Learning Dynamics of Cognitive Parallel Processing Based on a Collective Evaluation

Oussama Sabri, Alexandre Muzy

Learning Dynamics of Cognitive Parallel Processing Based on a Collective Evaluation

Oussama Sabri
Côte d'Azur University
Laboratory I3S
Sophia Antipolis, FRANCE.
Email: sabri@i3s.unice.fr

Alexandre Muzy
Côte d'Azur University
CNRS, Laboratory I3S
Sophia Antipolis, FRANCE.
Email: alexandre.muzy@cnrs.fr

***Abstract*— Learning dynamics at cognitive process level is difficult to study and emulate because of the complexity of intricate psychological and neuronal mechanisms and dynamics. When considering the parallel processing of a task, the difficulty relies on the execution concurrency making the process contributions indistinguishable. We present here a metric for rewarding increasingly the right parallel cognitive processes with respect to the wrong ones through learning steps. The metric, based on the symmetric difference between task parallel processes, proves to correctly achieve collective and individual credit assignment of the processes.**

Keywords: Learning dynamics, cognitive task, parallel processing, collective credit assignment.

I. INTRODUCTION

In psychology, information process theory distinguishes between series and parallel execution of the cognitive processes of a task, from a stimulus to an action [7]. While series execution requires the ending of a process before starting a new process, parallel execution concerns thus the ability of the brain to execute several processes in parallel. Learning a task allows parallelizing the execution of processes initially in series. For example, learning to drive, a driver will learn sequentially to change gear and turn the wheel while it becomes natural for a driver to account for both actions at the same time.

Looking at the learning dynamics into more details parallelization could be viewed as trying to combine different processes until finding the right combination. However, performing different processes in parallel makes hard the individual evaluation of the processes. On the one hand, achieving at the same time a right and a wrong process can unexpectedly lead to a right behavior at action level. It is hard to distinguish then the contribution of right and wrong tasks. On the other hand, it is usually the exact conjunction of the right processes that leads to achieving correctly a cognitive task. Therefore, an efficient evaluation of parallel processing would find the minimum number of right processes to execute, not too many, not too few. Another difficulty concerns the individual evaluation of processes with respect to the learning dynamics of parallel processing. Learning different processes in parallel consists of trying many processes, right and wrong, until finding the right combination. Then, right tasks have an individual contributions that should accumulate more reward than wrong tasks through trials.

We provide here an evaluation method for combining automatically processes in parallel through different trials. At the end of the evaluation process, we prove that the right processes get more individual reward than the wrong ones through different trials, allowing to find exactly the right set of tasks, without adding wrong processes nor missing right processes, while achieving the correct task at collective level. Also, we prove and quantify the fact that increasing the candidate set of processes makes it more and more difficult to distinguish the right from the wrong processes independently from the size target set of processes.

The manuscript is organized as follows: In Section II, the basic structures and learning mechanisms are introduced informally, as well as the evaluation issue, which belongs to the class of credit assignment problems; in Section III, the credit assignment problem of parallel processing is formalized showing the correctness of the evaluation metrics proposed and simulation results are provided.

II. PARALLEL PROCESSING EVALUATION

Although the processes presented here are internal processes, they are presented in the context of an agent-environment interaction thus providing a general context. The learning dynamics of the algorithm proposed here is then presented in a simple way. After, credit assignment, a well known generic problem in artificial intelligence, is discussed in the context of parallel processing learning evaluation.

A. Task parallel processing

Fig.1 presents the agent-environment interaction as well as the internal parallel processing. This is a usual representation used in reinforcement learning [7]. As our goal is to set a correct evaluation method of the learning dynamics, let us consider that the environment is a computer program in charge of evaluating the process performances of the agent.

Fig.2 presents the different parallel processing as networks of computational components (or processes) generated and tried over trials. Over these trials, the algorithm should provide a correct evaluation of the right and wrong processes, such that it is ensured that the right processes get iteratively more credit than the wrong ones.

2019 IEEE 18th Int'l Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'19)
P. Soda, R.A. Fiorini, Y. Wang, G. Jacobs, N. Howard, B. Widrow & J. Feldman (Eds.)
978-1-7281-1419-4/19/$31.00 ©2019 IEEE

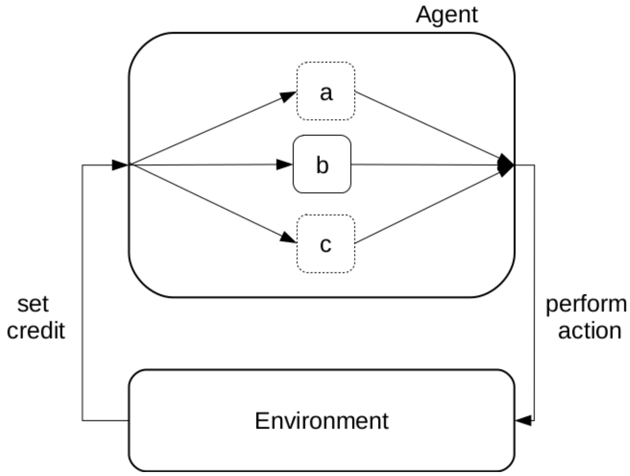

Fig. 1. Interaction between an "agent" performing a cognitive task composed of parallel processes a,b,c in parallel against an "environment". The achievement of the task in the environment provides an evaluation feedback to be used for credit assignment to the processes. Processes can be "right" (in continuous lines) or "wrong" (in dashed lines). Right processes participate positively to the task achievement while wrong ones participate negatively.

B. Credit assignment problem

The credit assignment problem corresponds to the issue of evaluating a system's overall performance due to various contributions of its components [5], [6]. In playing a complex game such as chess, one has a definite success criterion - the game is won or lost. But in the course of play, each ultimate success or failure is associated with a vast number of internal decisions. This is equivalent to learn the player to win the game. The evaluation difficulty consists of being able to incrementally evaluate the steps towards the success. As we will quantify, increasing at each step the number of possible choices makes harder to distinguish between right or wrong decisions.

In the context of credit assignment, the collective credit assignment problem is even harder [7]. This problem can be easily understood by learning to evaluate and compose a good team (at soccer, hockey, etc.) or a good brain assembly to perform a cognitive task. The components of the system (and their interactions) are then so difficult to experiment and model that component credit assignment is a difficult problem. Indeed, it is well known that it is usually not composing a team with good players that a good team is obtained. On the other hand, concerning the brain, the number of neuronal processes is so vast that it is also a challenge to identify them and discriminating the components (as players in a team) that really contribute to the achievement of the task at a collective level.

III. Credit assignment of parallel processes

In a previous work [3], the following conjecture was obtained

Conjecture 1. *Consider any composition whether series, parallel, or feedback in any combination. Let each component have right and wrong alternatives and let the outcome be measured by the number of right selections. If the alternatives are selected uniformly and independently at random in every trial, then the expected credit assigned by ACA to each right alternative always exceeds the expected credit assigned to the corresponding wrong alternative.*

This section provides a first element of proof of this conjecture, considering parallel components. The ACA metric is evaluated for every component in the search space and a proof is provided stipulating that the components of the target network get always, on the long term, a better evaluation than the components which are not in the target network.

Our goal is to set an ACA search algorithm able to generate and evaluate candidate networks of parallel components (processes). The evaluation should permit to determine the target component among the candidate components. Let us define more properly these elements:

Definition 1. *Score of parallel process networks Let C be a* finite set of candidate components *(processes) composed of $n \in \mathbb{N}^*$ elements. A* candidate network *is a sub-set of the set of candidate components, i.e., $N \subseteq C$. The* set of all candidate networks *consists of $\mathcal{N} = \mathcal{P}(C)$. The* target network *is an element of the set of candidate networks, $N^* \in \mathcal{N}$. The* the score*, denoted as* S*, is a measure of the component number difference between the target network $N^* \in \mathcal{N}$ and a given candidate network $N \in \mathcal{N}$. It is defined as*

$$\mathrm{S}(N) = \frac{1}{1+\mathrm{D}(N^*,N)}, \tag{1}$$

where $\mathrm{D}(N^*,N)$ is the cardinality of the symmetric difference between N^* and N

$$\mathrm{D}(N^*,N) = \mid (N^* \setminus N) \cup (N \setminus N^*) \mid = \mid M \cup W \mid$$

This difference counts the number of missing components *M (i.e., components that are in N^* and not in N through the difference $\mid N^* \setminus N \mid$) and the number of* wrong components *W (i.e., components that are in N and not in N^* through the difference $\mid N \setminus N^* \mid$).*

Remark that

1) $\forall N \in \mathcal{N}, \quad 0 < \dfrac{1}{|C|} \leqslant \mathrm{S}(N) \leqslant 1.$
2) $\mathrm{S}(N) = 1 \;\Leftrightarrow\; N = N^*.$

We show how to compute the score of a network through the following example.

Example 1. *Score of a candidate network Let $N^* = \{a,c,d\}$, so a search algorithm in charge of generating candidate networks and evaluating/comparing them with the target network will, e.g., generate and test a candidate network $N = \{a,d,f\}$ at one trial and $N' = \{a,c\}$, at another trial. Score $S(N) = \frac{1}{3}$ because the candidate network N is missing one component, c, and adding one wrong component, f. Score $S(C') = \frac{1}{2}$ because the candidate network N' is missing one component, d.*

While the score provides an evaluation at network level, our goal is to evaluate the components individually based on their trial-by-trial contributions.

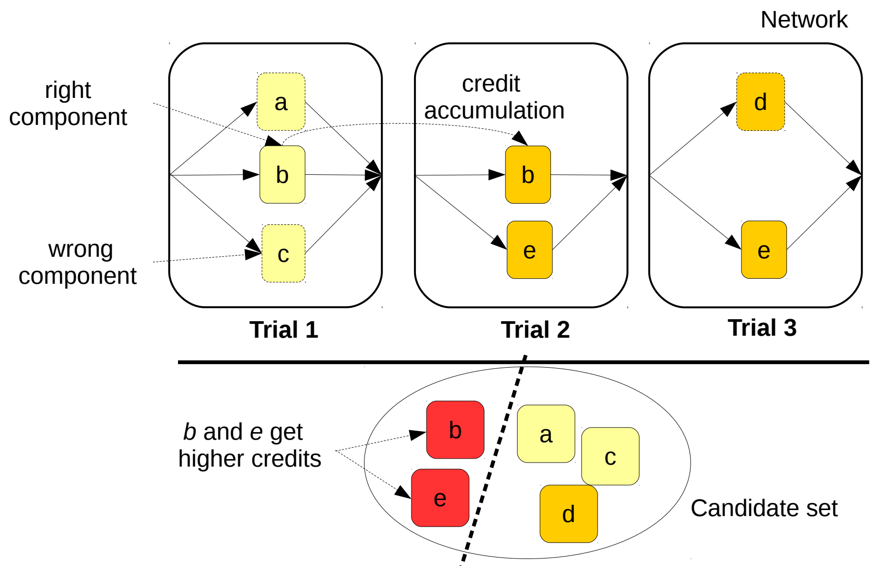

Fig. 2. Processes are "components" of a "network". Different networks are generated and evaluated overs trials. Components accumulate credit over the trials. Increasing credits consist of colors yellow, orange, dark orange, and red. Notice that at Trial 1, as the network gets the two components of the target network the components get more credit (dark orange) than in trials 2 and 3, At the end of the search the "right" components should have a higher credit than the "wrong" ones.

Definition 2. *We define the* average credit of a component *e,* evaluated *in each candidate network $E \in \mathcal{N}$ as follows*

$$\mathscr{cr}(e) = \frac{1}{n_e}\left(\sum_{E\in\mathcal{N}} \mathrm{S}(E)\right) = \frac{1}{n_e}\left(\sum_{E\in\mathcal{N}} \frac{1}{1+D(N^*,E)}\right)$$

Where $n_e = \mid \{E \subset \mathscr{P}(C) \mid e \in E\} \mid = 2^{|C|-1} = 2^{n-1}$ is the number of networks in $\mathcal{N} = \mathscr{P}(C)$ such that $e \in E$. Note that we have always $\mathscr{cr}(e) \in [0,1], \forall e \in N$.
Finally, we define the difference of right and wrong average credits, denoted Δ, the quantity

$$\Delta = \mathscr{cr}(a) - \mathscr{cr}(b), \quad \forall a \in N^*, \forall b \notin N^*.$$

Right (resp. wrong) average credits is simply $\mathscr{cr}(a)$ for $a \in \mathrm{N}^*$ (resp. for $a \notin \mathrm{N}^*$).

Lemma 1. *Considering the set of candidate networks $\mathcal{N}$ as defined in Def.2 and the target network N^*. A search algorithm evaluating all the candidate networks to find the target one, exhibits the following properties:*

1) $\mathscr{cr}(a) = \alpha \in]0,1], \ \forall a \in \mathrm{N}^*$ *(every component of the target network gets the same average credit),*
2) $\mathscr{cr}(b) = \beta \in]0,1[, \ \forall b \notin \mathrm{N}^*$ *(every component which does not belong to the target network gets the same average credit),*
3) Δ *is strictly positive and independent on the choice and the size of the target network N^*.*

Proof 1. *The proof follows three main steps: 1) the credit of a component in the target network, $a \in N^*$, is computed, 2) the credit of a component not in the target network, $b \notin N^*$, is computed, 3) the difference between both is computed. For that, consider a set of candidates C composed of $n \in \mathbb{N}^*$.*

1) *The credit of a component in the target network, $a \in N^*$, consists of averaging the score obtained by all the subsets of C that contains the component a. This set is is given by $\mathscr{P}(C \backslash \{a\}) \cup \{a\}$ which its cardinality is $n_a = 2^{n-1}$. The second remark is that the distance function $D(N^*,.)$ does not take into account the identity of a component, in the sens that one have the same distance, i.e. the same score; given by different candidate networks. The distance verifies a binomial distribution (the number of candidates sets with distance equal to k is C_k^{n-1}.) Hence, we have the average credit of a as follow*

$$\begin{aligned}\mathscr{cr}(a) &= \frac{1}{n_a}\left(\sum_{A\in\mathcal{N}} \mathrm{S}(A)\right) \\ &= \frac{1}{n_a}\left(\sum_{A\in\mathcal{N}} \frac{1}{1+D(N^*,A)}\right), \\ &= \frac{1}{2^{n-1}}\left(\sum_{k=0}^{n-1} \frac{C_k^{n-1}}{k+1}\right) = \alpha \in]0,1[.\end{aligned}$$

2) *For a component b not in the target N^*, we use the same observation as in 1. However, when evaluating the distance of a candidate network in this case to the target network, +1 will always be added to the distance because of b is not present in the target. While the distance still verifies the the binomial distribution, the*

average credit for $b \notin N^*$, consists of

$$\begin{aligned}\mathscr{cr}(b) &= \frac{1}{n_b}\left(\sum_{B\in\mathscr{N}} \mathrm{S}(B)\right)\\ &= \frac{1}{n_b}\left(\sum_{B\in\mathscr{N}} \frac{1}{1+D(N^*,B)}\right),\\ &= \frac{1}{2^{n-1}}\left(\sum_{k=0}^{n-1}\frac{C_k^{n-1}}{k+2}\right) = \beta \in]0,1[.\end{aligned}$$

3) *Let $a \in N^*$ and $b \notin N^*$. We have for all $E \in \mathscr{P}(\mathscr{N})$*

$$D\big(N^*, E\cup\{a\}\big) < D\big(N^*, E\cup\{b\}\big).$$

This implies that

$$\mathrm{S}\big(E\cup\{a\}, N^*\big) > \mathrm{S}\big(E\cup\{b\}, N^*\big), \quad \forall E \in \mathscr{N}.$$

Hence

$$\begin{aligned}&\sum_{E\in\mathscr{N}} \mathrm{S}\big(N^*, E\cup\{a\}\big) > \sum_{E\in\mathscr{N}} \mathrm{S}\big(N^*, E\cup\{b\}\big),\\ &\Leftrightarrow \sum_{E\in\mathscr{N}:a\in E} \mathrm{S}\big(N^*,E\big) > \sum_{E\in\mathscr{N}:b\in E} \mathrm{S}\big(N^*,E\big),\\ &\Leftrightarrow \frac{1}{2^{n-1}}\sum_{E\in\mathscr{N}:a\in E} \mathrm{S}\big(N^*,E\big) > \frac{1}{2^{n-1}}\sum_{E\in\mathscr{N}:b\in E} \mathrm{S}\big(N^*,E\big),\\ &\Leftrightarrow \mathscr{cr}(a) > \mathscr{cr}(b),\\ &\Leftrightarrow \Delta > 0.\end{aligned}$$

This proof ensures that the right components will get more accumulated credit, at individual level, than the wrong components, while ensuring the right global task at network level.

Example 2. *Consider the following set of components: $C = \{a,b,c\}$. And consider the target as $N^* = \{a,c\}$. Hence, the set of all possible candidates network will be $\mathscr{N} = \big\{\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\big\}$. Let us compute the credit of the component $a \in N^*$ and $b \notin N^*$*

$$\begin{aligned}\mathscr{cr}(a) &= \frac{1}{2^2}\big(S(\{a\})+S(\{a,b\})+S(\{a,c\})+S(\{a,b,c\})\big)\\ &= \frac{1}{4}\Big(\frac{1}{1+1}+\frac{1}{1+2}+\frac{1}{1+0}+\frac{1}{1+1}\Big)\\ &= \frac{1}{4}\Big(\frac{1}{2}+\frac{1}{3}+\frac{1}{1}+\frac{1}{2}\Big) = \frac{1}{4}\frac{7}{3} = \frac{7}{12} \simeq 0.583\end{aligned}$$

$$\begin{aligned}\mathscr{cr}(b) &= \frac{1}{2^2}\big(S(\{b\})+S(\{b,a\})+S(\{b,c\})+S(\{b,a,c\})\big)\\ &= \frac{1}{4}\Big(\frac{1}{1+3}+\frac{1}{1+2}+\frac{1}{1+2}+\frac{1}{1+1}\Big)\\ &= \frac{1}{4}\Big(\frac{1}{4}+\frac{1}{3}+\frac{1}{3}+\frac{1}{2}\Big) = \frac{1}{4}\frac{17}{12} = \frac{17}{48} \simeq 0.354.\end{aligned}$$

Hence, the difference of right and wrong credit is

$$\delta = \mathscr{cr}(a) - \mathscr{cr}(b) \simeq 0.23$$

Figure 3 shows the evolution of both score and average credits through trials. On can see that at the beginning of the simulation, the average credit of wrong components is below the one of right components. However, the the average credit of right components quickly exceeds the one of right components. This corresponds to the accumulation of credits through trials that can bee conceived as increasingly reinforcing the right components with respect to the wrong ones through trials (cf. Fig.3).

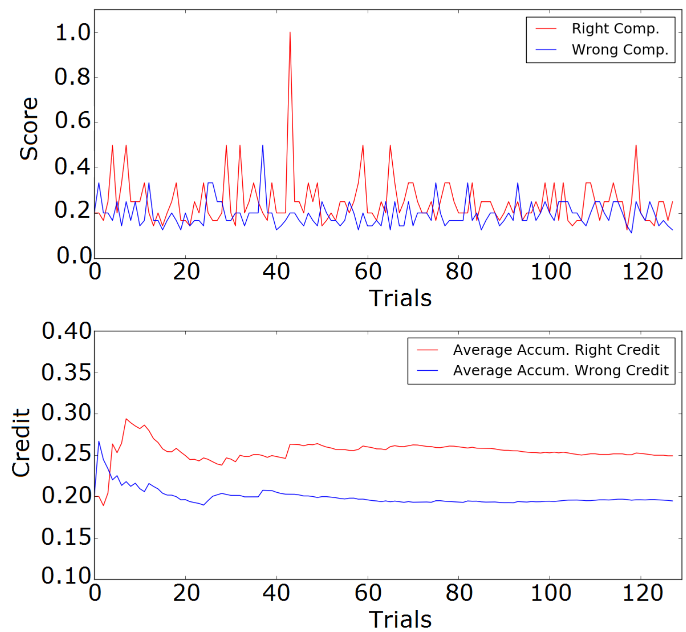

Fig. 3. At the top: Score evolution at each trial for each candidate network $N \in \mathscr{N}$. At the bottom: Average accumulated credits for right and wrong components. The simulation is done using a set of candidates C composed of $n = 8$ components. The average accumulated right credit (resp. wrong credit) was computed using an element belonging to the target (resp. does not belonging to the target).

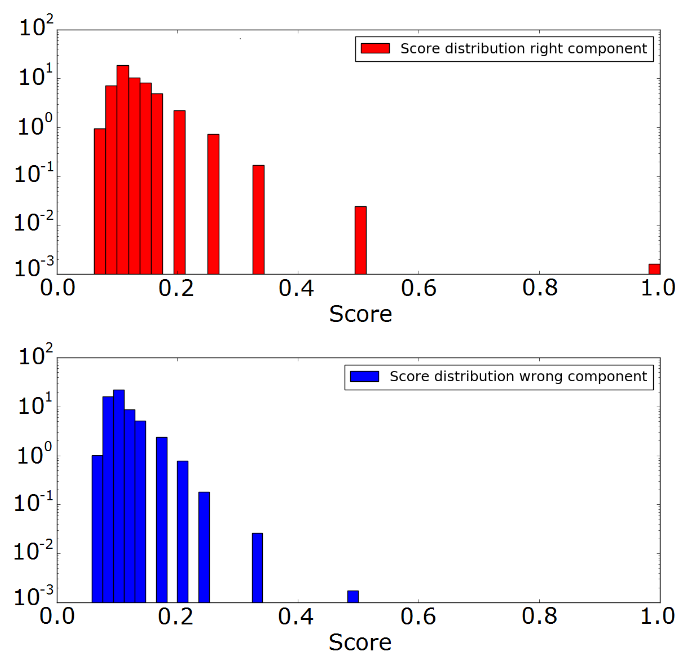

Fig. 4. Comparison of the credit distribution (logarithmic scale) between right (red) and wrong (blue) component. It shows that the evaluation of a right component has more score that a wrong component. This figure was generated using a set of 16 components.

Fig.4 shows the comparison of the score distribution between right and wrong components. It can be seen that higher scores are obtained for the right component, while for the wrong component gets smaller scores during the search. This is due to the fact that for a wrong component, the distance $D(.,N^*)$ will always add $+1$ compared to the right component, and by definition a wrong component does not belong to the target, hence at least one component will be missing. During the search for a right component, it should pass by a candidate network which corresponds to the target network, this can be seen in Fig.3 at trial 42, the score is 1 which means that the distance with evaluated candidate network N was 0 i.e. $N = N^*$.

The previous lemma has the following corollary:

Corollary 1. Δ *decreases as the size* n *of the components set* C *increases.*

Proof 2. *Denote, for a set of components* C *composed of* $n \in \mathbb{N}^*$ *elements,* $\Delta_n = \mathcal{cr}_n(a) - \mathcal{cr}_n(b)$ *as the difference between the right credit and the wrong credit. We stress that the right and wrong credit are both decreasing sequences (sequence with respect to the size of the set of components n). Hence the difference* Δ_n *will be decreasing too as n increases. From 3 in Theorem,* Δ_n *has zero as a lower bound. This gives that* $(\Delta_n)_{n \in \mathbb{N}^*} \searrow 0$ *as* $n \to \infty$.

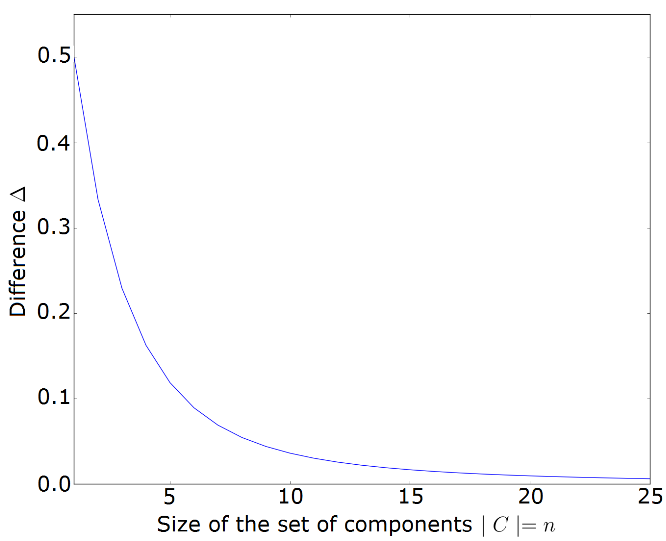

Fig. 5. Difference between right and wrong average credit Δ as a function the size of the set of candidate networks $\mathcal{N}$.

The main implication from previous corollary is that whatever the size of the target network, the difficulty to distinguish right and wrong components, through a credit-based evaluation, increases with the size of the set of candidate networks. On the one hand, this means that, for example, in a set of candidate networks of size 100, finding 2 cognitive processes is as difficult as finding 10 cognitive processes (based on component credit differences). On the other hand, finding 2 cognitive processes in 1000 candidate networks is obviously harder than in 10 candidate networks.

IV. CONCLUSION AND PERSPECTIVES

We presented a first step towards a general solution for credit assignment at cognitive task level. The parallel processes composing the task are evaluated individually accumulating credit through trials. The credit assignment proposed proved to evaluate correctly the right components with respect to the wrong ones while achieving the right cognitive task (represented by the target network of components). The task evaluation is based on a symmetric difference, which is a metric decreasing based on the number of missing and wrong processes (components) present in the candidate task (or network) with respect to the processes (components) present in the target task (or network). The evaluation proved to incrementally mimic the sequences of trials and errors of processes reinforcing the right processes with respect to the wrong ones. To achieve this goal, the symmetric difference proved to be a good partial reward mechanism to finally find the right target task (or network). This way, the whole search dynamics proved to be a good metaphor of learning dynamics.

A single metric was used to evaluate both the processes (at component level) and the task (at behavioral network level). At brain level neuro-imaging is used to correlate task achievement with neuronal activity either for parallel processing [8] or for sequential processing [9]. The brain regions more frequently active when a particular task is achieved are then correlated to the task. Our next perspective is to extend our proof of the ACA conjecture presented in [3] for parallel and sequential processing, based on an activity-based credit assignment [1]. Based on this new metric, the activity of the processes (components corresponding to neurons or brain regions) can be correlated to the score of task achievements. The difficulty then will be to set correctly the score evaluation functions at task level with respect to the evolution of local component activity, in order to get the components rewarded increasingly based on their activity contribution to different tasks.

ACKNOWLEDGMENT

This research is funded by the Computabrain project (http://univ-cotedazur.fr/en/idex/projet-structurant/cauca/projects/computabrain/) of the Université Côte d'Azur.

REFERENCES

[1] A. Muzy & B.P. Ziegler. *Activity-based credit assignment heuristic for simulation-based stochastic search in a hierarchical model-base of systems.* in IEEE Systems Journal, vol. 11, no. 4, pp. 1916-1927, 2017.

[2] B.P. Zieger, A. Muzy & E. Kofman. *Theory of modeling and simulation: Discrete event & iterative system computational foundations.* 3^{rd} edition, Academic Press. ISBN: 978-0-12-813370-5, 2018.

[3] A. Muzy, B.P. Zeigler. *A conjecture from learning simulations of series and parallel connections of components* in THE 26TH EUROPEAN MODELING & SIMULATION SYMPOSIUM, pp. 550-557, 2014.

[4] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[5] Holland , J. H. (2006). Studying complex adaptive systems. In : Journal of Systems Science and Complexity 19.1, p. 18.

[6] Minsky , M. (1961). Steps toward artificial intelligence. In: Proceedings of the IRE 49.1, p. 830.

[7] Eysenck, M. W., & Keane, M. T. (2013). Cognitive psychology: A student's handbook. Psychology press.
[8] Sigman, M., & Dehaene, S. (2008). Brain mechanisms of serial and parallel processing during dual-task performance. Journal of Neuroscience, 28(30), 7585-7598.
[9] Sigman, M., Jobert, A., Lebihan, D., & Dehaene, S. (2007). Parsing a sequence of brain activations at psychological times using fMRI neuroimage, 35(2), 655-668.