The Effects of Bounding Rationality on the Performance and Learning of CHREST Agents in Tileworld

Martyn Lloyd-Kelly, Peter C.R. Lane and Fernand Gobet

Abstract Learning in complex and complicated domains is fundamental to performing suitable and timely actions within them. The ability of chess masters to learn and recall huge numbers of board configurations to produce near-optimal actions provides evidence that *chunking* mechanisms are likely to underpin human learning. Cognitive theories based on chunking argue in favour for the notion of bounded rationality since relatively small chunks of information are learnt in comparison to the total information present in the environment. CHREST, a computational architecture that implements chunking theory, has previously been used to investigate learning in deterministic environments such as chess, where future states are solely dependent upon the actions of agents. In this paper, the CHREST architecture is implemented in agents situated in "Tileworld", a stochastic environment whose future state depends on both the actions of agents and factors intrinsic to the environment which agents have no control over. The effects of bounding agents' visual input on learning and performance in various scenarios where the complexity of Tileworld is altered is analysed using computer simulations. Our results show that interactions between independent variables are complex and have important implications for agents situated in stochastic environments where a balance must be struck between learning and performance.

Martyn Lloyd-Kelly
Department of Psychological Sciences, University of Liverpool, Liverpool, L69 3BX, UK e-mail: M.Lloyd-Kelly@liverpool.ac.uk

Peter C.R. Lane
School of Computer Science, University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK e-mail: peter.lane@bcs.org.uk

Fernand Gobet
Department of Psychological Sciences, University of Liverpool, Liverpool, L69 3BX, UK e-mail: fernand.gobet@liverpool.ac.uk

1 Introduction

Improving performance in a particular domain is potentially a problem of leviathan proportions depending upon the complexity of the domain in question and the cognitive limitations of an agent. Expert behaviour in complex domains such as chess has been studied extensively and of particular relevance to this discussion is that the difference in performance between chess masters and amateurs does not hinge upon mental ability (the depth of search space when considering an action does not differ significantly between masters and amateurs, for example) but rather on the breadth and quality of knowledge learnt and possessed by masters [5].

Chess' complex and complicated nature[1] results in an enormous state space. Jongman calculates that there are potentially 143.09 bits of information if all possible positions are considered [11], which buttresses Shannon's earlier calculations of a potential space of 10^{43} positions ($2^{143.09}$) [15]. However, Shannon's and Jongman's space includes redundant and implausible positions; de Groot and Gobet attempt to rectify this and calculate that the total space of possible chess positions contains 50 bits of information, giving 10^{15} positions [5]. Given known limitations on human cognition [7], the pertinent question is then: how do chess masters learn and retain such large[2] databases of chess positions?

One solution is *chunking* [3], whereby an agent aggregates many pieces of information from the environment into small units, and uses these chunks as input to its learning or production-rule system. This theory explains how substantial knowledge can lead to an improved ability to extract information from the environment, despite the cognitive limitations mentioned earlier. With respect to chess, Simon and Gilmartin propose that chess masters learn and retain between 10,000 to 100,000 chunks in memory [19], whilst Gobet and Simon propose a figure of 300,000 chunks [9].

The CHREST architecture [8] implements chunking computationally and has provided strong evidence, via simulation, that chunking mechanisms underpin human cognition in domains including board games [2, 5], implicit learning [13], and language acquisition [6, 10]. CHREST itself is a symbolic cognitive architecture, like ACT-R [1] and Soar [12]. However, unlike these other architectures, CHREST must acquire its internal knowledge from experience using *perceptual* chunking. The domains in which CHREST has been applied in the past are notably *deterministic*, i.e. future states and environmental complexity are determined by the actions of the agents situated within it; intrinsic environmental variables, relating to the environment (time allowed to make a decision, the conditions for removal of pieces in a game etc), contribute nothing.

Simon's proposal of *bounded rationality* [17] complements chunking theory since its assumptions regarding limitations of human cognition can serve to reduce the complexity of an environment for an agent. Note that an agent's rationality may

---

[1] Complex in that there are many pieces capable of being moved, complicated in that there may be many possible solutions to a given position.

[2] With respect to both the number of positions and the amount of information in each position.

be *voluntary* rather than *obligatory*; de Groot and Gobet show that a chess master's perceptual coverage of a chess board is larger than that of non-masters [5]. Non-masters voluntarily bind their rationality by decreasing their visual-field since this serves to reduce the total complexity of the chess board, which in turn facilitates learning since there is less information to process. However, this has a negative impact on performance, since an optimal solution in a smaller area of the board may no longer be optimal when a larger portion of the board is considered.

In this paper, we implement a set of multi-agent simulations to examine how the performance and learning of agents based on the CHREST architecture and situated in a stochastic environment of varying levels of complexity, are affected when voluntary rationality is bound to different degrees. The stochastic environment in which these agents are situated is the *Tileworld* environment [14], where agents must fill holes with tiles in a two-dimensional world. To enable such behaviour, visual information is used as input to a domain-specific production-rule system to generate actions; visual information and actions are also encoded as "patterns" and are used as input to each agent's CHREST architecture to enable learning during run-time. We place bounds upon voluntary rationality by manipulating an agent's "sight-radius" parameter; this dictates the size of an agent's visual input and thus the amount of information that can be passed as input to the agent's CHREST architecture and production-rule system.

To the best of our knowledge, a pattern-orientated model of learning such as CHREST has never been implemented in Tileworld before. Furthermore, we are not aware of any research that analyses the effects of bounding voluntary rationality to different degrees on an agent's learning and performance in the context of a stochastic environment whose intrinsic and extrinsic complexity can be precisely manipulated.

The paper is structured as follows: section 2 provides additional background to Tileworld and justifies its use in this research; section 3 discusses the CHREST architecture; section 4 describes how CHREST is incorporated into the agent architecture used; section 5 gives an overview of the simulations conducted; section 6 presents and discusses the results from these simulations; section 7 concludes the paper by summarising key points.

2 Tileworld

The Tileworld testbed is intended to provide a multi-agent simulation environment where the meta-level reasoning of agents can be analysed [14]. As mentioned in section 1, the Tileworld environment is stochastic; it is highly parameterized and a substantial degree of control can be exerted over intrinsic environment parameters that affect the environment's complexity. These properties allow experimenters to be exceptionally specific with regard to what factors affect the environment's future state and the actions of agents within the environment itself. Since our investigation requires a stochastic environment and a high degree of control over parameters which directly influence its stochastic nature, we assert that Tileworld satisfies such requirements and is fit for purpose.

In a typical Tileworld environment a number of tiles, holes and agents exist. Some variations of Tileworld include explicit obstacles, but these have not been included in our implementation since tiles, holes and agents naturally act as obstacles.[3] The goal of agents in Tileworld is to push tiles to fill holes; when this occurs, the "pusher" earns a point and both the tile and hole disappear. One of the crucial intrinsic factors which contributes to the stochastic nature of Tileworld is that the number of tiles and holes is not finite: new tiles and holes can be created at a defined time interval with a defined probability and can disappear after a defined period of time. A comprehensive list of all CHREST, agent and environmental parameters that can be set in our version of Tileworld, can be found in Table 1.

Table 1 Agent, CHREST and environment parameters and descriptions (see section 3 for details of CHREST parameters and section 4 for details of agent parameters).

Parameter	Type	Description
Add link time	CHREST	Time taken to create a visual-action link in LTM.
Deliberation time	CHREST	Time taken for the production-rule system to generate an action.
Discrimination time	CHREST	Time taken to discriminate a node in LTM.
Familiarisation time	CHREST	Time taken to familiarise a node in LTM.
Sight radius	Agent	Number of squares the agent can see to the north, south, east and west of itself.
Number of agents	Environment	Number of agents situated in the Tileworld.
Time limit	Environment	Length of time agents have in the Tileworld before environment is cleared.
Tile/hole birth interval	Environment	How much time must pass before a tile/hole has a chance of being "born".
Tile/hole birth prob.	Environment	Probability of a tile/hole being "born" after the value specified in the "Tile/hole birth interval" parameter has elapsed.
Tile/hole lifespan	Environment	Length of time a tile/hole is present in the Tileworld for before it is removed.

Depending upon the size of the Tileworld used, this intrinsic and extrinsic dynamism resulting from the actions of agents upon artifacts makes the environment immensely complicated and complex (assuming that the number of squares constituting the environment is sufficiently large). Simari and Parsons posit that the total number of states possible in a simplified version of Tileworld with a width of n squares, one agent and only holes (no tiles or obstacles) is $n^2 2^{n^2}$ [16].[4] They note that the limit for the tractability of direct calculations by a computer with reasonable

[3] Any Tileworld square can only be occupied by one artifact at most.

[4] The base 2 in the 2^{n^2} term of the expression is derived from the fact that Tileworld squares may be empty or may be occupied by one instance of an artifact class. In Simari and Parsons' version of Tileworld, there is only one artifact class: a hole. Therefore, in their version of Tileworld, a square will only ever be empty or occupied by a hole, which gives the base 2.

resources using a Markov Decision-Theory process to calculate optimal solutions in this simplified Tileworld is around $n = 4$, or $n = 5$. Therefore, Simon's concept of bounded rationality is practically a *necessity* in order for agents to perform any action, optimal or otherwise, when $n > 5$.

The complexity of the Tileworld environment used in these simulations is orders of magnitude greater than that used by Simari and Parsons. Our environment is a two-dimensional (35×35) grid that "wraps" (grid edges are not strict boundaries) and consists of multiple agents, tiles and holes. Following Simari and Parsons calculations, this would mean that there are 4 artifact classes that an agent may encounter (excluding itself): another agent, a tile, a hole and an empty square, resulting in $35^2 4^{35^2}$ or 4×10^{740} possible states. We assert that this degree of complexity, in addition to the ability to be able to precisely control parameters intrinsic to the environment that further alter Tileworld's complexity, provides a suitable test-bed for analysing the interplay between extrinsic and intrinsic environmental complexity on the learning rates and performance of agents situated within such an environment.

Aside from the standard Tileworld rules outlined in [14], in our version of Tileworld only one tile may be pushed at any time by an agent. For example, if an agent has two tiles to its east on consecutive squares, it is not able to push the tile closest to itself east since it is blocked by the tile two squares to the east of the agent.

3 CHREST

CHREST is an example of a symbolic cognitive architecture, with an emphasis on perception and learning. In this section, we present only those details of CHREST needed to understand the behaviour of agents in the simulations outlined; more detailed descriptions are available in [5, 8] and an implementation of CHREST is available at `http://chrest.info`.

The version of CHREST used is composed of two main components: short-term memory (STM) and long-term memory (LTM). STM and LTM hold chunks of patterns for differing modalities of which there are three: action, visual and verbal (verbal STM/LTM is not utilised in this investigation). To clarify, there is only one LTM with different modality sections whereas there are 3 independent STMs, one for each modality. The size of STM is limited to around 4 discrete chunks whereas LTM size is unlimited. Chunks are retrieved by sorting input patterns generated by the agent through a *discrimination network*. The role of the discrimination network is to sort an incoming pattern to the most relevant part of LTM. The discrimination network acts as a retrieval device and a similarity function, analogous to the hidden layers of a connectionist network, or the RETE network of Soar [12]. Unlike other cognitive architectures such as Soar and ACT-R [1], CHREST does not discriminate between types of LTM memory such as procedural, declarative or semantic.

Since Tileworld consists of squares which can contain a tile, hole, agent or nothing (empty squares are ignored), visual information perceived by an agent is encoded as an item-on-square triple by the agent's input/output component. This triple consists of: the artefact's type, its x-offset and its y-offset from the agent's current location. Thus, a visual-pattern indicating that a tile is located three squares to the east of an agent is represented as `[T 3 0]`, a hole one square to the north of the agent is represented as `[H 0 1]` and another agent one square north and one square west of the agent is represented as `[A 1 -1]`. An agent which can see one tile and one hole in its field of vision represents the visual scene as a pattern that contains a list of triples terminated by a "$" sign: `< [T 3 0] [H 0 1] $ >`. Visual patterns descend from the visual root node in LTM and are stored in visual STM.

Action-patterns are again encoded by the agent's input/output system from actions prescribed by the agent's production-rule system. Action-patterns are represented in a similar way to visual-patterns: triples consisting of an action code, direction and number of squares to move along. For example, the action "move north by one square to a tile" would be represented as a triple `[MTT 0 1]`.[5] Action patterns can be learnt as chunks in the same way as visual patterns except that these chunks descend from the action root node in LTM and are stored in the action STM.

Patterns are learnt as chunks using the processes of *familiarisation* and *discrimination*. When a pattern is input to CHREST, it is sorted through the discrimination network in LTM by checking for the presence of that pattern's information on the network's test links. If sorting retrieves a chunk from LTM, the pattern within this chunk is compared with the input pattern to determine if further learning should occur.[6] If the input pattern differs from the stored pattern (the input pattern may indicate a tile rather than a hole, for example), *discrimination* occurs and a new test link and child chunk is added to the network. If a part of the sorted pattern matches the pattern in the retrieved chunk but is more elaborate with respect to the information it contains, then *familiarisation* occurs to add this new information to the retrieved chunk. The two mechanisms of discrimination and familiarisation work together to increase the number of distinct chunks that the model can recognise and to supplement the details of already known chunks with further information.

If a visual-pattern corresponds to a completely familiarised chunk in visual LTM, and the same is true for an action-pattern in action LTM, then it is possible to associate the visual and action chunks in question together to create a visual-action link. In this investigation, these links are produced but not used in the agent's deliberation procedure since we did not want to contaminate an agent's performance with input from pattern-recognition systems. The only independent variables intrinsic to the agent that should affect its performance should be its sight radius. We mention this feature since it affects an agent's rate of learning: if an agent is creating a visual-action link, it can not discriminate or familiarise other patterns.

---

[5] MTT = "**M**ove **T**o **T**ile".

[6] Note that the chunk retrieved from LTM is also placed into STM but this version of CHREST does not make use of STM chunks.

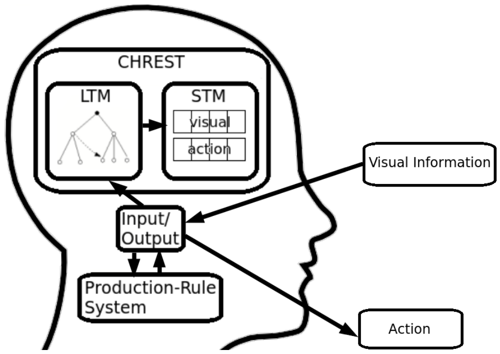

Fig. 1 Illustration of agent and CHREST architecture combination.

4 CHREST in Tileworld

As mentioned in section 1, agents in our simulations use visual-patterns as input to both their production-rule system and LTM discrimination network. These symbolic visual-patterns are created by the agent's input/output component that encodes information which agents "perceive" on squares that fall within their current sight radius (translation and structure of these patterns is discussed in section 3). The visual-patterns generated are then used as input to a goal-driven[7] production-rule system that produces actions which will be performed by the agent after being translated into action patterns and used as input to LTM (again, translation and structure of these patterns is discussed in section 3). As such, CHREST agents in Tileworld undertake an explicit deliberation and means-end reasoning procedure. The execution cycle used by CHREST agents is outlined in section 4.1 and the production-rule system is described in section 4.2. Note that learnt visual or action-patterns play no role in the deliberation procedure of agents, they are simply added to the CHREST agent's LTM discrimination network. Figure 1 illustrates the combination of the agent and CHREST architecture for clarification.

The production-rule system used by agents in the Tileworld environment implemented in this investigation can create up to 17 actions: the "move-randomly", "move-to-tile", "move-around-tile", "push-tile" actions (of which there are four variations each: north, south, east and west) and the "remain-stationary" action of which there is only one variation. Actions that are generated by agents are loaded for execution and performed after a certain amount of time has elapsed (this value is controlled by the "deliberation-time" parameter, δ, that is set by the experimenter). So, if an agent loads an action, α, for execution at simulation time, t, the agent will perform α at time $t + \delta$, simulating time passing whilst an agent decides upon what action to perform given its current visual information. Note that an agent will

[7] The agent's goals are implicit, i.e. goals are not explicitly represented in any data structure available to CHREST agents.

not generate additional visual-patterns between t and $t+\delta$ so learning is directly affected by deliberation.

As mentioned by Simari and Parsons in [16], an agent's *intention reconsideration* strategy weighs heavily upon the performance of an agent in the Tileworld environment. If the strategy implemented causes excessive intention reconsideration, then the agent will waste effort by constantly deliberating about what to do next rather than performing an action [20]. Informally, the intention reconsideration strategy implemented states: if the visual-pattern representing the current state of the agent's observable environment is equal to the visual-pattern which represented the state of the agent's observable environment used to generate the currently scheduled action, then execute the scheduled action, otherwise, generate a new action. Thus, the agent's production-rule system uses the most up-to-date visual information obtainable from the environment to inform action generation, but does not react to changes in the environment that occur between deciding upon what action to perform and performing the action. A more formal description of this intention reconsideration strategy is given in section 4.1.

4.1 CHREST Agent Execution Cycle

The agent execution cycle runs for every time increment in the Tileworld environment. Agents begin by determining if they have an action to perform.

1. An action, α, is to be performed at time, t: check to see if the current time is equal to t.
   a. The current time is equal to t: generate and attempt to learn a visual pattern, V' and compare this to the visual pattern used to generate α, V.
      i. $V = V'$: perform and attempt to learn α.
      ii. $V \neq V'$: generate a visual-pattern, V, and use this as input to the production-rule system and CHREST architecture.
   b. The current time is not equal to t: stop current execution cycle.
2. No action is to be performed: generate a visual-pattern, V, and use this as input to the production-rule system and CHREST architecture.

4.2 Production-Rule System

Given that the ultimate goal of agents in Tileworld is to push tiles into holes thus earning the agent a point, the production-rule system implemented takes a visual-pattern, V, as input and follows the procedure outlined below. Points at which actions are generated are highlighted in bold for clarity:

1. V indicates that the agent is surrounded, i.e. squares immediately north, east, south and west of the agent are occupied by non-movable tiles, holes or other agents. **Agent generates a "remain-stationary" action.**
2. V indicates that the agent is not surrounded and that tiles and holes are nearby: determine which hole is closest to the agent, H, then determine the tile that is closest to to H, T.
   a. Agent is 1 square away from T and can push it closer to H from the agent's current position: **Agent generates a "push-tile" action.**
   b. Agent is 1 square away from T but the agent's current position means that it can't push T closer to H: **Agent generates a "move-around-tile" action.**
   c. Agent is more than 1 square away from T **Agent generates a "move-to-tile" action.**
3. V indicates that the agent is not surrounded and that tiles are nearby but holes are not: determine distance of T from the agent.
   a. Agent is 1 square away from T: Agent faces T and attempts to push T along the agent's current heading (if the agent had to turn east to face T then the agent will attempt to push T east). **Agent generates a "push-tile" action.**
   b. Agent is more than 1 square away from T: **Agent generates a "move-to-tile" action.**
4. V indicates that the agent is not surrounded but can't see any tiles: **Agent generates "move-randomly" action.**

5 Simulation Overview

We study the effect of bounding voluntary rationality on learning and performance using scenarios where the intrinsic and extrinsic complexity of the environment is directly manipulated. Our experimental hypothesis states that, as environment complexity increases, bounding an agent's rationality to a greater degree will impinge on learning rates but improve performance. Two dependent variables are measured: to measure performance, we use the score of agents (how many holes the agent filled with tiles), and to measure the amount of information the agent has learned, we use the number of nodes in an agent's visual LTM. We do not consider the size of action LTM in our measure of learning since the maximum size of this LTM modality is 17 and given the length of time agents are situated in the Tileworld for (a simulated time of 4 hours), it is always the case that every action-pattern is learnt.

Since Tileworld's stochastic nature is controlled by a number of parameters (see section 2 and Table 1), some parameter values were altered and others kept constant. Table 2 delineates whether each parameter was varied or kept constant and what the parameter's value(s) was(were) set to[8].

[8] All parameters concerning time are specified in seconds.

To alter intrinsic environment complexity, the values for “tile/hole birth prob.” and “tile/hole lifespan” parameters were altered: higher tile/hole birth prob. values and lower tile/hole lifespan values equate to greater environment complexity, since more tiles/holes will appear but for shorter periods of time. One may expect the value for the “tile/hole birth interval” to also be varied in this case; however, the complexity of the environment can be significantly modified by varying the values for the “tile/hole birth prob.” and “tile/hole lifespan” parameters. To alter extrinsic environment complexity, we varied the “number of agents” parameter: the higher the number of agents, the greater the complexity of the environment, since introducing more agents should result in more interactions with artifacts in the environment, thus increasing environment dynamism.

Parameters which are varied in Table 2 have values that are grouped into three levels of complexity: below average, average and above average; for ease of reference, these complexity levels are given numerical values and used in Figures 2 and 3 in section 6: below average = 1, average = 2 and above average = 3. The values for the “number of agents” and “tile/hole lifespan” were determined by taking the average value (4 and 40, respectively) and either halving or doubling to produce the below average or above average counterpart. Values for the “sight-radius” parameter were derived by taking the minimum sight radius possible as the below average value[9] and then adding 1 for each increase in complexity. Values for the “tile/hole birth prob.” parameter were derived by simply taking the median probability, 0.5, as the average complexity value and then taking the lowest/highest values possible without guaranteeing tile/hole birth since this would significantly skew the results.

Values for parameters which are kept constant in Table 2 are justified thus: “add link time”, “discrimination time” and “familiarisation time” parameter values were taken from [18]. We chose 1 second for the “deliberation-time” parameter value since this coincides with the value chosen for the “tile/hole birth interval” parameter so that an agent may need to reconsider its planned action due to the appearance of a new tile or hole. After preliminary testing of the simulations we deemed 4 hours suitable enough for the “time-limit” parameter value since this gives agents ample time to learn and act within the environment.

In total, 27 scenarios were simulated and each repeated 10 times, permitting the collection of sufficient data so that a rigorous statistical analysis can be performed; values for dependent variables were averaged across agents for each scenario. Scenarios can be segregated into blocks of 9 where intrinsic complexity is altered: below average in scenarios 1–9; average for scenarios 10–18; above average for scenarios 19–27. Scenarios can then be segregated into blocks of 3 where extrinsic complexity is altered: above average in scenarios 1–3, 10–12 and 19–21; average in scenarios 4–6, 13–15 and 22–24; above average in scenarios 7–9, 16–18 and 25–27. Alterations to bounded rationality do not form blocks: above average in scenarios 1, 4, 7, 10, 13, 16, 19, 22 and 25; average in scenarios 2, 5, 8, 11, 14, 17, 20, 23 and 26; below average in scenarios 3, 6, 9, 12, 15, 18, 21, 24 and 27.

---

[9] Minimum “sight-radius” parameter value is 2 since agents must be able to see at least 1 square in front of a tile, so that its ability to be pushed can be determined.

Table 2 Agent, CHREST and Tileworld parameter values.

Parameter	Varied/Constant	Value(s)
Add link time	Constant	1s
Deliberation time	Constant	1s
Discrimination time	Constant	10s
Familiarisation time	Constant	2s
Sight radius	Varied	2 (below avg.), 3 (avg.), 4 (above avg.)
Number of agents	Varied	2 (below avg.), 4 (avg.), 8 (above avg.)
Time limit	Constant	14400s
Tile/hole birth interval	Constant	1s
Tile/hole birth prob.	Varied	0.1 (below avg.), 0.5 (avg.), 0.9 (above avg.)
Tile/hole lifespan	Varied	80s (below avg.), 40s (avg.), 20s (above avg.)

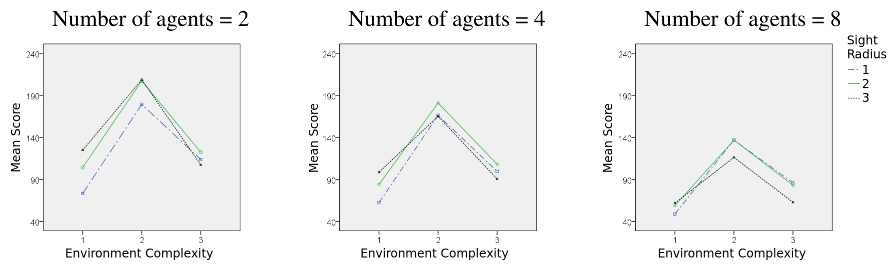

Fig. 2 Average score as a function of number of agents, complexity of the environment and sight radius.

6 Simulation Results

Figure 2 shows results pertaining to agent performance (mean agent score). To analyse the data, we carried out a $3 \times 3 \times 3$ analysis of variance (ANOVA), with environment complexity, number of agents and sight radius as between-subject variables. All one, two and three-way interactions were statistically significant ($p < 0.001$): sight radius $F(2,243) = 128.3$, number of players $F(2,243) = 1,832.0$, environment complexity $F(2,243) = 6,109.1$, sight radius $\times$ number of agents $F(4,243) = 75.5$, sight radius $\times$ environment complexity $F(4,243) = 142.5$, number of agents $\times$ environment complexity $F(4,243) = 66.2$ and sight radius $\times$ number of players $\times$ environment complexity $F(8,243) = 7.0$. The presence of multiple interactions means that the pattern of results is complex.

The main features of Figure 2 can be summarised as follows. First, complexity of the environment has a non-linear effect on mean score, with average complexity producing the highest scores. Second, as the number of players — and thus complexity — increases, mean score decreases. Third, sight radius, number of players and complexity of the environment interact in an intricate but critical way. With a below average level of environment complexity, the largest sight radius obtains the highest scores irrespective of the number of players, although minimally so with 8 players. With an above average level of environment complexity, the largest sight radius always obtains the lowest mean score. The pattern of results is less stable

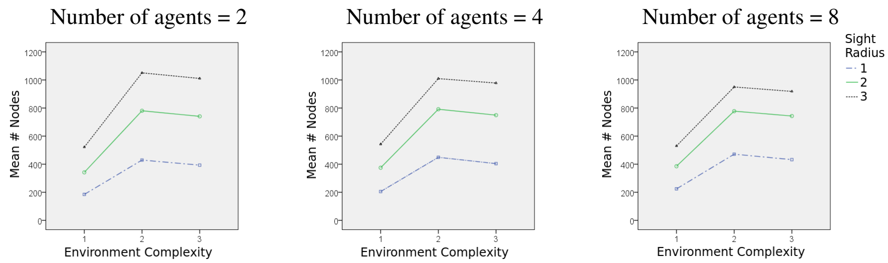

Fig. 3 Average number of visual LTM nodes as a function of number of agents, complexity of the environment and sight radius.

with an average level of complexity of the environment. As the number of players increases from 2 to 8, sight radius incurs a premium: the largest sight radius moves from obtaining the highest mean scores (although minimally so) to obtaining the lowest mean scores. Intuitively, it is as if reducing sight radius as complexity increases allows agents to filter out information and to be more selective in what they give their attention to. This is exactly what is predicted by Simon's theory of bounded rationality [17].

Figure 3 shows results pertaining to learning (mean number of nodes). As with the analysis of performance, the three main effects, the three two-way interactions and the three-way interaction were all statistically significant ($p < 0.001$): sight radius $F(2,243) = 42,187.5$, number of players $F(2,243) = 12.0$, environment complexity $F(2,243) = 31,474.6$, sight radius $\times$ number of agents $F(4,243) = 176.1$, sight radius $\times$ environment complexity $F(4,243) = 1,151.7$, number of agents $\times$ environment complexity $F(4,243) = 49.3$ and sight radius $\times$ number of players $\times$ environment complexity $F(8,243) = 19.3$.

Compared to performance, results concerning learning are simpler to interpret; two main patterns are apparent. First, the mean number of nodes learnt sharply increases from a below average level of environment complexity to an average level (an increase of about 100% in each case), and then slightly decreases from an average level of environment complexity to an above average level. Second, a large sight radius always produces a higher number of nodes, followed by a medium sight radius. Although number of agents has a statistically significant effect, this effect was very small compared to the effect of the other two independent variables and are picked up by the ANOVA due to the high statistical power of our simulations.

7 Conclusion

The relationship between complexity and bounded rationality is of considerable interest for artificial intelligence and cognitive science. One approach to this question considers how selective attention and learning, and more specifically chunking, helps humans to cope with the complexity of an environment. Having simulated key phenomena on chess expertise, the cognitive architecture CHREST provides plausible mechanisms to explain the relation between complexity and bounded rationality in a deterministic environment. In this paper, we extended CHREST's theoretical coverage to a stochastic environment, Tileworld, with the presence of multiple agents and the continuous appearance and disappearance of artifacts (tiles and holes) – Tileworld offers a challenging environment to study this question.

The simulations described in this paper looked at the effects of bounding an agent's rationality (sight radius) upon the performance of agents and their learning in context of the Tileworld environment whose complexity is altered by varying one extrinsic environment variable (number of agents situated in the environment) and two intrinsic environment variables (tile/hole birth probability and tile/hole lifespan). A systematic manipulation of these variables provided results of considerable theoretical importance.

In this paper, agents were not able to enjoy the benefits of learning with respect to improving their performance through the use of modified deliberation procedures; the complexity of the results suggests that this strategy was wise. The results not only demonstrate that too much complexity leads to weaker performance, but also that the ability to filter out complexity in taxing environments by reducing sight radius results in an improvement of performance in these environments but at the cost of learning new information. In future research, we plan to establish whether this "smaller-is-better" effect persists when agents can make use of the knowledge they have acquired to improve their deliberation procedures.

The research presented here is in the context of agent-modeling using a symbolic cognitive architecture and, as we have noted previously, there are no previous studies known to us of pattern-learning agents within the Tileworld environment. However, in the wider field of machine learning, there are some parallels to our main finding that bounding rationality can improve performance. The effect of bounding rationality in our domain is to limit the potential set of perceived perceptual patterns; the bounded rationality provides an heuristic bias towards a particular set of patterns which, the bias assumes, will be most useful for learning at that time. A related strategy is found in the area of active learning [4], in which selective sampling is used to identify the next set of unlearnt examples to include in the training pool. Future work will also look at the contrast between the theoretically-driven approaches derived from machine learning, and the heuristically-driven approaches derived from cognitive science.

References

[1] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebière, C., and Qin, Y. L. (2004). An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.

[2] Bossomaier, T., Traish, J., Gobet, F., and Lane, P. C. R. (2012). Neuro-cognitive model of move location in the game of Go. In *Proceedings of the 2012 Interna-*

tional Joint Conference on Neural Networks.

[3] Chase, W. G. and Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, 4:55–81.

[4] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15:201–221.

[5] de Groot, A. D. and Gobet, F. (1996). *Perception and Memory in Chess: Heuristics of the Professional Eye*. Van Gorcum, Assen.

[6] Freudenthal, D., Pine, J. M., and Gobet, F. (2009). Simulating the referential properties of Dutch, German and English root infinitives in MOSAIC. *Language Learning and Development*, 15:1–29.

[7] Gobet, F. and Lane, P. (2012). *Encyclopedia of the Science of Learning*, chapter Bounded Rationality and Learning. NY: Springer.

[8] Gobet, F., Lane, P. C. R., Croker, S. J., Cheng, P. C.-H., Jones, G., Oliver, I., and Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5:236–243.

[9] Gobet, F. and Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, 24:651–82.

[10] Jones, G. A., Gobet, F., and Pine, J. M. (2007). Linking working memory and long-term memory: A computational model of the learning of new words. *Developmental Science*, 10:853–873.

[11] Jongman, R. W. (1968). *Het Oog Van De Meester*. Assen: Van Gorcum.

[12] Laird, J. E. (2012). *The Soar Cognitive Architecture*. MIT Press.

[13] Lane, P. C. R. and Gobet, F. (2012). Chrest models of implicit learning and board game interpetation. In Bach, J., Goertzel, B., and Ikle, M., editors, *Proceedings of the Fifth Conference on Artificial General Intelligence*, volume LNAI 7716, pages 148–157, Berlin, Heidelberg. Springer-Verlag.

[14] Pollack, M. and Ringuette, M. (1990). Introducing the Tileworld: Experimentally evaluating agent architectures. In *Eighth National Conference on Artificial Intelligence*, pages 183–189. AAAI Press.

[15] Shannon, C. E. (1950). A chess-playing machine. *Philosophical magazine*, 182:41–51.

[16] Simari, G. I. and Parsons, S. D. (2004). On approximating the best decision for an autonomous agent. In *Sixth Workshop on Game Theoretic and Decision Theoretic Agents*, Third Conference on Autonomous Agents and Multi-agent Systems, pages 91–100.

[17] Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69:99–118.

[18] Simon, H. A. (1969). *The sciences of the artificial*. MIT Press, Cambridge, MA.

[19] Simon, H. A. and Gilmartin, K. J. (1973). A simulation of memory for chess positions. *Cognitive Psychology*, 5:29–46.

[20] Wooldridge, M. and Parsons, S. (1998). Intention reconsideration reconsidered. In *Intelligent Agents V*, pages 63–80. Springer-Verlag.