The Small Loop Problem: A Challenge for Artificial Emergent Cognition

Olivier L. GEORGEON[a,b] and James B. MARSHALL[c]
[a] *Université de Lyon, CNRS*
[b] *Université Lyon 1, LIRIS, UMR5205, F-69622, France*
[c] *Sarah Lawrence College, Bronxville, NY 10708, USA*

Abstract. We propose the Small Loop Problem as a challenge for biologically inspired cognitive architectures. This challenge consists of designing an agent that would autonomously organize its behavior through interaction with an initially unknown environment that offers basic sequential and spatial regularities. The Small Loop Problem demonstrates four principles that we consider crucial to the implementation of emergent cognition: environment-agnosticism, self-motivation, sequential regularity learning, and spatial regularity learning. While this problem is still unsolved, we report partial solutions that suggest that its resolution is realistic.

Keywords. Self-motivation, decision process, early-stage cognition.

Introduction

We introduce an idealized learning problem for an artificial agent that serves as a benchmark to demonstrate four principles of emergent cognition. We named this problem the *Small Loop Problem* (SLP). In a review of benchmarks for artificial intelligence, Rohrer [10] argued that a benchmark can constitute a formal statement of one's research goals. Accordingly, in parallel to presenting the SLP, this paper presents a statement and an argumentation in favor of four principles that we consider fundamental to emergent cognition: (a) environment-agnosticism, (b) self-motivation, (c) sequential regularity learning, and (d) spatial regularity learning.

The principle of environment-agnosticism (a) was proposed to account for the idea that the agent should not implement ontological presuppositions about the environment [8]. The SLP requires that the designer of the agent must not include predefined knowledge of the environment in the agent. Classical ways of including such knowledge would be in the form of logical rules that would exploit predefined semantics associated with sensory input, or in the form of a set of predefined states of the world that would be made desirable to the agent by the implementation of a reward function. Instead, the SLP requires environment-agnostic agents to learn the semantics of sensorimotor information and the ontological structure of their world by themselves.

The SLP approach to self-motivation (b) relates to the problem of implementing a *Discrete Time Decision Process* that learns a *policy function* P(t) to maximize a *value function* V(t) over time. Such a process implements motivation because the agent learns behaviors that fulfill an innate value system. This view assumes that such an innate value system was selected through phylogenetic evolution in the case of natural organisms to favor the survival of the organism and of its species. An agent that solves the SLP must implement a mechanism that learns such a policy function without relying on ontological presuppositions about the world. Note that traditional algorithms of reinforcement learning [12] do not fulfill this requirement because they require the designer to associate a reward value to predefined states of the world. Even Partially Observable Markov Decision Processes (POMDPs) require prior knowledge of a *state evaluation function* to assess *believed states* from observations [1].

In the SLP, the agent has a set of 6 possible actions $A=\{a_1, \dots a_6\}$ and a set of two possible observations $O=\{o_1, o_2\}$ (binary feedback). We define the set of possible interactions $I = A \times O$ as the set of the 12 tuples $i = [a_j, o_k]$ that associate a possible action with the possible observation resulting from that action. Each interaction i has a predefined numerical value v_i. The *value function* V(t) equals the value v_i of the interaction i enacted on step t. The policy function must learn to choose the action a_j at each time step t that would maximize the value function in an infinite horizon. Note the formal difference from reinforcement learning, which requires a reward function as a function of the state of the world and of the action, whereas the SLP formalism does not involve a formalization of the environment in terms of a set of states. When adapted to the SLP formalism, traditional reinforcement learning algorithms can learn short-term dependencies between observations and actions, but they fail to learn long-term temporal and spatial regularities that they need to learn to fully solve the SLP.

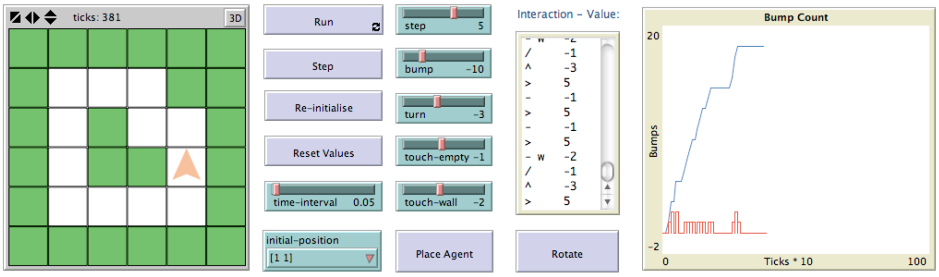
The principle of sequential regularity learning (c) follows from the fact that the agent must discover, learn, and exploit temporal regularities in its interaction with the environment to maximize the value function V(t). Doing so without prior assumptions on the environment remains an open challenge, which the SLP is designed to address.

Finally, the agent must discover, learn, and exploit spatial regularities that exist in its “body” structure and in the structure of the environment (d). Many neuroanatomists who study the evolution of animal brains argue that organization of behavior in space is a primordial purpose of cognition [e.g., 4]. Natural organisms generally have inborn brain structures that prepare them to deal with space (the tectum or superior colliculus). These observations suggest that spatial regularity learning is a key feature of emergent cognition. We designed the SLP to investigate how this feature could be integrated into a biologically inspired cognitive architecture with the other principles presented above.

Section 1 describes the SLP in detail. Section 2 reports our partial solution that implements environment-agnosticism, self-motivation, and sequence learning. Section 3 presents how we envision coupling our current solution with spatial regularity learning to move toward the full solution. The conclusion recapitulates the challenges raised by the SLP. While this problem may seem simplistic, it is still unsolved, and, we argue, it is important for the study of emergent cognition.

1. The Small Loop Problem (SLP)

The SLP consists of an artificial agent that "smartly" organize its behavior through autonomous interaction with the *Small Loop Environment*. The Small Loop Environment is the loop of white squares surrounded by green walls shown in Figure 1. Note that the SLP differs from benchmarks traditionally used in AI [e.g., 10] by the fact that the environment does not offer a final goal to reach.



step 5
bump -10
turn -3
touch-empty -1
touch-wall -2
time-interval 0.05
initial-position [1 1]

```
- w   -2
/     -1
^     -3
>     5
-     -1
>     5
-     -1
>     5
- w   -2
/     -1
^     -3
>     5
```

Figure 1. The Small Loop Platform in NetLogo.

The set of possible actions (A) contains the 6 following actions: try to move one square forward (a_1), turn 90° left (a_2), turn 90° right (a_3), touch front square (a_4), touch left square (a_5), touch right square (a_6). Each action returns a single bit observation as feedback ($O = \{o_1, o_2\}$). The 10 possible interactions are then: *step* ([a_1,o_1]), *bump* ([a_1,o_2]), *turn left* ([a_2,o_1]), *turn right* ([a_3,o_1]), *touch front/left/right empty* ([a_4,o_1]/[a_5,o_1]/[a_6,o_1]), *touch front/left/right wall* ([a_4,o_2]/[a_5,o_2]/[a_6,o_2]). Note that turn actions always return feedback o_1, which makes 10 interactions rather than 12.

The principle of environment agnosticism implies that the agent has no initial knowledge of the meaning of interactions: an interaction's label is meaningless to the agent. To demonstrate this, the SLP requires that swapping any label [a_j,o_k] with any other label [a_m, o_n] would still give rise to the same behavior when the agent is rerun.

The experimenter presets the values of interactions before running the agent (using the controls shown in Figure 1). We specify the following reference values: *step*: 5; *bump*: -10; *turn*: -3; *touch (empty or wall)*: -1. With these values, we expect the agent to learn to maximize moving forward, and avoid bumping and turning. To do so, we expect the agent to learn to use *touch* to perceive its environment and only turn in appropriate categories of situations (because *touch* has a less negative value than *bump* or *turn*). Additionally, if there is a wall ahead, the agent should touch on the side and turn towards that direction if the square is empty, so as to subsequently move forward.

Note that on each decision cycle, the best action to choose does not depend only on a single previous interaction but may depend on a sequence of several previous interactions and on the possibility of enacting several next interactions. This makes the SLP suitable to demonstrate sequential regularity learning. The highest-level most satisfying sequence consists of making a full tour of the loop, which can be repeated indefinitely. The value of this sequence is equal to 5 x 12 (move forward) -3 x 6 (turn) = 42, corresponding to 2.33 points/step.

The principles of self-motivation together with environment-agnosticism imply that the agent must adapt to any set of values. Trivial examples are those in which positive values are associated with *turn* or *bump* or *touch*: the agent would learn to keep spinning in place, or bumping, or touching indefinitely. The SLP thus consists of implementing a mechanism that tends to enact interactions with high values and to avoid interactions with negative values without any presupposition of what these interactions mean in the environment. To solve this problem, the agent must learn hierarchies of sequential regularities so it can use certain interactions to gain information to anticipate the consequences of later interactions.

Section 2 shows that a purely sequential learning mechanism can partially solve this problem. For a "smarter" organization of behavior, we, however, expect the agent to exploit spatial regularities. The agent should construct a *self model* that organizes interactions spatially. For example: touch left and turn left concern the agent's left side, touch front, move forward, and bump concern the agent's front. The agent should also categorize situations in the environment with regard to their spatial structure relative to the agent's position, for instance the categories: *left corner, right corner,* and *long edge of the loop.* This need for spatial categorization makes the Small Loop Problem suitable to demonstrate spatial regularity learning.

2. The sequential solution

We have reported an algorithm that brings a partial solution to a similar problem [7]. We now offer a NetLogo simulation online[1] to demonstrate the behavior of this algorithm on the Small Loop Environment. This demonstration shows that the agent usually learns to avoid bumping after approximately 300 steps and reaches a stable satisfying behavior that consists of circling the loop after approximately 600 steps. This demonstration also shows that the agent has difficulties in the upper right area of the loop because of the inverted corner.

Figure 2 shows the trace of an example run. This trace shows that the interactions were unorganized and poorly satisfying in the agent's terms until step 150. From step 190 on, the agent learned to touch ahead before trying to move forward, but it still got puzzled in the upper right area around step 220 and 270. In this particular instance, it learned to characterize *left corners* by the sequence that leads to them when circling the loop counterclockwise: "touch left empty, turn left, move forward, touch front wall". In this left corner context, the agent learned to chose *turn right* (steps 318, 354, 390), which allowed it to engage in full tours of the loop.

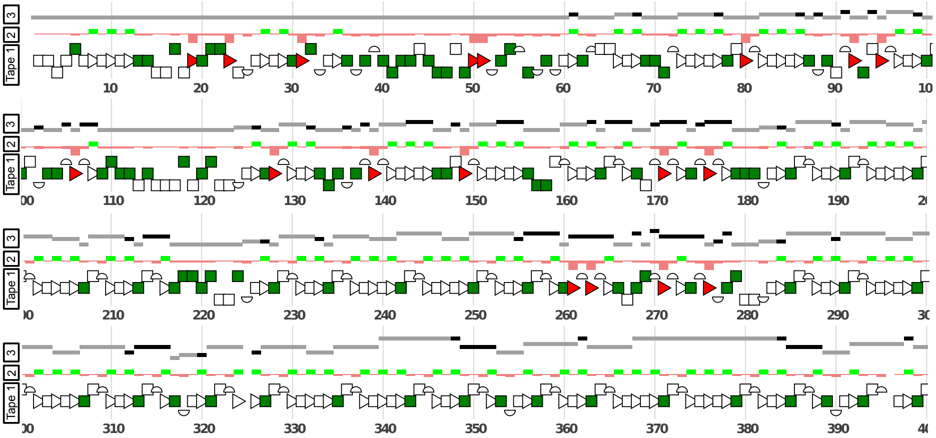


Figure 2. Example activity trace of an agent that learns sequential regularities.

In the trace in Figure 2, Tape 1 represents the interactions: touch empty (white squares), touch wall (green squares), turn (half-circles), move forward (white triangles), bump (red triangles); the upper part represents interactions *to the left*, the lower part interactions *to the right*. Tape 1 shows that the agent learned to avoid bumping after step 276 by always touching ahead before moving forward. Tape 2 represents the interactions' values as a bar-graph (green when positive, red when negative); it shows

[1] http://liris.cnrs.fr/ideal/demo/small-loop/

that the agent got more consistently positive interactions from step 290 on. Tape 3 represents the level of the enacted sequence in the hierarchy of learned sequences; it shows that the agent gradually exploited higher-level sequences. The value obtained when the behavior was stabilized was of 5x12 (move forward) – 3x6 (turn) – 1x17 (touch), corresponding to 0.71 points/step.

3. Towards the spatio-sequential solution

To give an idea of what would constitute a full solution, we started to implement a mechanism of spatial awareness. To do so, we endowed the agent with a *spatial memory* that kept track of the spatial location where interactions were enacted. The algorithm updates the spatial memory by translating its content when the agent moves forward and rotating it when the agent turns (a basic form of Simultaneous Localization And Mapping, SLAM). This solution, however, violates the principle of agnosticism because it assumes the relation between interactions and transformations in spatial memory. We also hard-wired the agent's "self model" to the spatial memory. For example, we hard coded the spatial position of the different *touch* interactions relative to the agent (left side, front, right side).

In essence, the algorithm learns *bundles of interactions* that represent observable *phenomena* in the environment. We define a bundle as the set of interactions afforded by a phenomenon [6]. The SLP provides two kinds of observable phenomena: *empty squares* and *walls*. The agent constructs bundles gradually as it explores the environment. Over time, the agent recognizes the phenomena that surround it and represents them by bundles localized in the agent's spatial memory. In turn, bundles in spatial memory generate weighted propositions (positive or negative) to enact the interactions that they contain. This mechanism increases the speed of the agent's adaptation because it helps the agent select interactions adapted to its spatial context. Figure 3 shows the effects of this mechanism in an example. A video is available online to show the entire run, the sequential trace, and the content of the spatial memory dynamically[2].

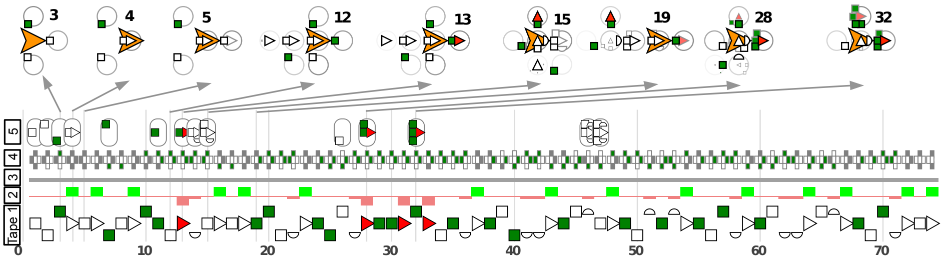


Figure 3. Example activity trace of an agent that learns spatial and sequential regularities.

In Figure 3, tapes 1 to 3 correspond to the same as in Figure 2. Tape 1 shows that the agent bumped only four times (steps 13, 28, 31, and 33). Tape 4 represents the four surrounding squares in the agent's spatial memory (squares whose content is unknown are gray). Tape 5 represents the construction of bundles over time (gray rounded rectangles that contain interactions). The upper part of Figure 3 shows snapshots of the

[2] http://e-ernest.blogspot.fr/2012/04/ernest-112.html

agent's spatial memory at different steps. Gray circles represent bundles localized in spatial memory. These circles are fading to represent decay in spatial memory.

On steps 1 to 3, the agent successively touched the three squares surrounding it. On step 4, it moved forward. Because the touching forward made on step 2 and the moving forward on step 4 concerned the same spatial location, these interactions were bundled together to represent an *empty square phenomenon*. On step 5, a new touching forward activated an *empty square bundle* in front of the agent. The interaction *move forward*, now belonging to this bundle, generated additional positive support to move forward (in the agent's decisional mechanism). In a similar way, the interaction *touching front wall* and *bumping* were bundled together on step 13. On step 19, the *touching front wall* activated the newly-created *wall bundle* in front of the agent. The bump interaction, then belonging to the wall bundle, generated negative support for trying to move forward, preventing the agent from bumping into the wall. On step 96 (not in Figure 3), the learned sequence *turn right – move forward* was added to the *empty square bundle*, which led the agent to subsequently enact this sequence when an empty square was again recognized on the right. This experiment shows that this mechanism significantly improves the agent's management of the upper right area. The agent started to circle the loop on step 70 (clockwise). The value obtained after stabilization was again of 0.71 points/step.

This example illustrates two limitations that we expect the full solution to solve. (a) The average value per step obtained after the learning phase should tend to the highest value made possible by the initial settings (2.33 with the settings proposed in Section 1). This requires implementing more elaborated learning mechanisms that make the agent appropriately renounce touching when the structure of the environment is better known. (b) The agent's "self-model" (i.e., the effects and positions of interactions in space) should not be assumed but rather learned. We, nonetheless, consider it valid to implement a spatial memory that assumes the two-dimensional grid structure of the environment. This assumption is justified by the fact that natural organisms have inborn brain structures that prepare them to deal with the spatial nature of their environment (e.g., the superior colliculus). The purpose of the SLP is, however, not to have the agent learn a full map of its environment but to adapt its behavior to local temporal and spatial context. Therefore, we posit that the area covered by the spatial memory should be smaller than the full environment space.

4. Conclusion

We propose the Small Loop Problem as a benchmark to evaluate agents that implement four principles of emergent cognition: environment agnosticism, self-motivation, sequential regularity learning, and spatial regularity learning. This benchmark contrasts with most existing benchmarks for unsupervised learning agents [e.g., 5, 11] in that it does not involve a final goal to reach. Instead, the agent's self-motivation comes from the fact that primitive interactions have different values.

We present two partial solutions. The first partial solution is based on an original sequential decision process. It demonstrates that the agent can organize its behavior by learning and exploiting sequential regularities of interactions without presuppositions on the meaning of interactions. The second partial solution illustrates an architecture that associates the sequential decision process with a spatial regularity learning mechanism. In its current version, this solution, however, conflicts with the agnosticism principle because it requires assumptions on the agent's self model. The Small Loop Challenge requires eliminating these assumptions.

Different solutions exist to learn spatial structures from uninterpreted sensors [e.g., 9], to learn self models [e.g., 3], and to categorize situations on the basis of self-motivation [e.g., 2]. Yet, the question of integrating these solutions together remains unsolved. Studies of natural organisms such as insects and archaic vertebrates suggest that these organisms manage to solve these problems through fundamental mechanisms of cognition whose replication in an artificial cognitive architecture remains a challenge. The Small Loop Problem offers a simple formalization of this challenge, which hopefully makes its resolution realistic. Solving this challenge will open the way to developing self-motivated agents capable of dealing with more complex spatiotemporal regularities in their interactions with the environment.

5. Acknowledgement

This work was supported by the French *Agence Nationale de la Recherche* (ANR) contract ANR-10-PDOC-007-01 and by a research fellowship from the *Collegium de Lyon*. We gratefully thank Frank Ritter and Christian Wolf for their comments on this report.

References

[1] Aström, K. 1965. "Optimal control of Markov processes with incomplete state information". Journal of Mathematical Analysis and Applications (10). 174-205.

[2] Blank, D.S., Kumar, D., Meeden, L. and Marshall, J. 2005. "Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture". Cybernetics and Systems, 32 (2). 125-150.

[3] Bongard, J., Zykov, V. and Lipson. 2006. "Resilient machines through continuous self-modeling". Science, 314. 1118-1121.

[4] Cotterill, R. 2001. "Cooperation of basal ganglia, cerebellum, sensory cerebrum and hippocampus: Possible implications for cognition, consciousness, intelligence and creativity". Progress in Neurobiology, 64. 1-33.

[5] Dietterich, T.G. 2000. An overview of MAXQ hierarchical reinforcement learning. In Proceedings of SARA02 4th International Symposium on Abstraction, Reformulation, and Approximation (London, UK, 2000), Springer-Verlag, 26-44.

[6] Gay, S., Georgeon, O.L. and Kim, J.W. 2012. Implementing spatial awareness in an environment-agnostic agent. In Proceedings of BRIMS2012, 21st Annual Conference on Behavior Representation in Modeling and Simulation (Amelia Island, Florida), 62-69.

[7] Georgeon, O.L. and Ritter, F.E. 2012. "An intrinsically-motivated schema mechanism to model and simulate emergent cognition". Cognitive Systems Research, 15-16. 73-92.

[8] Georgeon, O.L. and Sakellariou, I. 2012. Designing environment-agnostic agents. In Proceedings of ALA2012, Adaptive Learning Agents workshop at AAMAS2012, 11th International Conference on Autonomous Agents and Multiagent Systems (Valencia, Spain), 25-32.

[9] Pierce, D. and Kuipers, B. 1997. "Map learning with uninterpreted sensors and effectors". Artificial Intelligence, 92. 169-227.

[10] Rohrer. 2010. "Accelerating progress in Artificial General Intelligence: Choosing a benchmark for natural world interaction". Journal of Artificial General Intelligence, 2. 1-28.

[11] Sun, R. and Sessions, C. 2000. Automatic segmentation of sequences through hierarchical reinforcement learning. In Sequence Learning, Sun, R. and Giles, C.L. eds. Springer-Verlag, Berlin Heidelberg, 241–263.

[12] Sutton, R.S. and Barto, A.G. 1998. Reinforcement learning: An introduction. MIT Press, Cambridge, MA.