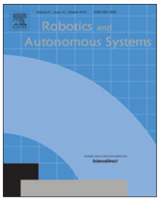
Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Hierarchical reinforcement learning as creative problem solving

Thomas R. Colin [a,*], Tony Belpaeme [a], Angelo Cangelosi [a], Nikolas Hemion [b]

[a] *Plymouth University, Drake Circus, Plymouth, Devon, United Kingdom*

[b] *AI-Lab, Aldebaran Robotics, 48 Rue Guynemer, 92130 Issy-les-Moulineaux, France*

HIGHLIGHTS

- Reinforcement learning's option switches are analogous to psychological insight.
- Insight and options reveal comparable capabilities for transformational creativity.
- Open problems remain: lifelong learning, switching when exploring, option discovery.

ARTICLE INFO

Article history:
Received 28 July 2015
Received in revised form 17 May 2016
Accepted 23 August 2016
Available online 11 September 2016

Keywords:
Creativity
Insight
Hierarchical reinforcement learning
Robotics

ABSTRACT

Although creativity is studied from philosophy to cognitive robotics, a definition has proven elusive. We argue for emphasizing the creative process (the cognition of the creative agent), rather than the creative product (the artifact or behavior). Owing to developments in experimental psychology, the process approach has become an increasingly attractive way of characterizing creative problem solving. In particular, the phenomenon of insight, in which an individual arrives at a solution through a sudden change in perspective, is a crucial component of the process of creativity.

These developments resonate with advances in machine learning, in particular hierarchical and modular approaches, as the field of artificial intelligence aims for general solutions to problems that typically rely on creativity in humans or other animals. We draw a parallel between the properties of insight according to psychology and the properties of Hierarchical Reinforcement Learning (HRL) systems for embodied agents. Using the Creative Systems Framework developed by Wiggins and Ritchie, we analyze both insight and HRL, establishing that they are creative in similar ways. We highlight the key challenges to be met in order to call an artificial system "insightful".

1. Introduction

People achieving extraordinary creative breakthroughs are not born creative; they require extensive hands-on experience to become capable of brilliance in a particular domain. This is not a straightforward result. Indeed, if creativity consists in the production of novelty, one might expect that habits inherited from past experience are a hindrance, and indeed past experience can cause mismatched transfer to new tasks [1]. Nonetheless, there is wide agreement that considerable domain experience is required for people to discover solutions to so-called "insight problems" [1,2] — problems considered difficult precisely because of negative transfer. Animal insight seems no different: macaques with significant laboratory experience are excellent at seeing right through novel experiments in a moment of insight, including for problems in which past experiments seem to discourage the successful behavior [3].

Consider embodied agents, such as robots, situated in an unknown environment, gathering experience from repeatedly interacting with their environment. How can such agents cope with a novel situation, one for which the learned response fails? Their situation resembles that of naive human beings or animals confronted with a previously unseen problem or environment. While some animals, and specifically humans, show remarkable adaptability by creatively developing novel and useful behaviors, artificial agents typically fall short when faced with change. We believe that discoveries from the cognitive sciences offer a way forward, despite the methodological difficulties associated with integrating contributions from multiple disciplines. Below, we integrate results from psychology and machine learning, and suggest a research program for giving artificial agents the capacity to be creative in problem-solving.

Essential to our approach is the notion that creativity is a process — that is, we assume that creativity is a specific manner in which individuals reason and make decisions. We leave aside the domain of artistic creativity which involves socially and culturally

[*] Corresponding author.
E-mail address: thomas.colin@plymouth.ac.uk (T.R. Colin).

http://dx.doi.org/10.1016/j.robot.2016.08.021
0921-8890/© 2016 Elsevier B.V. All rights reserved.

construed value; and we restrict our contribution to the domain of creative problem-solving. In the last 30 years, significant developments have been made in our understanding of human creativity in problem-solving, leading to gradual convergence towards a single integrated theory combining analytic search and insight [2,4].

When tying creativity in with machine learning, it seems that many properties of Hierarchical Reinforcement Learning (HRL) techniques match the description of insight in human beings — including both analytic progress and the ability to restructure the search space. When we analyze HRL and insight as creative systems, using the Creative Systems Framework (CSF) [5,6], these similarities become more striking. This suggests how HRL might be used to produce insightful behavior in artificial agents.

Such an approach is especially relevant to robotic creativity because it is based on control techniques: it manipulates policies in a sensorimotor space, rather than features or parameters in a conceptual space. This distinguishes it from methods used for some of the more abstract domains in computational creativity research, such as e.g. joke invention or musical composition.

Our contribution is two-fold. First, we propose a process-focused theoretical analysis of creativity in problem-solving. We develop this analysis in two disciplines: psychology (insight) and machine learning (HRL). Second, we use the CSF to unveil connections between psychological theories of insight and HRL methods. This sheds light on a novel way to build an agent whose behavior exhibits parallels with human creativity.

2. Assessing creativity

2.1. Concerns for computational creativity

Researchers involved in computational creativity often face two concerns.

The first one is the difficulty of defining creativity. There is widespread agreement on the following working definition: creativity is "the ability to generate novel, and valuable, ideas" [7]. This definition states that something merely novel[1] (alternatively, surprising, original, unusual) is not necessarily creative; otherwise random behavior would be considered highly creative [8]. This justifies the introduction of value or usefulness in the definition. But creativity researchers are aware of the limitations of this working definition; this is especially the case in computational creativity, where precise criteria are needed to assess proposed algorithms. What exactly constitutes sufficient novelty, and in relation to what do we measure value or usefulness?

The second concern is directed at the notion that a machine could be creative. The skeptic's arguments are similar to those disputed by Turing as he ponders the question "Can machines think?" [9]. Turing dismisses the initial question as uninteresting (because it is dependent on the conventional usage of the terms "machine" and "to think" in English), and replaces it with the eponymous test: can a digital computer do well in the imitation game? The substitution clarifies the debate and grounds it in experience. In the same spirit, Boden proposed a version of the Turing Test for computational artwork, which would be passed by creative products "indistinguishable from one produced by a human being; and/or, [seen] as having as much aesthetic value as one produced by a human being" [10].

In an attempt to answer both concerns, Colton and Wiggins [11] shift the burden of finding criteria onto an unbiased observer. They call computational creativity "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative". Below, we discuss this concept of creativity and challenge it in the context of creative problem-solving.

2.2. Products and processes

Boden [8,10] and Colton and Wiggins [11] focus on the end product (behavior or artifact) of creative agents or software. They ask whether the *output* of the algorithm is creative — implicitly excluding any reference to the inner workings of the agent. But their Turing test of creativity, by concentrating on appearances, rewards front-end improvements and variations on a given style over genuine novelty [12]. Recognizing these limitations, theorists of computational creativity have proposed increasingly sophisticated assessment methods for creative outputs [13,14], while also acknowledging the specificity of creative processes [5,15].

We take one step further in that direction. Assessment methods for creativity that focus on end-products, such as those inspired by the Turing test, imply the following:

- There is something special about creative products (i.e. novelty and value).
- By extension, processes are creative when they result in creative products.

We propose to turn this view around:

- There is something special about creative processes.
- By extension, products are creative when they result from the successful use of creative processes.

This resolves some issues with the product view: rather than assessing whether a product is novel, we can check whether it is the result of a copying process; rather than determining the value of a product, we can identify how it was produced, what it was produced for, and how well it fulfills that function. This approach also correctly classifies instances of mere chance as non-creative, even when the end product is indistinguishable from "the real thing" (e.g. when made by the proverbial monkeys with typewriters). And it implies that, should a computer achieve a result that would be called creative if achieved by a human being, but by using a non-creative process (such as exploiting its speed advantage to compute every possibility), we still should not call that computer creative. But this raises a question: what is special about creative processes?

2.3. Creativity as search

Fortunately, the computational creativity community has worked towards characterizing creative processes in a general manner [5,6,16]. Wiggins' Creative Systems Framework (CSF) [5] proposes an analysis of creativity as search, focusing especially on performing search in at least two levels: (1) the search *in* a problem space, and (2) the (meta-)search *of* a problem space.

The first level of search achieves exploratory creativity, such as the analytic discovery of new theorems from axioms, or of a control policy to achieve a task. The second level achieves the more elusive transformational creativity, which consists of a radical change of the domain being investigated; this appears to occur when solving insight problems.[2]

We will consider the simplified version of the framework proposed by Richie [6]. Because the framework aims at clarifying the nature of creative computation, it can be considered a definition of creativity, and that will be our interpretation. That is, any cognitive process that can be accurately described as performing the CSF's search and meta-search, without assistance from a human programmer, can be considered creative. If the framework is indeed

[1] Novel to the creative agent, in what Boden calls "Psychological-creativity" (as opposed to "Historical-creativity") [8].

[2] See [8] for a description of exploratory, combinatorial and transformational creativity.

a good characterization of creativity, it should capture natural examples of creativity, such as evolution or human creativity.

Such a framework can be used to fairly distribute creative credits between the developer and the algorithm: "Did the algorithm explore a search space given to it by its human developer? Was the algorithm capable of modifying that search space, and in what ways?" This should help in the production of algorithms which, perhaps, cannot yet compete with humans in terms of creativity; but nonetheless are, in some sense, genuinely independent creators. We believe such algorithms – rather than those producing Turing-passable outputs – are more promising stepping stones towards computational creativity. In Section 5, we will use the CFS to identify which aspects of the insight process and of HRL algorithms contribute to their creativity.

3. Human creativity: problem solving and insight

3.1. *Creative processes in humans*

An exploration of creative processes should not be limited to those occurring naturally; but such processes, because they have a track record of success and are observable, constitute an important starting point. There are at least two examples of creativity in nature. One is evolution, which has been widely used to investigate creativity in robots [17] and other domains [18]. The other is the human and animal cognitive capacity for creation.

In contrast with the study of evolution in biology, the study of human creativity in psychology has been multifaceted (see [19] for a review). Strands of research have diverged towards issues ranging from neuroaesthetics [19, pp. 305–306], to characterizing entrepreneurial personalities [19, pp. 255–256]. In this profusion of experimental literature, it has become difficult to distinguish fundamental research on creativity as an information-based process; furthermore, efforts at predicting and capturing creativity in an experimental setting have proven difficult [20].

Despite these difficulties, one aspect of the study of human creativity stands out: research on *insight*. Insight is the fast understanding of original, illuminating solutions to problems; it has received increased interest in the last three decades, with a resulting improvement in the scientific understanding of the phenomenon. In the next section, we describe human problem-solving and the role of insight.

3.2. *Human problem-solving*

Solving a problem is transforming a given situation into a desired one. This can be done in the mind, or in interaction with an environment [21]. The vast majority of problems can be understood as consisting of smaller problems, themselves composite, and so on until an atomic granularity is reached; such that any aspect of problem-solving discussed below can be understood to occur for an entire problem, as well as for a part of a complex problem.

Simon and Newell [22] analyze human problem-solving as a search process. A problem-solver works in a problem space, characterized by a set of knowledge states $\mathcal{U}$, an initial state $u_0 \in \mathcal{U}$, which must be transformed into a state u_n such that $u_n \in \mathcal{G}$, where $\mathcal{G} \subseteq \mathcal{U}$ is the set of goal states. This is done using operators from a set $\mathcal{Q}$ [22, p. 810]. However this approach, which lends itself to software implementation, suffers from a crucial limitation: the problem-space (states and operators) must be defined by the programmer, and no provision is given for transforming it in the course of problem-solving. Thus Simon and Newell's approach accounts for exploratory creativity, but not for transformational creativity. If a problem-space makes use of inappropriate representations for the problem at hand, perhaps grouping features in an improper way, or excluding certain operators, no amount of heuristic search can arrive at a solution.

The Gestalt tradition [23] analyzes human and animal problem-solving according to a different paradigm. In this view, problem-solving is not understood using the metaphor of sequential movements in a problem-space. Instead, gestaltists emphasize changes in the representation of the problem as a whole that transform a problematic situation into a practically solved one. For example, Köhler [24] had chimpanzees attempt to retrieve inaccessible bananas; in order to solve the problem, they had to realize that tree branches in their enclosure could be broken off and used as sticks, after which the solution was trivial. Insightful problem-solving has since been observed in orangutans [25] and corvids [26] –confirming the reality of a phenomenon that is difficult to explain as Thorndikian trial-and-error [27]. However, although restructuring constitutes transformational creativity, Gestalt theory gives little in the way of explanation of the cognitive mechanisms responsible for it.

The phenomenon of insight in problem-solving is better understood by considering its relation to both paradigms: Simon and Newell's search and Gestalt restructuring. Problem-solving with insight follows the sequence [2,4]:

1. Search in a problem space
2. Consistent failure (impasse)
3. Restructuring, and solution or significant progress
4. Test of perceived solution.

The third step is known as insight, popularly known as the "Aha!"-moment, giving the problem-solver the impression that significant progress has been suddenly achieved [28]; sometimes that impression is mistaken, hence the necessity to test it in the fourth step.

Insight is not necessary for exploratory or combinatorial creativity; for example, theorems can, in principle, be discovered through pure heuristic deduction, progressing smoothly and without restructuring the problem space. Yet insight is accepted as at least one of the cognitive processes involved in human creativity, at the stage of generating new ideas or behaviors [19]. Some have called it the exclusive mark of creativity (e.g. [2,20]) on the basis of its ability to transform the problem space.

3.3. *Restructuring*

The key to understanding insight is the restructuring process — what causes it, and what it consists of. In Gestalt theory, restructuring was thought to arrive at a new representation through viewing the problem naively, rejecting the contribution of experience [23]. This view seems untenable due to the sheer computational complexity of that task, and the speed at which it occurs in humans; it fails to account for experimental evidence showing that insight is more likely to occur with prior experience [1,3] and may even be impossible to subjects lacking domain experience. For example, [29] reports a success rate below 5% on the 9-dot problem (cf. Fig. 1), even when preventing fixation; whereas [30] demonstrated success on the same problem following acquisition of relevant experience.

So what is restructuring? It includes several modifications affecting the problem space, often difficult to tell apart from one another, each of which has been experimentally demonstrated:

1. The heuristics used to select operators [31];
2. The representation of the problem, e.g. the chunking of perceptual elements into objects [32] (cf. Fig. 2);
3. Constraints on available operators [33].

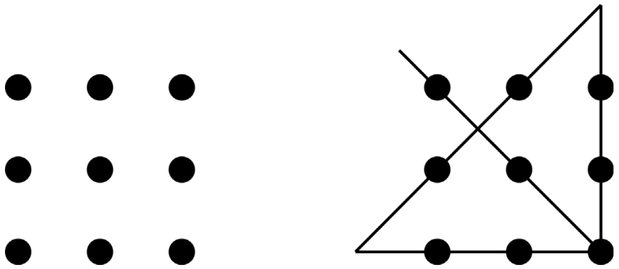


Fig. 1. Left, the 9-dots problem: subjects are instructed to draw four straight lines connecting all dots, without lifting their pen. Right, the solution, which involves drawing lines outside the square formed by the dots and turning on non-dot points.

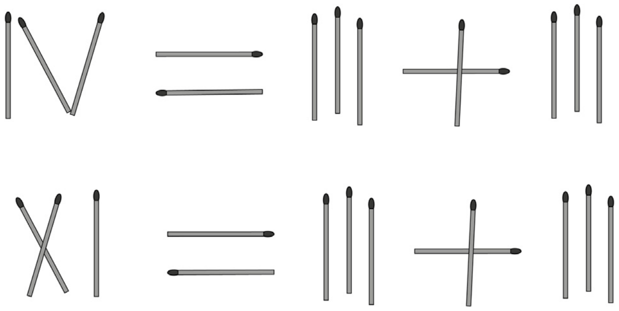


Fig. 2. Two of the problems used in [32]. The objective is to transform an incorrect equation into a correct one by moving a single matchstick, where forming the sign "≠" is prohibited. The solution to the first problem consists in changing "IV" into "VI", and in the second problem, "XI" into "VI". The second problem proves to be more difficult for human subjects. The authors argue that this is due to "chunking": subjects make two chunks for "IV", whereas "X" is perceived as a single element, thus inhibiting search on the latter.

Whatever its form, restructuring can be triggered either from new information obtained in the course of problem-solving (in which case problem-solving can still be described as analytical), or from failure to succeed using the initial problem-space (this is insight "proper"). Both are interesting in the context of creative problem-solving.

3.4. A theory of restructuring

From an information processing perspective, the most elaborate description of the restructuring process is given by Ohlsson [2, p. 108–109], illustrated in Fig. 3. Below, we summarize Ohlsson's account:

- **Architecture:** Problem-solving is based on perception of a current state (e.g. the "state" units in Fig. 3). Output from this state perception is propagated in a series of selective layers of processing units — where a processing unit could itself be a neural network. Each of these processing units receives weighted inputs from units in previous layers, and forwards weighted outputs to units in further layers. Each unit connects to units on the next layer, potentially activating them, while also providing relevant problem-related information. All initial weights are learned prior to problem-solving based on past successes and other factors, providing the problem-solver with preferences based on experience.
- **Within-layer dynamics:** In the course of problem-solving, units are activated (when the sum of inputs is above a certain threshold) and deactivated (when the sum of inputs is below threshold). When an activated unit proves unsuccessful (based on interaction with the environment, planning, or heuristic estimation), it receives negative feedback reducing its level of activation. On top of exciting forward connections and inhibiting feedback, these layers have internal dynamics implementing Winner-Take-All behavior [34]. Thus when a unit is de-activated, within-layer dynamics (in layers WTA#1, …WTA#n in Fig. 3) cause an alternative unit to activate instead.
- **Between-layers dynamics:** Negative feedback is also propagated to previous layers. When the feedback propagated to previous layers leads to deactivating one unit, all of the dependent subsequent activation must be redistributed accordingly –this is restructuring understood as redistribution of activation. Depending on the amount of affected subsequent activation, such restructuring can have a large or small impact. When the entire network is affected, this can be interpreted as the cognitive basis of a strong "Aha!"-experience.

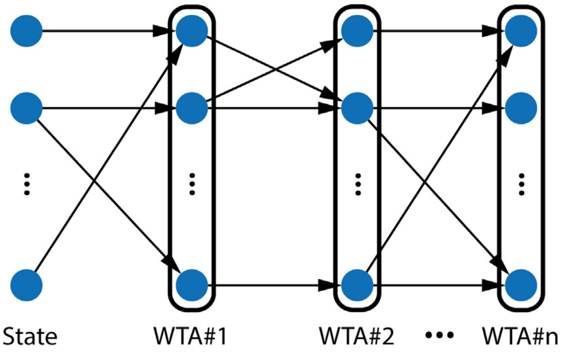


Fig. 3. A possible visualization of Ohlsson's theory [2]. A series of Winner-Take-All (WTA) layers of processing units serves to make decisions based on a perceived state and forward connections. Connection weights are determined by past experience. Unsuccessful options cause negative feedback to backpropagate through the layers, reducing activation. This can push a WTA layer to switch activation to a different option, triggering a chain reaction of redistribution of activation in the units of subsequent layers.

Because processing units can also manage representations, this theory accounts for representational change — including sudden, large-scale changes. Further, as the weights are learned from experience, each unit comes with bias corresponding to a search heuristic. Finally, the theory also describes instances of systematic search (since units of a given layer are activated sequentially unless a change occurs in a previous layer or a solution is found), and, via state changes, it accounts for cases of analytic thinking [4] in which restructuring is triggered by the discovery of new information during failed attempts.

3.5. Implementing insight

The information-processing account of insight by Ohlsson, described above, is not backed up by neuroscientific evidence; it is hard to see how any such evidence could be produced in the near future, as insight is particularly difficult to experiment with: it is individual-dependent, and can happen only once per insight-problem[3] [20]. Thus Ohlsson's theory is best understood as spelling out a hypothesis as to the sort of process that may account for insight. The level of detail of the theory is roughly consistent with the amount of evidence available; this leaves it lacking as a blueprint for an algorithm. In the remainder of this article, we discuss ways in which this sort of processing can be implemented,

[3] There is however a growing body of work in neuroscience with respect to RL [35], HRL [36], creativity and insight [37,38].

starting with computational models explicitly designed to account for insight phenomena.

Simon claims [39] that various extensions of the General Problem Solver [40] are sufficient to account for insight: these programs are capable of generating a representation from a suitably constrained description, and of switching between heuristics and levels of abstraction. However, Simon fails to show how these various abilities can be integrated into a single program, or how any of these solutions could scale to the problems that human or animals tackle. More recently, C.J. MacLellan [41] proposed a model of human insight, which he tested on the nine-dot problem; however, the model does not account for learning or for representational restructuring (focusing instead on heuristics).

We argue below that the algorithms that provide the best match with the psychological insight process are not found in the fields of computational creativity or cognitive modeling. Rather, they emerged from research in machine learning, and more specifically reinforcement learning. To our knowledge, only Vigorito and Barto [42] and Smith and Garnett [43] have claimed that such algorithms are creative in a manner resembling human creativity — and neither have mentioned any link to insight (nor have insight researchers in psychology made any explicit link to computational reinforcement learning). Below, we introduce hierarchical reinforcement learning (HRL) and present the ways in which it achieves many of the properties of insightful problem-solving.

4. Reinforcement learning

4.1. Background

Reinforcement Learning (RL) is the problem of acting in an unknown world to maximize the sum of future (discounted) rewards, or return. The RL problem is typically formalized as a Markov Decision Process (MDP) consisting of a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where:

- $\mathcal{S}$ is a finite set of states,
- $\mathcal{A}$ is a finite set of actions,
- $\mathcal{P}$ is the state transition probability function, such that $\mathcal{P}(s'|s, a) = \mathbf{Pr}(S_{t+1} = s'|S_t = s, A_t = a)$,
- $\mathcal{R}(s, a, s')$ is the scalar reward,
- $\gamma \in (0, 1]$ is a discount rate on future rewards.

The goal for the agent is to maximize the return $G_t = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}(s_{t+k}, a_{t+k}, s_{t+k+1})$ from its current state s_t. A strategy is to attempt to discover and use the optimal policy π^*. By definition, π^* picks with probability 1 the action (assumed unique for clarity of exposition) expected to yield the largest return when the agent keeps on acting optimally thereafter:

$$\pi^*(s_t, a_t) = \begin{cases} 1 \text{ if } a_t = \arg\max\limits_{a \in \mathcal{A}}(\mathbb{E}_{\pi^*}(G_t|s_t, a)) \\ 0 \text{ otherwise.} \end{cases}$$

Popular techniques for discovering π^* include direct policy search and temporal difference (TD) algorithms. Note that in real-world settings such as robotics, policies rarely make use of discrete state–action pairs (s, a), but instead approximate them using linear or non-linear supervised learning techniques, for instance neural networks with backpropagation.[4]

In humans, creativity has an improvisational, interactive nature [47,48]: humans do not wait for the finished result or product to evaluate it (contrary to most optimization techniques, for instance genetic programming [49]), but instead maintain an evaluation of the future outcome, and use it to direct and adjust their behavior. Likewise, most RL algorithms make explicit use of temporal structure. For instance, in TD algorithms policies can be altered on-line in the course of solving the problem at hand, rather than at the end of an episode; and (biased) estimated return can be used to "bootstrap" towards a promising solution. Thus the execution of any action immediately generates a novel experience for the learner, and dynamically affects its way of behaving. This is important to achieve good performance in large problem spaces where experience is costly and episodes take considerable time to complete, such as in robotics or in creative domains. But in order to achieve creativity, another component is required: the generation of novelty, which is discussed next.

4.2. Novelty, exploration, and curiosity

In psychology, the interplay between "divergent thinking" and "convergent thinking" is a widely recognized characteristic of creativity [50,51]. In particular, this interplay has been theorized as "blind variation" and "selective retention" [52,53]. This is strongly reminiscent of another essential aspect of RL algorithms: the dilemma between exploration and exploitation. Indeed, to gather rewards, an RL agent must use the actions that have proven useful in the past; but in order to improve its policy, it must try out new actions and observe their consequences [45,54]. Thus, rather than greedily using the estimated best policy $\hat{\pi}_t^*$, the agent must take the need for exploration into account, and decide to miss out on predictable rewards in exchange for observing something new. When and how to explore is one of the main open problems in RL; we believe any answer to that question has important implications for a theory of creativity. In this subsection, we briefly review some approaches to exploration in Reinforcement Learning.

The most straightforward and still most commonly used technique is to employ a *soft* version of $\hat{\pi}^*$; popular implementations are ϵ-greedy and SoftMax action selection. For example, an ϵ-greedy agent ($0 < \epsilon < 1$) picks actions according to $\hat{\pi}^*$ with probability $1 - \epsilon$, and otherwise picks another action at random. Since exploration occurs on a fraction of time-steps, these algorithms tend to focus exploration around promising trajectories [55]. This restricts the state space in a manner that can be beneficial (ignoring seemingly irrelevant regions), but in the absence of other mechanisms it can also be detrimental (intensifying exploration around an already well-known policy). Several methods have been proposed to diversify exploration.

Many techniques are based on encouraging an exhaustive or near-exhaustive exploration of the state space, typically by keeping count of state visits, as in the popular R-MAX algorithm [56]. The sample complexity of these techniques grows linearly with the size of the state space [55], but by relaxing the exhaustiveness of this approach (ceasing to explore when "good enough" behavior has been discovered), it can produce good results even in robotics domains [57]. Nonetheless, it seems insufficient to deal with creative domains characterized by very large problem spaces; in such domains, systematic exploration – even in simulation – could last a (robot's) lifetime without discovering anything "good enough". Further, exhaustiveness is (intuitively) antinomic with creativity: surely an efficient search would explore only the most promising states.

Exploration does not have to be random or exhaustive — it can be guided by the expectation of learning. This has been done in two ways, which we will label intrinsic motivation [54,58] and artificial curiosity [59]. The latter can be understood as a special case of the former.[5]

[4] This often introduces complications, for instance because the sequence of samples is not independent and identically distributed. Describing these algorithms is beyond the scope of this article; see [44–46].

[5] According to our taxonomy; in the artificial intelligence literature, "intrinsic motivation" and "artificial curiosity" are often used interchangeably.

Table 1
Characteristics of insight and their HRL counterparts.

Insight	HRL
Integrated with analytic processes [4]	Based on the standard RL framework [70]
Apparently discontinuous progress [28]	Explorative "jumps" [42]
Operator constraints changes [33]	Operator constraints changes [67]
Heuristic change [31]	Option-dependent initial value function [70]
Representational change [32]	Option-dependent state abstraction [68]

Intrinsic motivation in RL, in the framework proposed by Singh et al. [58], consists in modifying the reward function to improve the performance of an agent. Whereas the traditional approach to RL is to provide reward exclusively when the goal is achieved, intrinsically motivated agents also receive "shaping" rewards whenever they encounter a state associated with learning (sometimes called "salient states"). These shaping rewards can be provided by the programmer, or, for example, learned via an evolutionary algorithm over several generations of agents.

Artificial curiosity [59] is also based on rewarding learning. An artificial agent is "curious" when it uses information-theoretic measures to detect and predict learning or surprise, and receives reward when that occurs. Implementations have used various such measures. Examples include rewarding state prediction error (surprising events) [60,61], prediction improvement [62], or competence improvement [63]. These systems typically include a learning mechanism to compute the amount of surprise or learning that the agent undergoes, whereas intrinsic motivation can often rely on a static reward function. Some of that work investigates optimal exploration from a Bayesian perspective [64], although that remains unfeasible for the general RL problem [65, pp. 25–28].

Below, we consider a fourth kind of method to improve exploration, which consists of using the structure of the problem-space to increase the efficiency of exploration. When the structure is learned (rather than provided to the agent), these methods can be said to allow an agent to learn to explore. This is achieved by varying the temporal granularity and the type of exploratory actions using behavioral hierarchy. These methods are often fruitfully combined with curiosity or intrinsic motivation; as a result, their contribution to exploration often goes unnoticed [66].

4.3. Modular exploration of structured environments

Behavioral hierarchy in reinforcement learning [67–69] is commonly formalized using Sutton's options framework [70]. Options are a generalization of primitive actions that include temporally extended, closed-loop courses of action. They consist of three components: a policy $\pi : \mathcal{S}\times\mathcal{A} \rightarrow [0, 1]$, a termination condition $\beta : \mathcal{S} \rightarrow [0, 1]$, and an initiation set $\mathcal{I} \subseteq \mathcal{S}$. An option $\langle \mathcal{I}, \pi, \beta\rangle$ is available in state s_t if and only if $s_t \in \mathcal{I}$; if taken, its policy π is executed until it stochastically terminates at s_{t+n} (this can occur with probability $\beta(s)$ at any visited state s). In HRL, an agent consists of a hierarchy of at least two levels of options, the higher levels delegating tasks to lower levels, until an atomic option (equivalent to an action in "flat" RL) is executed. In the next paragraphs, we detail the various benefits of options.

An important advantage of options is the possibility of using different state abstractions depending on the option [71–73]. This allows the policy of an option to decide on actions using only relevant features — regardless of which features are useful within other options. This means that exploration can make use of these abstractions: depending on the currently running option, the agent takes into account different perceived features of the environment when learning from an exploratory move.

Options are also useful for their ability to re-use sub-policies between different tasks [74] — a property much discussed in the emerging research area of transfer learning in RL [75]. Because options are closed-loop, can use approximate value functions and learn on-line (depending on the learning algorithm), they can adapt to being used in a slightly different context — such that a new (but related) problem can be explored using an option learned in earlier problems; furthermore, on-line learning can allow for resolving minor differences in the new problem. Note that this resembles analogy-making as described in work on neural-symbolic integration [76] where a similar sequence of abstraction, transfer, and repair is found.

Most importantly with respect to creativity, options provide temporal abstraction, allowing for exploring the state space at multiple granularity sizes. An obvious advantage is the reduction of processing costs in planning. But this also makes it possible to reach otherwise unattainable sections of the state space — especially in environments where undirected or unmotivated exploration does not allow for reaching some states [42]. Many real life tasks are of this form: for most robot models, the state "broken" is absorbing and accessible from many other states, preventing both random walks and exhaustive search from reaching far into the state space.

4.4. Hierarchical reinforcement learning and insight

In Section 3, we introduced the psychological process of insight in the context of problem solving; we have now also introduced the options framework in reinforcement learning. These two processes are both used to solve problems, and they do so in remarkably similar fashion.

Table 1 shows the correspondence between the properties of insight, discussed in Section 3, and those of modular exploration in HRL, discussed above. It appears that the conceptual framework of HRL allows for the sort of restructuring observed in human beings during insight problem-solving, including value functions and state abstractions that are option-dependent, as well as explorative jumps. Single systems have exhibited many of these properties at once — e.g. in the MAXQ framework [68] or the options framework [77]. In this last case, the options were learned by the agent.

Besides apparent algorithmic similarities, there is a similarity in the sort of problem-solving behavior that can emerge in insightful agents and HRL agents. Table 2 draws analogies between the insight sequence and the behavior of an exploring hierarchical reinforcement learning algorithm.

In this subsection, we have presented an informal comparison of insight and HRL as described by their respective specialists, summarized in Tables 1 and 2. However, these similarities are only relevant to our argument if they actually contribute to the *creativity* of the underlying processes. It is possible to perform a more principled analysis of these two problem-solving processes based on a framework equipped to assess similarity in terms of creativity: the Creative Systems Framework (CSF), introduced in Section 2.3, and detailed below.

5. Analysis of insight and HRL using the CSF

5.1. The creative systems framework

The CSF formalism is inspired by Boden's philosophical discussion of creativity [8], and its aim is to describe and assess creative software [5,6]. The CSF analyzes creativity as search; its main

Table 2
The insight sequence and its HRL counterparts.

Stage	Observed behavior	HRL
Problem perception	Read text; manipulate material; etc. Understand what the problem is about	Pick high-level policy
Problem-solving	Regular progress with occasional trial and error	Transfer of high-level policy, resolve errors
Impasse	None	Encounter negative time-difference errors, re-evaluate high-level policy
Restructuring	Report change in strategy and perception	Switch policies
Insight	Exclaim “Aha!” etc.	Transfer new policy, encounter positive errors
Post-insight	Resume problem-solving, sometimes fail	Finish transferring new policy, sometimes fail

contribution, in our view, is to formalize the distinction between object-level search and meta-level search, corresponding respectively to exploratory and transformational creativity. In doing so, it reveals the characteristics of so-called “creative” systems that are constitutive of their creativity. It is thus a useful tool for assessing how two distinct systems relate to one-another with respect to creativity. Below, we introduce the CSF and use it to analyze the insight process on the one hand, and HRL on the other hand.

The CSF describes creative processes at two levels [5,6]. The first is the **object-level** of exploratory creativity, and is defined as a tuple $\langle \mathcal{P}, \mathcal{N}, \mathcal{V}, \mathcal{Q} \rangle$, where:

- $\mathcal{P}$ is the set of possible products.
- $\mathcal{N} \in [0, 1]^{\mathcal{P}}$ is the acceptability mapping[6] (specifying how acceptable a product is).
- $\mathcal{V} \in [0, 1]^{\mathcal{P}}$ is the value mapping (specifying how valuable a product is).
- A mapping $\mathcal{Q}$ from $[0, 1]^{\mathcal{P}} \times [0, 1]^{\mathcal{P}}$ to the set of mappings from tuples($\mathcal{P}$) to tuples($\mathcal{P}$) is called the *exploration scheme*. Intuitively, $\mathcal{Q}$ is the set of possible ways to generate a new set of products based on the previous generation of products, taking into account their acceptability and value.

In addition to the object-level, the **meta-level** accounts for transformational creativity: exploration between problem-spaces. A meta-level creative system for $\mathcal{P}$ consists of:

- $ECS(\mathcal{P})$, a set of triples $\langle N, V, Q \rangle$ where N, V and Q are possible values for $\mathcal{N}$, $\mathcal{V}$ and $\mathcal{Q}$ respectively. We will call this $\mathcal{P}^{\text{meta}}$ for coherence.
- $\mathcal{N}^{\text{meta}} : \mathcal{P}^{\text{meta}} \rightarrow [0, 1]$, the acceptability mapping.
- $\mathcal{V}^{\text{meta}} : \mathcal{P}^{\text{meta}} \rightarrow [0, 1]$, the value mapping.
- $\mathcal{Q}^{\text{meta}}$, the structured search for elements $p \in \mathcal{P}^{\text{meta}}$, similar to $\mathcal{Q}$ but operating on elements of $\mathcal{P}^{\text{meta}}$.

5.2. CSF and insight

As the theory of insight is not a formal theory, we can only relate it loosely to the CSF. However, the roots of the CSF [8,16] in human creativity on one hand, and in search on the other hand, make it easier to establish links between the formal components of the CSF and the components of human-problem solving.

The object-level of the CSF maps to search without restructuring, that is, to human problem-solving as described by Simon and Newell [22], in which previous experience provides the learner with an appropriate problem-space.

The meta-level maps to search between problem spaces achieved during insight moments. Those are the cases in which the initial approach fails, and redistribution of activation is required to yield a novel strategy; these consist in switches between units at more abstract layers and subsequent changes in Ohlsson’s model of insight [2] as described in Section 3.4. The various forms of restructuring found in human insight must thus be included in the search operator $\mathcal{Q}^{\text{meta}}$ of the CSF.

The above analysis of insight can be misunderstood as follows: if insight consists in restructuring the problem based on past experience, one might claim that novelty is missing [4]. In this view, the problem-solver is merely making successive interpretations of the problem in several already known problem-spaces. However, insight is required precisely when a perceived problem fails to fit the problem-space it is immediately associated with; that is, the creativity of insight consists in associating an unlikely policy to a novel problem. Furthermore, restructuring often does not succeed at immediately providing a solution. Additional adaptations are required to achieve an adequate fit between the perceived problem and the tentative representation and policy. These further changes constitute search within the new object-space obtained following restructuring, and show that our description of insight accounts for genuine novelty.

5.3. CSF and HRL

For HRL, we propose the following interpretation of what Wiggins [5] calls the object-level search, and which we might call the policy-level search in the context of a creatively behaving robot:

- $\mathcal{P}$ is the set of possible policies.
- $\mathcal{V}$ is the discounted return G_t from executing a policy from the current state s_t (squashed to $(0, 1)$ for compatibility). In non-artistic creative domains, it is not clear that two distinct variables, value and acceptability, are required; therefore we will simply assume that $N \in \mathcal{N}$ maps all policies to 1, deeming them always “acceptable”.
- $\mathcal{Q}$ is the method by which a set of policies and their observed discounted return are used to generate new policies. Most time-difference algorithms make use of a *critic* which estimates the value of intermediary states or state–action pairs, and which can be used e.g. to modify a policy on-line and bootstrap. To account for this in the CSF, we can consider each policy change as a function of previous policies and experience.

Thus an RL search method $Q \in \mathcal{Q}$ is characterized by all properties not obtained from experience. This includes any fixed hyper-parameter (learning rate, exploration rate or temperature, initial values of any parameter...), any modifications made to the sensory system of the agent (e.g. hand-designed features) for the purpose of assisting the algorithm, and so on.

Below we describe the meta-level, which searches through pairs (V, Q) where V, Q are possible values for $\mathcal{V}$, $\mathcal{Q}$ respectively. Surely a random trial-and-error exploration of all RL algorithms is possible, but intractable even for easy problems. In the case of insight, restructuring was a result of interacting with the problem, but was also based on alternative strategies provided by experience on other problems. We suggest adopting the same

[6] For sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{B}^{\mathcal{A}}$ is the set of mappings from $\mathcal{A}$ to $\mathcal{B}$ — thus $\mathcal{N} \in [0, 1]^{\mathcal{P}}$ assigns a real value $x \in [0, 1]$ to each element in $\mathcal{P}$.

Table 3
Overview of the CSF analysis of insight and HRL.

	Insight	HRL
Search	Search in the current problem-space as per e.g. Simon and Newell [22].	Exploitation, exploration and learning within the current option.
Meta-search	Restructuring based on prior experience as per Ohlsson [2].	Switch to a different option and transfer.

interpretation in the case of reinforcement learning: considering previously learned options as potential problem-spaces. Thus:

- $\mathcal{P}^{\text{meta}}$ is a set of pairs $\langle V, Q\rangle$ provided by experience, where Q does not vary along various (hyper)parameters, but makes use of a unique option $\omega = \langle \mathcal{I}_\omega, \pi_\omega, \beta_\omega\rangle$.
- $\mathcal{V}^{\text{meta}}$ is the value of $p \in \mathcal{P}^{\text{meta}}$, applied to the current problem.
- $\mathcal{Q}^{\text{meta}}$ is the method by which pairs $\langle V, Q\rangle$ are modified or retrieved from experience.

Under this interpretation, HRL is creative when it learns options, uses them to set-up the initial characteristics of search, and browses through them when the initial attempt fails. The extent to which an HRL algorithm is creative thus depends on the variety of options that the agent can learn and use for transfer, and how much those options can affect the object-level search. We have seen that options can vary in terms of the initial policy and value function and in terms of temporal and state abstraction. This covers much of the restructuring observed in humans during the insight process. This makes HRL a promising candidate for human-like embodied artificial creativity.

In analyzing both the insight process and HRL using the CSF, we have given a more detailed look at HRL as "search" and "meta-search". This has shown that the analogy between HRL and insight relies precisely on the aspects of each method that are relevant with respect to creativity (according to the CSF). While this section provided a more detailed analysis, Table 3 summarizes the outcome succinctly.

5.4. Modular exploration as creative search

Fig. 4 demonstrates the analogy between insight and HRL on a simple navigation task. At the top level is the option space, or meta-level: when presented with a new problem, an RL agent can select an option, in the same manner that it would normally select an action, based on its expected return in the problem at hand. Because RL allows for online updates to the policy, the option can, in turn, be used as the starting point for exploration when the known policy does not succeed right away (4a). This corresponds to object-level search.

When a switch between options occurs on-line, in the course of the search process, transformational creativity occurs — similar to moments of insight in humans or animals. If the agent is performing model-free trial-and-error, something resembling Fig. 5 can be observed; whereas if the agent makes use of planning, the option change is not likely to have any observable behavioral effect — except perhaps the utterance of an "Aha!".

6. Challenges

Architectures close to those discussed above have been implemented, including on a navigation task similar to the one shown in Fig. 4 [67], and on real robots [77]. However, there has not been, so far, any simulated or embodied agent capable of displaying insightful behavior resembling what is observed in primates or corvids [25,26]. This section discusses the state of the art and the remaining challenges to making creative robots that use modular exploration.

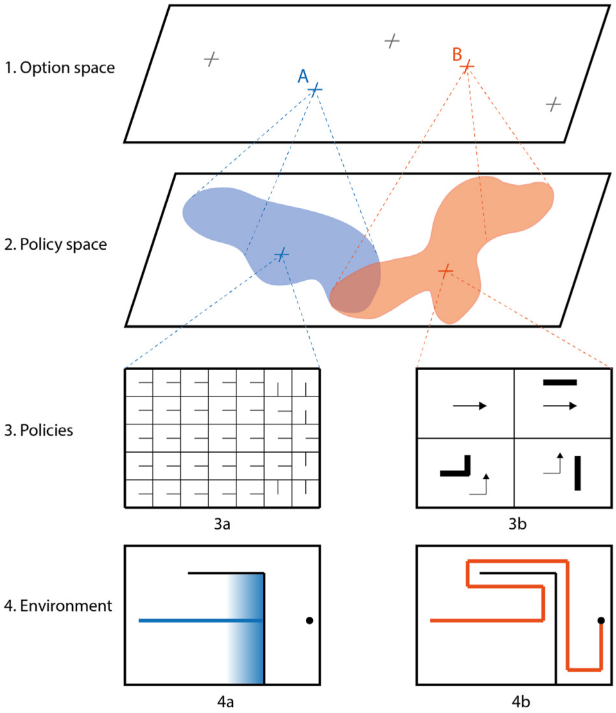


Fig. 4. 1. Options can constitute a starting point for problem-solving. When faced with a new problem, an agent may start its search in an "option space" containing re-usable skills. Each skill corresponds to a representation of the environment and policy. 2. The policy space is the space of all possible policies within an environment. An option presents an initial solution, but also contains information about the expected value of alternative behaviors, and is encoded using certain state abstractions and temporal abstractions. Given an environment, and a time-horizon (a limited lifetime for trial and error, or processing speed for planning), a different subset of policies is reachable, depending on whether the agent initially uses option A or option B - this is shown as two areas of the universal policy space corresponding to A and B. 3. In this example, the agent is faced with a navigation problem: the objective is to reach the black dot. The policy corresponding to option A (3a) is based on the coordinates of the agent in the plane, as in a grid-world. The policy corresponding to option B (3b) is based on following walls. 4. The representation used for option A makes the obstacle essentially invisible as no perceived features stand for walls — in the course of learning, the agent will repeatedly bump against the wall while modifying its policy. In (4a), the agent does not succeed within the time-horizon. In contrast, option B is immediately successful (4b).

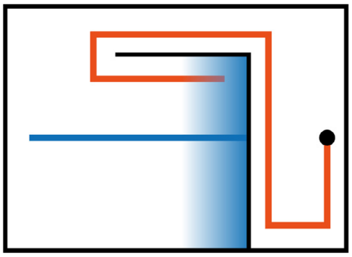


Fig. 5. Switching options online, based on negative temporal difference errors, could lead to behavior resembling human and animal insightful behavior: from successful problem-solving, to impasse, to representational change and eventual success.

- **Lifelong learning:** Comparatively little research within the RL community is dedicated to systems capable of *life-long learning*. While similar in some ways to transfer learning, life-long learning [78] differs in several crucial ways because agents are not told when a problem starts [75]: thus, if they learn transferable skills, they must find how to organize previous experience into discrete skills; and when applying such skills, they must not only solve problems, but also identify problems to solve. However, perhaps due to the recent gain of interest for transfer learning and developmental robotics, there have been renewed efforts in this domain, e.g. [79,80].

- **Option switching for exploration:** We have presented the moment of insight as similar to using a high-level exploratory action after an impasse; that is, the failure of an option to achieve its goal despite repeated attempts. Some hierarchical reinforcement learning algorithms (e.g. [81]) use continued failure as a terminal condition for options. How to optimally explore with options, however, is to date not fully understood: switching between options as soon as a different option has higher expected return might lead to back-and-forth switches, preventing the large jumps afforded by using options in the first place. What is the optimal amount of perseverance?
- **Problem-solving speed:** Reinforcement learning is used to *learn policies*, not merely to *solve problems*. Learning fast can lead to instability and catastrophic forgetting; but learning slowly leads to performances hardly compatible with creativity. Is there a middle-ground? Work on transient learning and tracking hints that "learning" fast, and forgetting most of it, might be a way to implement search [82,83]; however this assumes that the agent knows when problems start and end.
- **Option discovery:** Although many methods are being proposed and implemented [77,79,84], option discovery remains an open problem for HRL, with no generally accepted and widely successful solution to date; much progress can still be made in this area. A promising new approach is offered by Bacon [85].

7. Discussion and conclusion

We offered a novel perspective on computational creativity, highlighting the process of creativity instead of the end product. While a focus on the end product is common in artificial creativity, approaching creativity as a process allows an entry into artificial creativity which has been largely neglected. We focused on one aspect of the creative process, that of insight. A number of striking parallels can be drawn between the psychological creative process and Reinforcement Learning, and Hierarchical Reinforcement Learning in particular.

Reinforcement Learning distinguishes itself by making use of techniques designed from the ground up for embodied, situated agents, rather than for highly abstract domains. Its ontology contains *actions*, *states*, and temporally-extended *policies* rather than concepts; and its representations are directly tied to action. This matches characteristics of humans or animals as situated creators [25,26,47,48], and makes the approach especially relevant and promising for robotic creativity. In RL, by introducing a hierarchy, transfer becomes possible between situations that are analogous at an abstract level. In this sense, our approach is closely related to the research program proposed in [76,86,87]. It is a promising way to achieve the three key challenges on the road to computational creativity identified by Buchanan [16], namely: accumulation of experience, meta-level thinking, and transfer.

The promise of computational creativity in the context of machine learning is that creativity might be able to tackle problems that are currently too big to learn: most real-world problems have a prohibitively large search space and current exploration approaches are ill-equipped to sample non-trivial search spaces, such as those found in even simple robotics problems. Creativity offers a way forward: instead of random exploration or heuristic-based exploration, creativity has the potential to explore areas of the search space at a time when classic approaches would still be exploring in the neighborhood of failed earlier attempts. In addition, the reuse of components (known as "options" in HRL terminology) has the potential to use smaller solutions to tackle bigger problems, and to switch between different exploration approaches, potentially leading to an *Aha Erlebnis* in the search process.

Creativity, mainly through being still ill-defined and elusive, is best studied using an interdisciplinary approach. The cognitive sciences and ethology offer angles on the creative processes that prove useful when considering computational creativity. In this, Hierarchical Reinforcement Learning is a particularly promising approach, as it matches the core characteristics of the insight process. Insight results in a sudden increase in performance while exploring a problem, while other optimization algorithms often show a gradual improvement in performance, atypical for how people arrive at solutions.

Finally, the advantage of a model is that it can be exercised — research into creativity, both computational and psychological, can only benefit from implementable models of the creative process, built on explicit assumptions and producing predictions that can be tested and falsified. Robots are the most natural platform for these tests, as they most closely resemble embodied, unsupervised agents such as animal and human creators.

Acknowledgments

This work was completed as part of Marie Curie Initial Training Network FP7-PEOPLE-2013-ITN, CogNovo, grant number 604764. We would like to thank Paul Baxter for the valuable input to this work.

References

[1] J. Wiley, Expertise as mental set: The effects of domain knowledge in creative problem solving, Mem. Cogn. 26 (4) (1998) 716–730.
[2] S. Ohlsson, Deep Learning: How The Mind Overrides Experience, Cambridge University Press, 2011.
[3] H.F. Harlow, The formation of learning sets, Psychol. Rev. 56 (1) (1949) 51.
[4] R.W. Weisberg, Toward an integrated theory of insight in problem solving, Think. Reason. 21 (1) (2015) 5–39.
[5] G.A. Wiggins, A preliminary framework for description, analysis and comparison of creative systems, Knowl.-Based Syst. 19 (7) (2006) 449–458.
[6] G. Ritchie, A closer look at creativity as search, in: International Conference on Computational Creativity, Dublin, 2012.
[7] M.A. Boden, Computer models of creativity, AI Mag. 30 (3) (2009) 23.
[8] M.A. Boden, The Creative Mind: Myths and Mechanisms, Psychology Press, 2004.
[9] A.M. Turing, Computing machinery and intelligence, Mind 59 (236) (1950) 433–460.
[10] M.A. Boden, The Turing test and artistic creativity, Kybernetes 39 (3) (2010) 409–413.
[11] S. Colton, G.A. Wiggins, Computational creativity: the final frontier? in: ECAI, vol. 12, 2012, pp. 21–26.
[12] A. Pease, S. Colton, The Turing test and computational creativity, TURING (2012) 110.

[13] S. Colton, Creativity versus the perception of creativity in computational systems, in: AAAI Spring Symposium: Creative Intelligent Systems, 2008.

[14] S. Colton, A. Pease, J. Charnley, Computational creativity theory: The FACE and IDEA descriptive models, in: Proceedings of the Second International Conference on Computational Creativity, 2011, pp. 90–95.

[15] A. Pease, S. Colton, Computational creativity theory: Inspirations behind the FACE and the IDEA models, in: Proceedings of the Second International Conference on Computational Creativity, 2011.

[16] B.G. Buchanan, Creativity at the metalevel: AAAI-2000 presidential address, AI Mag. 22 (3) (2001) 13.

[17] J. Bird, D. Stokes, Evolving minimally creative robots, in: Proceedings of the 3rd International Joint Workshop on Computational Creativity, ECAI'06, 2006, pp. 1–5.

[18] P. Bentley, D. Corne, Creative Evolutionary Systems, Morgan Kaufmann, 2002.

[19] R.K. Sawyer, Explaining Creativity: The Science of Human Innovation, Oxford University Press, 2011.

[20] I.K. Ash, P.J. Cushen, J. Wiley, Obstacles in investigating the role of restructuring in insightful problem solving, J. Prob. Solving 2 (2) (2009) 3.

[21] H.A. Simon, The MIT Encyclopedia of The Cognitive Sciences, MIT Press, 2001 Ch. Problem solving.

[22] A. Newell, H.A. Simon, Human Problem Solving, Vol. 104, Prentice-Hall, Englewood Cliffs, NJ, 1972.

[23] K. Koffka, Principles of Gestalt Psychology, Routledge, 1935.

[24] W. Köhler, Intelligenzprüfungen an Menschenaffen [The mentality of apes], Springer-Verlag, Berlin, 1921.

[25] N. Mendes, D. Hanus, J. Call, Raising the level: orangutans use water as a tool, Biol. Lett. 3 (5) (2007) 453–455.

[26] A.H. Taylor, R.D. Gray, Animal cognition: Aesop's fable flies from fiction to fact, Curr. Biol. 19 (17) (2009) R731–R732.

[27] E.L. Thorndike, Animal intelligence: an experimental study of the associative processes in animals, Psychol. Rev. Monograph Supplements 2 (4) (1898) i.

[28] J. Metcalfe, D. Wiebe, Intuition in insight and noninsight problem solving, Mem. Cogn. 15 (3) (1987) 238–246.

[29] R.W. Weisberg, J.W. Alba, An examination of the alleged role of fixation in the solution of several ''insight'' problems, J. Exp. Psychol. [Gen] 110 (2) (1981) 169.

[30] T.C. Kershaw, S. Ohlsson, Training for insight: The case of the nine-dot problem, in: Proceedings of The Twenty-Third Annual Conference of The Cognitive Science Society, 2001, pp. 489–493.

[31] C.A. Kaplan, H.A. Simon, In search of insight, Cogn. Psychol. 22 (3) (1990) 374–419.

[32] G. Knoblich, S. Ohlsson, G.E. Raney, An eye movement study of insight problem solving, Mem. Cogn. 29 (7) (2001) 1000–1009.

[33] J.N. MacGregor, T.C. Ormerod, E.P. Chronicle, Information processing and insight: A process model of performance on the nine-dot and related problems, J. Exp. Psychol. [Learn Mem. Cogn.] 27 (1) (2001) 176.

[34] J.A. Feldman, D.H. Ballard, Connectionist models and their properties, Cogn. Sci. 6 (3) (1982) 205–254.

[35] W. Schultz, Behavioral theories and the neurophysiology of reward, Annu. Rev. Psychol. 57 (2006) 87–115.

[36] J.J. Ribas-Fernandes, A. Solway, C. Diuk, J.T. McGuire, A.G. Barto, Y. Niv, M.M. Botvinick, A neural signature of hierarchical reinforcement learning, Neuron 71 (2) (2011) 370–379.

[37] A. Dietrich, R. Kanso, A review of EEG, ERP, and neuroimaging studies of creativity and insight, Psychol. Bull. 136 (5) (2010) 822.

[38] J. Kounios, M. Beeman, The cognitive neuroscience of insight, Annu. Rev. Psychol. 65 (2014) 71–93.

[39] H.A. Simon, The information processing explanation of gestalt phenomena, Comput. Hum. Behav. 2 (4) (1986) 241–255.

[40] A. Newell, J.C. Shaw, H.A. Simon, Report on a general problem-solving program, in: IFIP Congress, 1959, pp. 256–264.

[41] C.J. MacLellan, An elaboration account of insight, in: AAAI Fall Symposium: Advances in Cognitive Systems, 2011.

[42] C.M. Vigorito, A.G. Barto, Hierarchical representations of behavior for efficient creative search, in: AAAI Spring Symposium: Creative Intelligent Systems, 2008, pp. 135–141.

[43] B.D. Smith, G.E. Garnett, Evolutionary and Biologically Inspired Music, Sound, Art and Design: First International Conference, EvoMUSART 2012, Málaga, Spain, April 11–13, 2012. Proceedings, Springer, Berlin, Heidelberg, Berlin, Heidelberg, 2012, pp. 223–234, Ch. Reinforcement Learning and the Creative, Automated Music Improviser.

[44] L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: A survey, J. Artificial Intelligence Res. (1996) 237–285.

[45] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press Cambridge, 1998.

[46] C. Szepesvári, Algorithms for reinforcement learning, Synthesis lectures on artificial intelligence and machine learning 4 (1) (2010) 1–103.

[47] V.P. Glăveanu, Rewriting the language of creativity: The five A's framework, Review of General Psychology 17 (1) (2013) 69.

[48] T. Ingold, The creativity of undergoing, Pragmat. Cogn. 22 (1) (2014) 124–139.

[49] J.R. Koza, M.A. Keane, M.J. Streeter, T.P. Adams, L.W. Jones, Invention and creativity in automated design by means of genetic programming, (AI EDAM) Artif. Intell. Eng. Des. Anal. Manuf. 18 (03) (2004) 245–269.

[50] J.P. Guilford, The structure of intellect, Psychol. Bull. 53 (4) (1956) 267.

[51] S. Mednick, The associative basis of the creative process, Psychol. Rev. 69 (3) (1962) 220.

[52] D.T. Campbell, Blind variation and selective retentions in creative thought as in other knowledge processes, Psychol. Rev. 67 (6) (1960) 380.

[53] D.K. Simonton, Creative thought as blind-variation and selective-retention: Combinatorial models of exceptional creativity, Phys. Life Rev. 7 (2) (2010) 156–179.

[54] A.G. Barto, Intrinsic motivation and reinforcement learning , in: Intrinsically Motivated Learning in Natural and Artificial Systems, Springer, 2013, pp. 17–47.

[55] S.B. Thrun, Efficient Exploration in Reinforcement Learning, Tech. Rep. CMU-CS-92-102, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PE, 1992.

[56] R.I. Brafman, M. Tennenholtz, R-MAX-a general polynomial time algorithm for near-optimal reinforcement learning, J. Mach. Learn. Res. 3 (2003) 213–231.

[57] T. Hester, M. Quinlan, P. Stone, Generalized model learning for reinforcement learning on a humanoid robot, in: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 2369–2374.

[58] S. Singh, R.L. Lewis, A.G. Barto, J. Sorg, Intrinsically motivated reinforcement learning: An evolutionary perspective, IEEE Trans. Auton. Mental Dev. 2 (2) (2010) 70–82.

[59] P.-Y. Oudeyer, F. Kaplan, How can we define intrinsic motivation? in: Proceedings of The 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 2008.

[60] X. Huang, J. Weng, Novelty and reinforcement learning in the value system of developmental robots, in: Lund University Cognitive Studies, 2002.

[61] J. Schmidhuber, Curious model-building control systems, in: Neural Networks, 1991. 1991 IEEE International Joint Conference on, IEEE, 1991, pp. 1458–1463.

[62] P.-Y. Oudeyer, F. Kaplan, V.V. Hafner, Intrinsic motivation systems for autonomous mental development, IEEE Trans. Evol. Comput. 11 (2) (2007) 265–286.

[63] V.G. Santucci, G. Baldassarre, M. Mirolli, Intrinsic motivation mechanisms for competence acquisition, in: Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE international conference on, IEEE, 2012, pp. 1–6.

[64] Y. Sun, F. Gomez, J. Schmidhuber, Planning to be surprised: Optimal bayesian exploration in dynamic environments, in: Artificial General Intelligence, Springer, 2011, pp. 41–51.

[65] N.K. Jong, Structured exploration for reinforcement learning, (Ph.D. thesis), 2010 URL http://www.cs.utexas.edu/users/ai-lab/?nkj-thesis.

[66] A.G. Barto, G. Konidaris, C. Vigorito, Behavioral hierarchy: exploration and representation, in: Computational and Robotic Models of the Hierarchical Organization of Behavior, Springer, 2013, pp. 13–46.

[67] R. Parr, S. Russell, Reinforcement learning with hierarchies of machines, Adv. Neural Inf. Process. Syst. (1998) 1043–1049.

[68] T.G. Dietterich, Hierarchical reinforcement learning with the MAXQ value function decomposition, J. Artificial Intelligence Res. 13 (2000) 227–303.

[69] A.G. Barto, S. Mahadevan, Recent advances in hierarchical reinforcement learning, Discrete Event Dyn. Syst. 13 (1–2) (2003) 41–77.

[70] R.S. Sutton, D. Precup, S. Singh, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, Artificial intelligence 112 (1) (1999) 181–211.

[71] T.G. Dietterich, State abstraction in MAXQ hierarchical reinforcement learning, in: Adv. Neural Inf. Process. Syst., 2000, pp. 994–1000.

[72] A. Jonsson, A.G. Barto, Automated state abstraction for options using the U-TREE algorithm, Advances in neural information processing systems (2001) 1054–1060.

[73] N.K. Jong, P. Stone, State abstraction discovery from irrelevant state variables, in: IJCAI, Citeseer, 2005, pp. 752–757.

[74] T.J. Perkins, D. Precup, et al., Using Options for Knowledge Transfer in Reinforcement Learning, University of Massachusetts, Amherst, MA, USA, Tech. Rep.

[75] M.E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey, J. Mach. Learn. Res. 10 (2009) 1633–1685.

[76] T.R. Besold, K.-U. Kühnberger, Towards integrated neural–symbolic systems for human-level AI: two research programs helping to bridge the gaps, Biol. Inspired Cogn. Architectures 14 (2015) 97–110.

[77] G. Konidaris, S. Kuindersma, R.A. Grupen, A.G. Barto, Autonomous skill acquisition on a mobile manipulator, in: AAAI, 2011.

[78] S. Thrun, A. Schwartz, et al., Finding structure in reinforcement learning, Adv. Neural Inf. Process. Syst. (1995) 385–392.

[79] E. Brunskill, L. Li, PAC-inspired option discovery in lifelong reinforcement learning, in: Proceedings of the 31st International Conference on Machine Learning, ICML-14, 2014, pp. 316–324.

[80] R.S. Sutton, J. Modayil, M. Delp, T. Degris, P.M. Pilarski, A. White, D. Precup, Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction, in: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 761–768.
[81] P. Dayan, G.E. Hinton, Feudal reinforcement learning, in: Advances in Neural Information Processing Systems, Morgan Kaufmann Publishers, 1993 271–271.
[82] D. Silver, R.S. Sutton, M. Müller, Sample-based learning and search with permanent and transient memories, in: Proceedings of The 25th International Conference on Machine Learning, ACM, 2008, pp. 968–975.
[83] R.S. Sutton, A. Koop, D. Silver, On the role of tracking in stationary environments, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 871–878.
[84] M. Stolle, D. Precup, Learning options in reinforcement learning, in: SARA, Springer, 2002, pp. 212–223.
[85] P.-L. Bacon, D. Precup, The option-critic architecture, in: NIPS Deep Reinforcement Learning Workshop, 2015.
[86] A. Chella, S. Gaglio, R. Pirrone, Conceptual representations of actions for autonomous robots, Robot. Auton. Syst. 34 (4) (2001) 251–263.
[87] A.d. Garcez, T.R. Besold, L. de Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L.C. Lamb, R. Miikkulainen, D.L. Silver, Neural-symbolic learning and reasoning: contributions and challenges, in: Proceedings of the AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches, Stanford, 2015.

Thomas R. Colin received a M.Sc. degree in software engineering from EFREI, Villejuif, France, in 2008, and the M.Sc. degree (cum laude) in Cognitive Artificial Intelligence from Utrecht University, Utrecht, the Netherlands, in 2013.

He is currently a Marie Curie Fellow and Ph.D. student within the CogNovo program at Plymouth University, Plymouth , U.K. He focuses on developing an interdisciplinary understanding of creativity through the implementation of creative control in robots.

Tony Belpaeme is Professor of Cognitive Systems and Robotics at Plymouth University. He is associated with the Cognition Institute and the Centre for Robotics and Neural Systems. His research interests include human–robot interaction, social systems, and artificial intelligence in general. Belpaeme's lab tries to further the science and technology behind artificial intelligence of social robots, taking inspiration from human cognition to build useful machines.

Angelo Cangelosi received the Ph.D. degree in psychology and computational modeling from the University of Genoa, Genoa, Italy, in 1997, while also working as a visiting scholar at the National Research Council, Rome, the University of California San Diego, La Jolla, CA, and the University of Southampton, Southampton, U.K.

He is currently a Professor of Artificial Intelligence and Cognition at the University of Plymouth, Plymouth, U.K., where he leads the Centre for Robotics and Neural Systems. He has produced more than 200 scientific publications and has been awarded numerous research grants from the United Kingdom and international funding agencies.

Nikolas Hemion received a Ph.D. (Dr.-Ing.) degree in 2013 from Bielefeld University, Bielefeld, Germany.

He is currently a researcher in the AI-Lab at Aldebaran Robotics, Paris, France. His research interests focus on cognitive architecture in developmental robotics, self-organized learning of sensorimotor representations, and emergence of interaction.