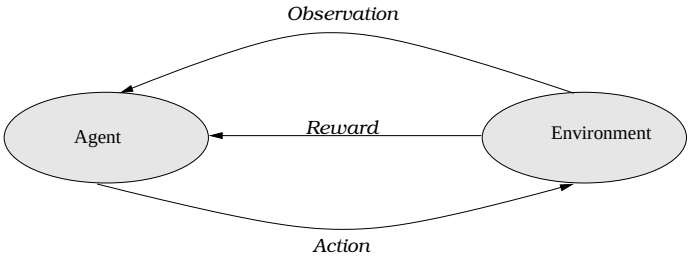
For instance, this framework allows us to model and abstract any type of interactions between agents and environments, and it also embraces the *embodiment thesis* [4] by embedding the artificial agent in a flow of observations and events generated by the environment.

3.1 Agent-Environment Framework

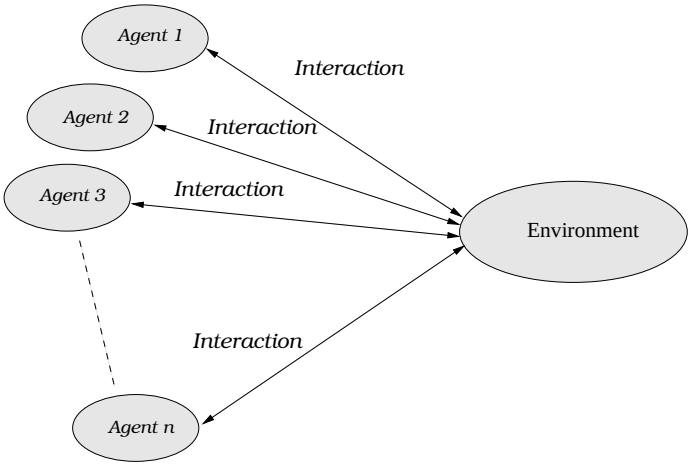
An environment is the world where an agent π or a group of agents $\{\pi_1, \pi_2, \ldots, \pi_n\}$ can interact using a set of observations, actions and rewards [15]. The environment generates observations from the set of observations $\mathcal{O}$, and rewards from $\mathcal{R} \subseteq \mathbb{Q}$, and sends them to the agent. The agent performs actions from a limited set of actions $\mathcal{A}$ in response. An iteration or step i stands for one sequence of observation-action-reward. An observation at iteration i is denoted by o_i, while the

corresponding action and reward for the same iteration are denoted by a_i and r_i respectively. The string $o_1a_1r_1o_2a_2r_2$ is an example of a sequence of interactions over 2 iterations between one agent and its environment.

We define the multiagent-environment framework (Fig. 1b) as an extension of the above such that, $o_{i,j}, a_{i,j}$ and $r_{i,j}$ are respectively the observation, action and reward for agent π_j at iteration i. The order of interactions is as follows: first, the environment sends observations to all the agents at the same time, the agents communicate and then perform actions according to these observations and communications, and finally the environment provides them back with rewards. For instance, the first interaction of agents π_1, π_2 in the multiagent-environment setting denoted by $o_1a_1r_1$ is equivalent to $o_{1,1}o_{1,2}a_{1,1}a_{1,2}r_{1,1}r_{1,2}$. To achieve the aims outlined in this paper we have to assess



(a) Agent-environment framework.



(b) Multiagent-environment framework.

Fig. 1: The agent-environment framework in individual-agent and multiagent settings.

the agents' performances in isolation and collectively, hence the need for an environment setting in which we can run formal intelligence tests of measurable complexities on artificial agents using the recently described framework. For our purpose, we have chosen the anytime universal intelligence test [14]. The 3 main reasons justifying our choice are that firstly, this will be a new way of looking at collective intelligence in multiagent systems through adopting an information-theoretical approach

for evaluating intelligence. Secondly, this test is derived from formal and mathematical backgrounds that have been practically used to evaluate diverse kinds (including machines) of entities. Finally, and more importantly, the test embraces all of the concerns we raised in the introduction. For instance the test is "universal" in the sens that it was designed to evaluate any kind of system (of any level of intelligence) over unbiased settings. The test can be stopped at "anytime" and can also adapts to the time-scale of the evaluated entity.

3.2 Evaluating the Intelligence of Artificial Agents

We introduce hereafter a simple spacial environment class, the Λ environment class [14, Sec. 6] that focuses on a restricted - but important - set of tasks in AI. This environment class implements the theory behind the Anytime Universal Intelligence test [14]. The general idea is to evaluate an agent that can perform a limited set of actions, by placing it in a grid of cells with two special objects, *Good* ($\oplus$) and *Evil* ($\ominus$) traveling in the space using movement patterns of measurable complexity. The rewards are defined as a function of the position of the evaluated agent with respect to the positions of $\oplus$ and $\ominus$. We present hereafter our own implementation Λ^*, consisting of an extension and modification on the original environment class.

Structure of the Test We generate an environment space as an m-by-n grid-world populated with objects from $\Omega = \{\pi_1, \pi_2, \ldots, \pi_x, \oplus, \ominus\}$, the finite set of objects. The set of evaluated agents $\Pi \subseteq \Omega$ is $\{\pi_1, \pi_2, \ldots, \pi_x\}$. Each element in Ω can have actions from a finite set of actions $\mathcal{A}$ ={*left, right, up, down, up-left, up-right, down-left, down-right, stay*}. All objects can share the same cell at the same time except for $\oplus$ and $\ominus$ where in this case, one of them is randomly chosen to move to the intended cell while the other one keeps its old position. Following the multiagent-environment architecture described in subsection 3.1, a test episode (over the Λ^* environment) consisting of a series of ϑ iterations $o_i a_i r_i$ such that $1 \leq i \leq \vartheta$, is modeled as follows:

1. the environment space is first initialized to an m-by-n toroidal grid-world where $\mathrm{m}, \mathrm{n} > 2$
2. it is then populated with a subset of evaluated agents from $\Pi \subseteq \Omega$ and the two special objects $\oplus$ and $\ominus$
3. the environment sends to each agent a description of its range of 1 *Moore neighbor* cells and their contents (the rewards in these cells) as an observation[3]
4. the agents (communicate and) respond to the observations by performing an action in $\mathcal{A}$, and the special objects perform the next action in their (recurrent) movement pattern.
5. the environment then returns a reward to each evaluated agent based on its position with respect to the locations of the special objects
6. this process is repeated again from point (3) until a test episode is finished.

We are using a toroidal grid space (periodic boundaries) in the sense that moving off one border makes you appear on the facing one. Consequently, the distance $d_{a,b}$ between two objects "a" and "b" is calculated using the surpassing rule (toroidal distance) such that, in a 5-by-5 grid space for example, the distance between cell (1, 3) and (5, 3) is equal to 1.

[3] For the rest of this paper, we will use the term neighborhood, or neighbor cells, to indicate the range of 1 Moore neighborhood.

Rewarding function The environment sends a reward to each evaluated agent from the set of rewards $\mathcal{R} \subseteq \mathbb{Q}$ where $-1 \leq \mathcal{R} \leq 1$. For instance, given an agent π_j, its reward $r_j \in \mathcal{R}$ is calculated as follows:

$$r_j \leftarrow \begin{cases} +1, & \text{if } d_{\pi_j,\oplus} = 0 \quad (1a) \\ -1, & \text{if } d_{\pi_j,\ominus} = 0 \quad (1b) \\ +0.5, & \text{if } d_{\pi_j,\oplus} = 1 \quad (1c) \\ -0.5, & \text{if } d_{\pi_j,\ominus} = 1 \quad (1d) \\ 0, & \text{otherwise.} \quad (1e) \end{cases}$$

Note that when an agent is neighbor to $\oplus$ and $\ominus$ simultaneously, its reward is calculated as the sum of the positive and negative rewards returned from Equations (1a), (1b), (1c) or (1d), as appropriate.

Agents do not have a full representation of the space as the environment only sends them observations of their neighborhood. An agent π_j is called an *informed agent* when the (absolute) value of its reward $|r_j| > 0$. The environment is dynamic since the agents and the special objects might change their states or positions at each iteration.

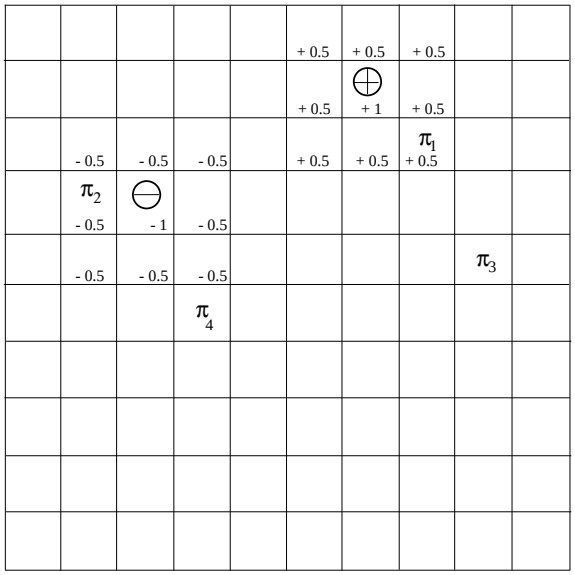


Fig. 2: A representation of the Λ^* environment class showing the Good ($\oplus$) and Evil ($\ominus$) special objects, as well as four agents in the space. The numerical values in the cells adjacent to the special objects correspond to the rewards given by the environment in the rang of 1 Moore neighbor cells of $\oplus$ and $\ominus$.

Task complexity We regard the Kolmogorov complexity [17] (randomness) of the movement patterns of the special objects as a measure of the algorithmic complexity $K(\mu)$ of the environment

μ in which they operate. We measured the Lempel-Ziv complexity [16] of the movement patterns as an approximation to $K(\mu)$, as suggested in [8,16]. Note that $\oplus$ and $\ominus$ have equally complex, but (possibly) different movement patterns. The recurrent segment of the movement pattern is at least of length one and at most $\lfloor \vartheta/2 \rfloor$, cyclically repeated until the final iteration of the test.

Space complexity We measure the search space complexity $\mathcal{H}(\mu)$ as the amount of *uncertainty* in μ, expressed by Shannon's entropy [23]. Let N be the set of all possible states of an environment μ, such that a state s_μ is the set holding the current positions of the special objects $\oplus$ and $\ominus$ in the m-by-n space. Thus the number of states $|N|$ is equal to the permutation ${}^{m\times n}P_2 = \frac{(m\times n)!}{(m\times n-2)!}$.

The entropy is maximal at the beginning of the test as there is complete uncertainty about the current state of μ. Therefore $p(s_\mu)$ follows a uniform distribution and is equal to $1/|N|$. Using $\log_2$ as a base for our calculations, this results with: $\mathcal{H}(\mu) = -\sum_{s_\mu \in N} p(s_\mu) \log_2 p(s_\mu) = \log_2 |N|$ bits.

In spit of its simplicity, the Λ^* environment class provides us with a "universal" test that is "anytime", and where the complexity of the tasks is quantified using a (rough) approximation of their Kolmogorov complexity, and the complexity of the search space can be measured as the entropy, better understood as the uncertainty in the space. Moreover, the Λ^* class can possibly be mapped or extended to model a range of games/problems in the literature as it is an abstraction of searching for a moving target (while avoiding injury), nest selection when there is one and only one best nest, as well as pattern recognition problems.

Intelligence We extend the metric of intelligence of individual agents as defined in [14, Definition 10] into an intelligence metric (Definition 2) returning a reward accumulation per-agent measure of success for a group of (isolated or cooperative) agents Π (Definition 1) over a selection of Λ^* environments L.

Definition 1. *Given a Λ^* environment μ of task and space complexities $K(\mu)$ and $\mathcal{H}(\mu)$ respectively, and a set of agents $\Pi = \{\pi_1, \ldots, \pi_n\}$, the average score $\tilde{r}_{\Pi,\mu,\vartheta}$ of Π over a test episode of ϑ-iterations is:*

$$\tilde{r}_{\Pi,\mu,\vartheta} = \frac{\sum_{j=1}^{n} \sum_{i=1}^{\vartheta} r_j^i}{n \times \vartheta}$$

Definition 2. *The intelligence $\Upsilon(\Pi)$ (taken from a finite set of reward-sensitive environments of equal uncertainty, and finite number of interactions) of a set of (isolated or cooperative) agents Π is:*

$$\Upsilon(\Pi) = \frac{1}{\omega} \sum_{\mu \in L} \tilde{r}_{\Pi,\mu,\vartheta}$$

where L is a finite subset of ω environments of task complexities $K(\mu)$ (extracted from the set of all environment complexities with a uniform probability), being ϑ-actions reward sensitive.

4 Agent Types and Behaviors

We implemented five agent behaviors to be tested in isolation and later in groups over the Λ^* environment class.

4.1 Isolated Agent Behaviors

A description of the isolated agents' behaviors is given below.

Local Search Agents Given an agent π_j, we denote by c_j^i and $r(c_j^i)$ the cell where π_j is located at iteration i, and the reward in this cell respectively. Let N_j^i and $R(N_j^i)$ denote respectively the set of neighbor cells of agent π_j (including c_j^i) at iteration i, and the reward values in these cells. $\mathtt{R}(c_j^i, a)$ is a function that returns the reward agent π_j gets after performing action $a \in \mathcal{A}$ when it is in cell c_j^i. We define the behavior of local search agents in (2):

$$a_j^i \leftarrow \begin{cases} \arg\max\limits_{a \in \mathcal{A}} \mathtt{R}(c_j^i, a) & \textit{if } \pi_j \textit{ informed} \\ \textit{any } a \in \mathcal{A} & \textit{otherwise.} \end{cases} \tag{2}$$

Reinforcement Learning Agents We have evaluated the reinforcement learning agents consisting of the following two behaviors:

- *Q-learning* behavior where agents learn using an action-quality function in order to find the best action-selection policy for a given Markov Decision Process.

- *Sarsa* behavior where agents learn a Markov Decision Process policy using an on-policy temporal-difference learning technique.

We take a q-matrix state to be the unique combination of a given (cell) position c at a given iteration i of test, leading to a total number of states of $(\mathrm{m} \times \mathrm{n})^{\vartheta}$[4]. Before evaluating Q-learning and Sarsa agents, we trained them previous to each episode using a discount factor of 0.9 and a learning rate of 0.5.

Oracle Agent The *oracle* agent knows/predicts the future movements of the special agent $\oplus$. At each step i of an episode, this agent approaches the subsequent $i+1$ cell destination of $\oplus$ seeking maximum payoff. However, if $\oplus$ has a constant/unique movement pattern (e.g. moves constantly to the right) pushing it away from the oracle, then the oracle will move in the opposite direction in order to intercept $\oplus$ in the upcoming test steps. Once it intercepts $\oplus$, it then continues operating using its normal behavior.

Random Agent a random agent randomly choses an action from the finite set of actions $\mathcal{A}$ at each iteration until the end of an episode.

The scores of the random and oracle agents will be used as a baseline for the experiments where a random agent is used as lower bound, and an oracle as an upper bound.

4.2 Agent Collectives

The abovementioned agent types were also evaluated collectively in groups. A description of these collectives is given below:

[4] Recall that m and n represent the grid space row and column dimensions respectively, and ϑ is the number of iteration in a test episode.

Local search collective using stigmergy (indirect Communication) We propose a simple algorithm for enabling communication between local search agents using stigmergy (indirect communication). For instance, we let the informed agents induce fake rewards in the environment, thus indirectly informing neighbor agents about the proximity of the special objects. Note that fake rewards will not affect the score (real reward payoff) of the agents. Let $\hat{r}(c^i)$ denote the fake reward value in cell c at iteration i and $\hat{\mathtt{R}}(c^i_j, a)$ is a function returning the fake reward agent π_j gets after performing action $a \in \mathcal{A}$ when it is in cell c^i_j. Fake rewards are induced by each agent in the environment according to Algorithm 1 over three steps as follows:

1. find $r^{max}$and r^{min}, the highest positive and lowest negative reward values respectively in neighbor cells N^i_j
2. set the fake reward value $\hat{r}$ to: $\gamma(r^{max} + r^{min})$, where γ is a discounting factor
3. for each cell $c^i \in N^i_j$, set the fake reward $\hat{r}(c^i)$ in this cell to be equal to: $r(c^i) + \hat{r}$

Algorithm 1 Fake reward generation over one iteration i

Input: Ω (set of objects), Π (set of evaluated agents), $0 < \gamma < 1$ (fake reward discounting factor).
Begin
  for j $\leftarrow$ 1 **to** $|\Pi|$ **do** ▷ loop over agents
    if $\exists c^i \in N^i_j | r(c^i) \neq 0$ **then** ▷ if agent informed
      $r^{max} \leftarrow \max R(N^i_j)$
      $r^{min} \leftarrow \min R(N^i_j)$
      $\hat{r} \leftarrow \gamma(r^{max} + r^{min})$
      for all $c^i \in N^i_j$ **do**
        $\hat{r}(c^i) \leftarrow \hat{r} + r(c^i)$
      end for
    end if
  end for
End

The agents then choose an action from the set of actions $\mathcal{A}$ using a local search behavior by relying on fake rewards this time instead of the real rewards, as follows:

$$a^i_j \leftarrow \begin{cases} \arg\max\limits_{a \in \mathcal{A}} \hat{\mathtt{R}}(c^i_j, a), & if\ \pi_j\ informed \\ any\ a \in \mathcal{A}, & otherwise. \end{cases}$$

Local Search Collectives Using Auctions (direct Communication) In this behavior, the evaluated agents go into a *single dimensional English auction* [20] at each iteration i, and they bid on the right to lead the agents in their neighborhood. At each iteration, each agent generates a value of how much reward exists in its neighborhood, which is then used as its bidding money for the auction. The richest agent wins the auction visibly to all the other agents, more formally: $winner^i \leftarrow \arg\max\limits_{\pi_j \in \Pi} \max R(N^i_j)$. The winner then selects the *target* cell to be approached by all the other agents in the collective as follows:

$$target \leftarrow \arg \max_{c \in N^i_{winner}} r(c).$$

Local Search Collectives Using Imitation A group of isolated local search agents is put in the same space with one (unevaluated) oracle agent. Local search agents imitate the oracle by following it into the same cell only when it is in their visibility range (neighborhood) otherwise, they operate using their normal behavior.

Reinforcement Learning Collectives To enable collaboration between reinforcement learning agents we let them share and update a common q-matrix, thus making them all learn and coordinate simultaneously.

We evaluated both Q-learning and Sarsa collectives independently.

5 Experimentation

We have evaluated the isolated agents and agent collectives previously described over 10000 episodes of $\vartheta = 50$ iterations each. In each episode we test over a different environment μ of a fixed (size) uncertainty $\mathcal{H}(\mu)$, but using different movement patterns for the special objects such that the task complexity $K(\mu) \in [2, 20]$ (recall Subsection 3.2 and [8]). Each environment is initialized with different initial cell positions for all objects in Ω. We repeated the experiment over different (environment sizes) uncertainties $\mathcal{H}(\mu)$, and the intelligence of the evaluated agents was recorded across the experiments.

5.1 Results and Discussion

Figure 3 shows the intelligence measures $\Upsilon(\Pi)$ (from Definition 2) of the different groups of agents Π, taken from the experiments described above. Note that the coefficient of variation $\hat{\sigma}/\Upsilon(\Pi)$, is less than 6% across our experiments. We discuss below some of the factors influencing the effectiveness/intelligence of agent collectives:

Collective decision-making: we observe that the same agent selection using different collective decision-making techniques achieves different results. For instance, Fig. 3 shows that, adopting auctions (direct communication) to achieve consensus between local search agents can be more effective than using stigmergy or indirect communication. The results also illustrate the power of imitation. Introducing heterogeneity in the group (by including a smart agent) allowed local search agents to form smart coalitions and overstep their performance threshold in the homogeneous setting.

Uncertainty: we observe that increasing the uncertainty $\mathcal{H}(\mu)$ of the environments negatively affects the performance of the evaluated agents. For instance, the higher the uncertainty in the environment, the larger the gap is between the scores of isolated local search agents and agent collectives on one side, and between the scores of local search collectives on the other side.

Learning, complexity and intelligence: we also evaluated the agents over recurrent patterns with different $K(\mu)$ complexities (randomness)[5]. To ensure that the scores underline a measure of the learning task (rather than search), we initialize the environment by allocating the evaluated agents to adjacent (neighbor) grid cells to $\oplus$. Results illustrated in Fig. 4 below, show that the performance of reinforcement learning collectives gradually decreases when evaluated (for a fixed number

[5] Again using the same number of time-steps ($\vartheta = 50$) iterations

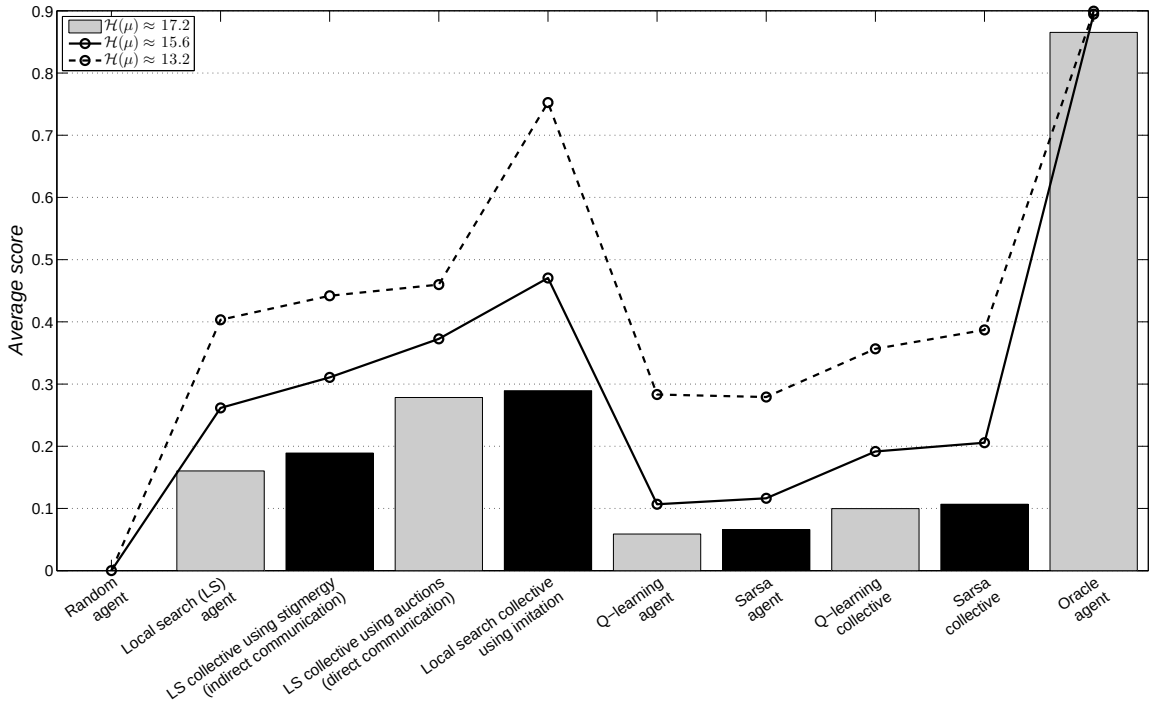
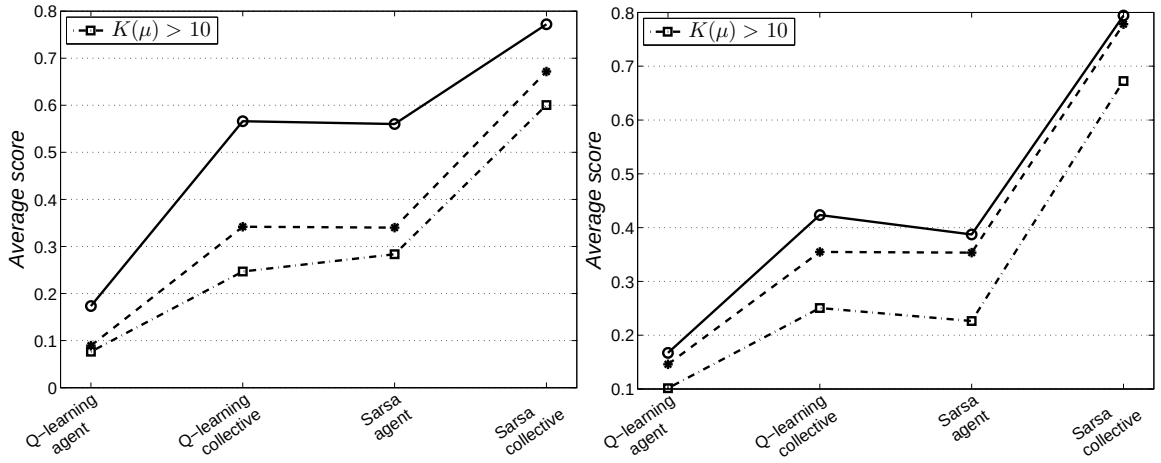


Fig. 3: Intelligence measures $\Upsilon(\Pi)$ of the different groups of agents Π with $|\Pi| = 5$ agents over different uncertainties $\mathcal{H}(\mu)$.

of time-steps) over patterns with increasing algorithmic complexities $K(\mu)$. For instance, learning (a MDP behind) random patterns is difficult as opposed to learning systematic ones. Moreover, the gap between the scores of reinforcement learning agents seems to decline with the uncertainty $\mathcal{H}(\mu)$ of the environments. Yet, as presumed, the performance of local search agents (not showing in Fig. 4) is not affected by the pattern complexities when evaluated in this setting. For instance, a local search agent locally explores candidate solutions with no attempt to learn a policy or a value function that maximizes its payoff - which also validates the results from this experiment. Accordingly, using local search to tackle tasks/problems of high $K(\mu)$ complexities (given a limited amount of time-steps) might be more advantageous than applying reinforcement learning.

Simulation: a simulation visualizing the behavior of the previously described agents/collective types over the Λ^* environment was designed. A short video (demo and brief discussion) from a sample experiment, showing the collective behavior of local search agents using stigmergy in comparison to isolated local search agents, is available in [1]. The visualization shows the formation of non-strategic coalitions among neighbor local search agents while they sought positive rewards. After a few iterations of the test, many cooperative agents pile up in the same grid-cells and seem to work as one group following an identical target.



(a) Average intelligence scores in $\mathcal{H}(\mu) \approx 13.2$ bits environments.

(b) Average intelligence scores in $\mathcal{H}(\mu) \approx 15.6$ bits environments.

Fig. 4: Average scores for local search and reinforcement learning agents/collectives over increasing intervals of $K(\mu)$ complexities (pattern randomness), with a $|\Pi| = 5$ agents.

6 Conclusion and future work

This paper touches upon several sub-communities in multiagent systems, including simulation, learning and decision-making. It also introduces a novel information-theoretical branch of work that quantifies and analyzes the capacity for, and spread of intelligence among agents in cooperative systems. Results show that cooperative groups of algorithms can be more effective/intelligent per agent than others and they are influenced by many factors, some of which are the uncertainty in the search space and the algorithmic complexity of the addressed canonical task. Our future work consists of extending our metric of intelligence by measuring coordination in multiagent systems, and introducing noisy information in our tests for their high relevance to multiagent problems.

References

1. Simulation on "the collective behavior of local search agents". [Online] `https://www.youtube.com/watch?v=ynr-PvVFicc&feature=youtu.be` (November 2014)
2. Amato, C., Oliehoek, F.A.: Scalable planning and learning for multiagent pomdps (2015), `http://people.csail.mit.edu/camato/publications/AmatoOliehoek_AAAI15.pdf`
3. Bien, Z., Bang, W.C., Kim, D.Y., Han, J.S.: Machine intelligence quotient: its measurements and applications. Fuzzy Sets and Systems 127(1), 3 – 16 (2002), `http://www.sciencedirect.com/science/article/pii/S016501140100149X`
4. Brooks, R.A.: Intelligence without reason. In: Proceedings of the 1991 International Joint Conference on Artificial Intelligence. pp. 569–595 (1991)
5. Chaitin, G.J.: Godel's theorem and information. International Journal of Theoretical Physics 21(12), 941–954 (1982)
6. Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. In: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence. pp. 746–752. AAAI

'98, American Association for Artificial Intelligence, Menlo Park, CA, USA (1998), http://dl.acm.org/citation.cfm?id=295240.295800
7. Dowe, D.L., Hernández-Orallo, J., Das, P.K.: Compression and intelligence: Social environments and communication. In: Proceedings of the 4th International Conference on Artificial General Intelligence. pp. 204–211. AGI'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2032873.2032895
8. Evans, S., Hershey, J., Saulnier, G.: Kolmogorov complexity estimation and analysis. In: Sixth World Conference on Systemics, Cybernetics and Informatics (2002)
9. Farinelli, A., Nardi, D., Pigliacampo, R., Rossi, M., Settembre, G.P.: Cooperative situation assessment in a maritime scenario. International Journal of Intelligent Systems 27(5), 477–501 (2012), http://dx.doi.org/10.1002/int.21532
10. Halmes, M.: Measurements of collective machine intelligence. Computing Research Repository (CoRR) abs/1306.6649 (2013), http://arxiv.org/abs/1306.6649
11. Hazon, N., Lin, R., Kraus, S.: How to change a group's collective decision? In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. pp. 198–205. IJCAI '13, AAAI Press (2013), http://dl.acm.org/citation.cfm?id=2540128.2540159
12. Helsinger, A., Lazarus, R., Wright, W., Zinky, J.: Tools and techniques for performance measurement of large distributed multiagent systems. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 843–850. AAMAS'03, ACM (2003), http://doi.acm.org/10.1145/860575.860711
13. Hernández-Orallo, J.: Beyond the turing test. J. of Logic, Lang. and Inf. 9(4), 447–466 (Oct 2000)
14. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: Towards an anytime intelligence test. Artificial Intelligence 174(18), 1508–1539 (Dec 2010), http://dx.doi.org/10.1016/j.artint.2010.09.006
15. Legg, S., Hutter, M.: Universal intelligence: A definition of machine intelligence. Minds and Machines 17(4), 391–444 (2007)
16. Lempel, A., Ziv, J.: On the Complexity of Finite Sequences. Information Theory, IEEE Transactions on 22(1), 75–81 (Jan 1976), http://dx.doi.org/10.1109/tit.1976.1055501
17. Li, M., Vitányi, P.: An introduction to Kolmogorov complexity and its applications (3rd ed.). Springer-Verlag New York, Inc. (2008)
18. Moriyama, K., Kurihara, S., Numao, M.: Cooperation-eliciting prisoner's dilemma payoffs for reinforcement learning agents. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems. pp. 1619–1620. AAMAS'14, Richland, SC (2014), http://dl.acm.org/citation.cfm?id=2615731.2616091
19. Oppy, G., Dowe, D.L.: The Turing Test. In: Zalta, E.N. (ed.) Stanford Encyclopedia of Philosophy. Stanford University (2011), http://plato.stanford.edu/entries/turing-test/
20. Parsons, S., Rodriguez-Aguilar, J.A., Klein, M.: Auctions and bidding: A guide for computer scientists. ACM Comput. Surv. 43(2), 10:1–10:59 (Feb 2011), http://doi.acm.org/10.1145/1883612.1883617
21. Scerri, P.: Modulating communication to improve multi-agent learning convergence. In: Sorokin, A., Pardalos, P.M. (eds.) Dynamics of Information Systems: Algorithmic Approaches, Springer Proceedings in Mathematics & Statistics, vol. 51, pp. 231–250. Springer New York (2013), http://dx.doi.org/10.1007/978-1-4614-7582-8_7
22. Settembre, G.P., Scerri, P., Farinelli, A., Sycara, K.P., Nardi, D.: A decentralized approach to cooperative situation assessment in multi-robot systems. In: AAMAS. pp. 31–38 (2008)
23. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal, The 27(3), 379–423 (July 1948)
24. Valentini, G., Hamann, H., Dorigo, M.: Self-organized collective decision making: The weighted voter model. In: Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems. pp. 45–52. AAMAS'14 (2014), http://dl.acm.org/citation.cfm?id=2615731.2615742
25. Wolpert, D.H., Tumer, K.: An introduction to collective intelligence. Tech. rep., Handbook of Agent technology. AAAI (1999), http://arxiv.org/abs/cs.LG/9908014

26. Zick, Y., Markakis, E., Elkind, E.: Arbitration and stability in cooperative games with overlapping coalitions. J. Artif. Intell. Res. (JAIR) 50, 847–884 (2014), http://dx.doi.org/10.1613/jair.4237