# Biologically and Cognitively Inspired General Game Playing: Self-Motivation in Sokoban, Gambler's Problem and Pacman

Tom Anthony, Daniel Polani, Chrystopher L. Nehaniv

***Abstract*—We use *empowerment*, a recently introduced biologically inspired measure, to allow an AI player to assign utility values to states within a previously un-encountered game where it has no knowledge of any existing goal states. We demonstrate how an extension to this concept, *open-ended empowerment*, allows the player to group together candidate action sequences into a number of 'strategies'. This grouping is determined by their *strategic affinity*, and these groupings are used to select a repertoire of action sequences that aim to maintain anticipated utility.**

**We show how this method provides a *proto-heuristic* for non-terminal states in advance of specifying the concrete game goals, and propose it as a principled candidate model for "intuitive" strategy selection, in line with other recent work on generation of "self-motivated agent behaviour". We in particular demonstrate that the technique, despite being generically defined independently of scenario, performs quite well in relatively disparate scenarios, such as a Sokoban-inspired box-pushing scenario, in Dubin and Savage's *Gambler's Problem*, and in a simplified Pacman game, and it suggests novel and principle-based candidate routes towards more general game-playing algorithms.**

***Index Terms*—Artificial intelligence (AI), information theory, Games**

## I. Introduction

### A. Motivation

*"Act always so as to increase the number of choices."*
- Heinz von Foerster

In many games, including those which are currently largely inaccessible to computer techniques, there exists for many states of the game a subset of the possible actions which are considered 'preferable'. In some games it is easy to identify these actions, but for many more complex games identifying them can be extremely difficult. While in games such as Chess algorithmic descriptions of the quality of a situation have led to powerful computer strategies, whilst capturing the intuitive concept of the *beauty* of a position, which often is believed to guide human master players, remains elusive (1). One is unable to provide precise rules for a beauty heuristic, which, furthermore, would identify advantageous moves. This intuition tallies with the ability of master Chess players to appreciate the structural aspects of a position in the game, and from this identify important states and moves.

T.Anthony, D. Polani and C.L.Nehaniv are with the Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire, Hatfield, Herts, AL10 9AB, U.K. e-mail: ({T.C.Anthony,D.Polani,C.L.Nehaniv}@herts.ac.uk).

Whilst there exist exceedingly successful algorithmic solutions for some games, much of the success derives from a combination of computing power with human explicitly-designed heuristics. In other words, whatever structural knowledge in a game Artificial Intelligence (AI) has been put in by its human designers. In this unsatisfactory situation, the core challenge for AI remains: can we produce algorithms able to identify these or other structural patterns in a more general way which would apply to a broader collection of games and puzzles? In addition, can we create an AI player motivated to identify these structures itself?

Before and since the Dartmouth Conference, game tree search algorithms were proposed to identify good actions or moves for a given state (2; 3; 4). However, it has since been largely felt that tree search algorithms, with all their practical successes, make a limited contribution in moving us towards 'intelligence' mechanism that could be interpreted as plausible from point of view of human-like cognition; by using brute-force computation the algorithms sidestep the necessity of identifying how 'beauty' and related structural cues would be detected (or constructed) by an artificial agent. On Gary Kasparov's first game loss to Deep Blue in 1996, Douglas Hofstadter noted "It was a watershed event, but it doesn't have to do with computers becoming intelligent", adding "you can bypass deep thinking in playing chess, the way you can fly without flapping your wings" (5). Previously, John McCarthy had gone so far as to call this problem of AI a 'scandalous weakness', predicting it could be overcome by brute-force for Chess but not yet Go [1] and criticising that with Chess the solutions were simply 'substituting large amounts of computation for understanding' (6). Recently, games such as Arimaa were created purely to challenge these shortcomings (7) and provoke research to find alternative methods. Arimaa is a game played with Chess pieces, on a Chess board, and with simple rules, but normally a player with only a few of games experience can beat the best computer AIs. Tree-search approaches are unlikely to make any headway, as Arimaa's average branching factor exceeds 17,000 (Chess's is $\sim$40, Go's is $\sim$200).

At a conceptual level, using various pruning methods and other shortcuts to optimize the process, tree search algorithms generally rely on the concept of performing an exhaustive search to a certain depth. With these optimizations the search

[1] of course, recent progress in Go-playing AI may render McCarthy's pessimistic prediction concerning performance (as so often in the history of computation) moot, but, even so, the qualitative criticism stands.

will not, in reality, be an exhaustive search, the approach is unlikely to be mimicking a human approach. Furthermore, at leaf nodes of such a search the state is usually evaluated with hand-crafted heuristics, created the AI designer for the specific game or problem.

These approaches do not indicate how higher-level concepts might be extracted from simple rules of the game, or how structured strategies might be identified by a human. For example, a given Chess position a human might consider two strategies at a given moment (e.g 'attack opponent queen' or 'defend my king') before considering which moves in particular to use to enact the chosen strategy. Tree search approaches do not operate on a level which either presupposes or provides conceptual game structures (the human-made AI heuristics may, of course, incorporate them, but this is then an explicit proviso by the human AI designer).

More recently, Monte Carlo Tree Search (MCTS) algorithms (8) have been developed which overcome a number of the limitations of the more traditional tree search approaches. MCTS primarily works to select a move from a given position by trying each possible next move and then completing the rest of the game to a terminal state by making random moves for all players until such a state is reached. Many iterations of this are repeated for each subsequent possible action from the game tree node being evaluated, and statistics are kept for how often the game is won (by examination of the terminal node) when starting with each of these first moves. The most common variant, known as Upper Confidence bounds applied to Trees (UCT) (9), biases the sampling towards repeated simulations of those nodes currently showing a higher win percentage; this is used to offset the problem that these moves may only appear to be good as long as one small set of particular opponent responses avoids evaluation.

MCTS algorithms represent an important breakthrough in themselves, and lead us to a better understanding of tree searching. However, whilst MCTS has significantly extended the potential ability of tree search algorithms, it remains limited by similar conceptual constraints as previous tree search methods. We suggest that an improved understanding of how to deconstruct problems into smaller local problems which may be addressed individually, with either tree search or an alternative approach, is more cognitively plausible and would prove decisive in overcoming many of the conceptual problems associated with tree search algorithms. However, so far generic techniques for such a deconstruction, especially using a sufficiently general approach that requires little domain-specific knowledge, have proved difficult to find.

To move towards this goal, in this paper we propose a model stemming from cognitive and biological considerations. For this purpose we adopt the perspective of intelligence arising from situatedness and embodiment (10) and view the AI player as an agent that is 'embodied' within an environment (11; 12). The agent's actuator options correspond to the legal moves within the game, and its sensors reflect the state of the game (those parts available to that player according to the relevant rules).

Furthermore, we create an incentive towards structured decisions by imposing a cost on the search/decision process; this is closely related to the concept of *bounded rationality* (13; 14) which deals with decision making when working with limited information, cognitive capacity, and time and is used as a model of human decision-making in economics (15). As natural cost functionals for decision processes, we use information-theoretical quantities; there is a significant body of evidence that such quantities have not only a prominent role in learning theory (16; 17), but also that various aspects of biological cognition can be successfully described and understood by assuming informational processing costs being imposed on organisms (18; 19; 20; 21; 22; 23; 24).

Thus, our present adoption of an information-theoretic framework in the context of decisions in games is plausible not only from a learning- and decision-theoretic point of view, but also from the perspective of a biologically oriented high-level view of cognition where the pay-off conferred by any decision has to be traded off with the informational effort of achieving it.

Here, more specifically, we combine this "thinking in informational constraints" with *empowerment* (25; 26), another information-theoretic concept generalising the notion of 'mobility' (27) or 'options' available to an agent in its environment. Empowerment provides on the one hand an informational generalization of mobility in the usual game-playing sense, and in addition it makes available the complete arsenal of information-theoretic methodologies. This allows one to treat stochastic systems, systems with incomplete information, dynamical systems, games of complete information and other systems in essentially the same coherent way (28). To put the present paper in context and for reasons of self-containedness, we will, in section I-C, briefly review earlier work.

In game terms, this above technique could be thought of as a type of 'proto-heuristic' that transcends specific game dynamics and works as a default strategy to be applied, before the game-specific mechanics is refined. This could prove useful either independently or as heuristics from genesis which could be used to guide an AI players behaviour in a new game whilst game-specific heuristics were developed during play. In the present paper we do not go as far as exploring the idea of building game-specific heuristics on top of the proto-heuristics, but focus on deploying the method to generate useful behaviour primitives. We demonstrate the operation of proto-heuristics in three game scenarios and show that intuitively 'sensible' behaviours are selected.

### B. Information Theory

To develop the method, we require Shannon's theory of information for which we give a very basic introduction. To begin we introduce *entropy*, which is a measure of uncertainty; the entropy of a variable $A$ is defined as:

$$H(A) = -\sum_{a \in A} p(a) \log p(a). \tag{1}$$

where $p(a)$ is the probability that $A$ is in the state $a$. The logarithm can be taken to any chosen base; in our paper we always use 2, and the entropy is thus measured in *bits*. If

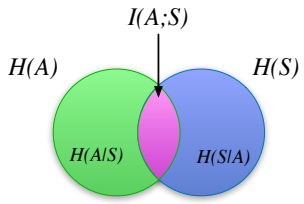


Fig. 1. Visualization of the primary information theory quantities and their relationships. $H(A)$ is the entropy of the random variable $A$, $H(A|S)$ is the conditional entropy of the same variable conditioned on knowing the value of $S$. Finally $I(A;S)$ is the mutual information between these two variables, which measures the amount of information that the two variables reveal about one another.

$S$ is another random variable jointly distributed with $A$, the *conditional entropy* is:

$$H(S|A) = -\sum_{a\in A} p(a) \sum_{s\in S} p(s|a) \log p(s|a). \tag{2}$$

This measures the remaining uncertainty about the value of $S$, if we know the value of $A$; its relationship to entropy is visualised in Fig 1. Finally, this also allows us to measure the *mutual information* between two random variables:

$$\begin{aligned} I(A;S) &= H(S) - H(S|A) \\ &= \sum_{a\in A}\sum_{s\in S} p(a,s) \log\left(\frac{p(a,s)}{p(a)\,p(s)}\right) \end{aligned} \tag{3}$$

Mutual information can be thought of as the reduction in uncertainty about the variable $A$ or $S$, given that we know the value of the other; or otherwise, a measure of how much knowing one variable informs us about the other. The mutual information is symmetric, so we could also use $I(A;S) = H(A) - H(A|S)$ (29). In this paper we will later also examine the mutual information between a particular value of a random variable with another random variable:

$$I(a;S) = p(a) \sum_{s\in S} p(s|a) \log\left(\frac{p(a,s)}{p(a)\,p(s)}\right). \tag{4}$$

Finally, we introduce the information-theoretic concept of the *channel capacity* (30; 29) which we will use later. It is defined as:

$$C(p(s|a)) = \max_{p(a)} I(A;S). \tag{5}$$

### C. Empowerment

Empowerment, based on the information-theoretic perception-action loop formalism introduced in (26, 31), is a quantity characterizing the "sensorimotor adaptedness" of an agent in its environment. It quantifies the ability of situated agents to influence their environments.

For the purposes of puzzle-solving and game-play, we translate the setting as follows: consider the player carrying out a move as a sender sending a message, and observing the subsequent state on the board as receiving the response to this message. In terms of Shannon information, when the agent performs an action, it 'injects' information into the environment, and subsequently the agent re-acquires part of this information from the environment via its sensors. Note that in the present paper, we discuss only puzzles and games with perfect information, but the formalism carries over directly to the case of imperfect information game.

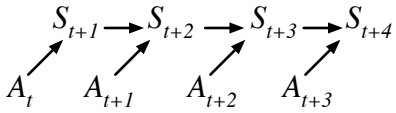


Fig. 2. Bayesian network representation of the perception-action loop.

For the scenarios relevant in this paper, we will employ a slightly simplified version of the empowerment formalism. The player (agent) is represented by a Bayesian network, shown in fig. 2, with the random variable $S_t$ the state of the game (as per the player's sensors), and $A_t$ a random variable denoting the action at time $t$.

As mentioned above, we consider the communication channel formed by the action, state pair $A_t, S_{t+1}$ and compute the channel capacity, i.e. the maximum possible Shannon information that one action $A_t$ can 'inject' or store into the subsequent state $S_{t+1}$. We define empowerment as this 'motori-sensor' channel capacity:

$$\mathfrak{E} = C(p(s|a)) = \max_{p(a)} I(A;S). \tag{6}$$

If we consider the game as homogenous in time, we can ignore the time index, and empowerment only depends on the actual state $s_t$. We will imply this starting state as given throughout the rest of the text.

Note that the channel capacity is measured as the maximum mutual information taken over all possible input distributions, $p(a)$, and depends only on $p(s|a)$, which is fixed for a given starting state $s_t$. This corresponds to potential maximum amount of information about its prior actions an agent can later observe. One algorithm that can be used to find this maximum is the iterative Blahut-Arimoto algorithm (32).

Empowerment measures the *potential* change in subsequent states that the player can cause by its current actions - it is a generalized measure of mobility; generalized, because it can smoothly incorporate randomness or incomplete information. If noise causes actions to produce less controllable results, this is detected via a lower empowerment value.

Instead of a single action, it makes often sense to consider an action sequence of length$n > 1$ and its effect on the state. In this case, which we will use throughout most of the paper, we will speak about $n$-step empowerment. Formally, we first construct a compound random variable of the last $n$ actuations $(A_t, A_{t-1}, A_{t-2}, \ldots, A_{t-n+1}) = A_t^n$. We now maximize the mutual information between this variable and the state at time $t+n$, represented by $S_{t+n}$. $n$-step empowerment is the channel capacity between these:

$$\mathfrak{E} = C(p(s_{t+n}|a_t^n)) = \max_{p(a_t^n)} I(A_t^n; S_{t+n}). \tag{7}$$

Fig. 3. An example showing scaled empowerment values for $n = 5$ in a simple maze example. A darker shade indicates higher empowerment, which correlates well with the average shortest path from each cell to all others. Empowerment values in this example range from 2.58 to 4.64 bits.

It should be noted that $\mathfrak{E}$ depends on $S_t$ (the current state of the world) but to keep the notation unburdened, we will always assume conditioning on the current state $S_t$ implicitly and not explicitly write it.

Although we will mostly present entirely deterministic scenarios, to demonstrate that empowerment is defined in full generality for non-deterministic probabilistic environments we will present a few stochastic scenarios.

In the present paper we will present two extensions to the empowerment formalism which are of particular relevance for puzzles and games. The first, discussed in section III, is impoverished empowerment; it sets constraints on the number of action sequences an agent can retain, and was originally introduced in (33). The second, presented in section V, introduces the concept of a soft horizon for empowerment which allows an agent to use a 'hazy' prediction of the future to inform action selection. Combined these present a model of resource limitation on the actions that can be retained in memory by the player and corresponds to formulating a 'bounded rationality' constraint on empowerment fully inside the framework of information theory.

*1) Review of Previous Results:* For purposes of self-containedness and context-setting, it is important to review some earlier results.

*a) Maze:* This scenario consists of a typical maze scenario in a 2-D gridworld, through which the agent can move horizontally or vertically with its movement constrained by walls. With one north, east, south, west movement per time step, empowerment was measured for each cell in the grid-world for *empowerment horizons* given by $n$ (26). The results showed that by only considering a single time step ($n = 1$), the empowerment values are still quite crude, but as the horizon is extended it can be seen that empowerment is increasingly able to differentiate the preferable (in terms of mobility) states from the less preferable ones.

The important observation is that the empowerment values for a cell correlate well with the length of the average shortest path from this cell to all others; more generally, it correlates strongly with the graph-theoretic measure of closeness centrality (34). This is of interest, since the centrality measure has been designed specifically for graphs, while empowerment can be used in a more general context of (possibly stochastic) actions of generic agents in an environment.

*b) Box Pushing:* We consider an agent in a grid world with the same action set, but with the walls removed. Instead, a box is introduced which the agent is able to manipulate by pushing: when the agent moves from a square adjacent to the box into the cell occupied by the box, it causes the box to move in the same direction by one cell.

Here, empowerment indicated a preference for the agent to be closer to the box. Far from the box, the actions of the agent caused only the latter to move, but close to the box, the agent's actions could additionally manipulate the box' position, leading to a significant increase in empowerment. The bottom line of this experiment is that empowerment identifies manipulable objects, or, more generally, manipulable degrees of freedom.

*c) Pole Balancing:* As another class of scenarios, empowerment was considered in the pole-balancing task often encountered in control theory (28). The scenario describes a pole balancing upright upon a cart, which moves on an even surface. The agent in this scenario has two actions: applying an always equal force to either move the cart forwards or backwards. Traditionally, the aim is to move the cart forwards and backwards as necessary to keep the pole balanced, starting from some initial angle. Should the pole move too far in one direction, then it will exceed a threshold making it impossible to recover and will fall flat.

In the empowerment-based approach, no external reward was used; instead, empowerment, which derives directly from the intrinsic system dynamics, was utilized to guide the system's behaviour. Empowerment identified the pole being perfectly upright as amongst the most empowered states, and at progressively steeper angles, empowerment drops, until it falls to 0 when the pole becomes utterly uncontrollable.

Using an action selector that greedily maximizes predicted empowerment in the following time step, leads to pole-balancing. This observation generalizes to other, more complex balancing scenarios (35).

## II. General Game Playing

In summary, the scenarios reviewed above indicate that empowerment is able to provide a default utility which 1. derives only from the structure of the problem itself and not from an external reward 2. identifies the desirability of states in way that matches intuition and 3. carries over between scenarios of apparently different character.

This makes it a promising candidate to assign a proto-utility to states of a given system, even before a utility (and a goal) have been explicitly specified. Essentially it does this by measuring an agent's potential to controllably reach as many different states as possible from the current one.

Empowerment is more than a naive mobility measure; in calculating empowerment for a given state, it incorporates the structure and dynamics of the agent's world and embodiment. We note that a duality exists between states and actions in the transition graph of any environment, and empowerment is one technique which exploits this. Now, in order to move towards a method that could be used for General Game Playing, there are three primary issues we must first address:

1) There are reasons to suspect that the ability of biological cognition to structure its decision-making process is driven by the necessity to economize its information processing (36; 37). In other words, we postulate that suitable bounded rationality assumptions are necessary to generate structured behaviour. We will represent these assumptions entirely in terms of the language of our information-theoretic framework, in terms of limited 'informational bandwidth' of actions. For games this cognitive cost to processing the environment is especially true where we desire an AI player to play in real-time or at least as fast as a human player.
2) For $n$-step empowerment to be effective in most scenarios, including games, the reliance on a strict horizon depth is problematic and needs to be addressed.
3) The action policy generated by empowerment should identify that different states have different utilities. Naive mobility-like empowerment does not account for the fact that being able to reach some states can be more advantageous that being able to reach others.

In sections III we will address the first issue. As for issue 2 and 3, it turns out that they are very related to one another; they will be discussed further in sections IV and V.

Finally, in section VI we will bring together all considerations and apply it to a selection of game scenarios.

## III. Impoverished empowerment

In the spirit of bounded rationality outlined above, we modified the $n$-step empowerment algorithm to introduce a constraint on the bandwidth of action sequences that an agent could retain. We call this modified concept 'impoverished empowerment' (33). This allows us to identify possible favourable trade-offs, where a large reduction in the bandwidth of action sequences has little impact of empowerment.

While in the original empowerment definition, all possible action sequences leading to various states are considered, in impoverished empowerment, one considers only a strongly restricted set of action sequences. Therefore, we need to identify action sequences which are most empowering, i.e those that contribute most to the agent's empowerment; how one action sequence can be more empowering than another is a function of the action sequence's stochasticity (does it usually get where it wanted to go), and whether other action sequences lead to the same state (are there other ways to do this).

### A. Scenario

To investigate the impoverished empowerment concept we revisited the scenario from (26); a player is situated within a 2-dimensional infinite gridworld and can select one of 4 actions in any single time step. The actions the agent can execute are moving North, South, East, and West; each moving the agent by one space into the corresponding cell, provided it is not occupied by a wall. In the scenario the state of the world is completely determined by the position of the agent.

### B. Impoverished Empowerment Algorithm

This bandwidth reduction works by clustering the available action sequences together into a selected number of groups, from which a single representative action sequence is then selected. The selected action sequences are then form a reduced set of action sequences, for which we can calculate the empowerment.

*Stage 1*

Compute the empowerment in the conventional way, obtaining a empowerment-maximizing probability distribution $p(a_t^n)$ for all $n$-step action sequences $a$ (typically with $n < 6$).

Having calculated the empowerment we have two distributions: $p(a_t^n)$ is the capacity achieving distribution of action sequences and $p(s_{t+n}|a_t^n)$ is the channel that represents the results of an agent's interactions with the environment. Here, for conciseness, we will write $A$ to represent action *sequences*, not only single actions.

*Stage 2*

In traditional empowerment computation, $p(a_t^n)$ is retained for all $n$-step sequences $a$. Here, however, we assumed a bandwidth limitation on how many such action sequences can be retained. Instead of 'remembering' $p(a_t^n)$ for all action sequences $a$, we *impoverish* $p(a_t^n)$. i.e. we are going to 'thin down' the action sequences to the desired bandwidth limit.

To stay entirely in the information-theoretic framework, we employ the so-called *information bottleneck* method (38; 39). Here, one assumes that the probability $p(s_{t+n}|a_t^n)$ is given, meaning you need a model of what will be possible outcomes for a given action by a player in a given state. In single player games this is easily determined, whereas in multiplayer games we need a model of the other players (we discuss this more in section VII-A).

We start by setting our designed bandwidth limit by selecting a cardinality for a variable $G$ where $|G| \leq |A_t^n|$; we now wish to find a distribution $p(g|a_t^n)$, where $g$ is a group of action sequences with $g \in G$.

The information bottleneck algorithm can be used to produce this mapping, using the original channel as an input. It acts to maximize $I(S_{t+n}; G)$ whilst keeping $I(G; A_t^n)$ constant; it can be thought of 'squeezing' the information $A_t^n$ shares with $S_{t+n}$ though the new variable $G$ to maximize the information $G$ shares with $S_{t+n}$ whilst discarding the irrelevant aspects. By setting a cardinality for $G$ and then running the information bottleneck algorithm we obtain a conditional distribution $p(g|a_t^n)$, which acts as a mapping of actions to groups.

The result of this is action sequences that usually lead to identical states are clustered together to into groups. However, if the number of groups is less than the number of observed states then beyond the identical state action sequences, the grouping is arbitrary, as seen in fig. 4. This is because there is nothing to imply any relation between states, be it spatial or otherwise - states are only consistently grouped with others that lead to the same state.

Contrary to what could be expected, the introduction of some noise in the environment actually improves this situation, and improves the clustering of actions to those that are more 'similar' (in this case spatially); this comes about because the

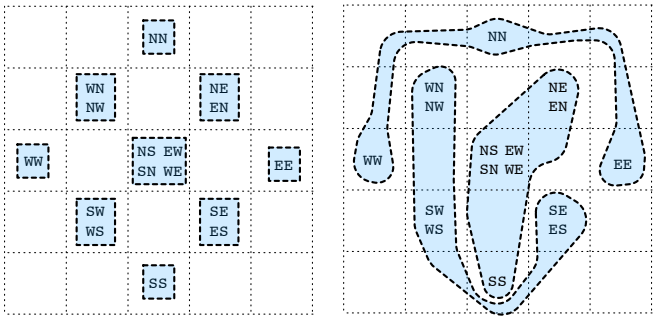


Fig. 4. Visualization of action sequence grouping using Impoverished Empowerment in an empty gridworld with 2-steps. Lighter lines represent the grid cells, darker lines the groupings. It can be seen that should the bandwidth restriction fit neatly to the number of reachable states, as on the left, that those action sequences that lead to the same state are neatly grouped together. However, if the bandwidth is restricted further, then the grouping starts to become somewhat arbitrary (with action sequences that lead to identical states still grouped). This shortcoming is one area that 'open-ended' empowerment addresses.

ability to get 'blown of course' in a scenario, meaning you sometimes end up not in the expected outcome state but in a nearby one results in a slight overlap of outcome states between similar action sequences. However, it is clear that relying on noise for such a result is less than ideal and a better solution to this problem is introduced in section V.

*Stage 3*

Because our aim is to select a subset of our original action sequences to form the new action policy for the agent, we must use an algorithm to 'decompose' this conditional distribution $p(g|a_t^n)$ into a new distribution of action sequences, which has an entropy within the specified bandwidth limit.

We wish to maximize empowerment, so for each $g$ we select the action sequence which provides the most towards our empowerment (i.e the highest value of $I(a_t^n; S_{t+n}|g)$). However, when selecting a representative action sequence for a given $g$ we must consider $p(g|a_t^n)$ (i.e does this action sequence truly represent this group) so we weight on that; however in most cases the mapping between $g$ and $a_t^n$ is a hard partitioning so this is not normally important.

This results in collapsing groups to their 'dominant' action sequence.

### C. Impoverishment Results

Fig. 5 shows three typical outcomes of this algorithm; in this example we have a bandwidth constraint of 2 bits corresponding to 4 action sequences, operating on sequences with a length of 6 actions; this is a reduction of $4^6 = 4096$ action sequences down to 4. The walls are represented by black, the starting position of the agent is the green center square, and the selected trajectories by the thin arrowed lines with a pink cell marking the end location of the sequence. The result that emerges consistently for different starting states is a set of 'skeleton' action sequences set extending into the state space around the agent. In particular, note that stepping through the doorway which intuitively constitutes an environmental feature of particular salient interest is very often found amongst the 4 action sequences.

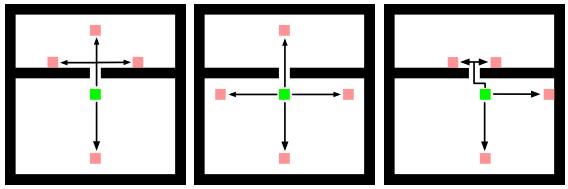

Fig. 5. Typical behaviours where 4 action sequences were selected from $4^6$ possibilities. The selected actions represent a 'skeleton' of the possible action sequence available to an agent. We note that doors represent some sort of transition point and usually some actions that travel through doorways are selected as preferable. The green cell represents the initial cell of the agent and the pink cells represent the final location of the agent in case.

Inspection reveals that a characteristic feature of the sequences surviving the impoverishment is the end points of each sequence usually each have a single unique sequence (of the available $4^6$) that reaches them.

This can be understood by the following considerations: In order to maintain empowerment whilst reducing bandwidth, the most effective way is to eliminate redundantly reachable states first since these 'waste' action bandwidth without providing a richer set of end states. Uniquely reachable states provide 'more state for less action bandwidth' and are therefore advantageous to retain during impoverishment; in Fig. 5, the last action sequences the agent will retain are those leading to states that have only a single unique sequence that reaches them. This is a consequence of selecting the action from each group by $I(a_t^n; S_{t+n})$, and may or may not be desirable; however, in the open-ended empowerment model to follow we will see this result disappears.

## IV. The Horizon

Identifying the correct value for $n$, for $n$-step empowerment (i.e. the empowerment horizon depth) is critical for being able to make good use of empowerment in an unknown scenario. A value of $n$ which is too small can mean that states are not correctly differentiated from one another as some degrees of freedom lie beyond the agents horizon. By contrast a value of $n$ which is too large (given the size of the world) can allow the agent to believe all states are equally empowered (33).

Furthermore, it is unlikely that a static value of $n$ would be suitable in many non-trivial scenarios (where different parts of the scenario require different search horizons), and having a 'hard' horizon compounds this.

### A. Softening the horizon

We understand intuitively that, when planning ahead in a game, a human player does not employ a hard horizon, but instead probably examines some moves ahead precisely, and beyond that has a somewhat hazy prediction of likely outcomes.

In the open-ended empowerment model we will start from this premise and use a similar 'softening' of the horizon, and demonstrate how not only does it help to address the problem of the hard horizon, but also helps identify relationships between action sequences which allows the previously presented clustering process to operate more effectively. It allows us to

group sets of action sequences together into 'alike' sequences (determined by the overlap in their potential future states), with the resulting groups of action sequences representing 'strategies' . This will be shown later to be useful for making complex puzzles easier to manage for agents. Furthermore, we will show that this horizon softening can help to estimate any ongoing utility we may have in future states, having followed an action sequence. We acknowledge that some states in our observable future may be more 'empowered' than others (e.g they do not lead down a 'dead-end' and thus provide some ongoing empowerment), and in the following section we add that understanding to our formalism.

## V. 'Open-ended' Empowerment

Open-ended empowerment is an extension of the impoverished empowerment model and provides two profound improvements: the clustering of action sequences into groups is enhanced such that the clusters formed represent *strategies*, and it allows an agent to roughly forecast future empowerment following an action sequence. We will show that these features also suggest a solution to having to pre-determine the appropriate horizon value, $n$, for a given scenario.

### A. Split the horizon

As before, we designate a set of actions $A$, which an agent can select from in any given time step. For convenience we label the number of possible actions in any time step, $|A|$, as $c$.

We form all possible action sequences of length $n$, representing all possible action 'trajectories' a player could take in $n$ time steps, such that we have $c^n$ trajectories.

From here, we can imagine a set of $c^n$ possible states the agent arrived in corresponding to the trajectory the agent took, $S_{t+n}$. It is likely that there are less than $c^n$ unique states, because some trajectories are likely commutative and the game world is Markovian and also because the world is possibly stochastic, but for now we proceed on the assumption that $c^n$ trajectories leads to $c^n$ states (we show how to optimize this in section V-C).

Next, we consider from each of these $c^n$ states what the player could do in an additional $m$ time steps, using the same action set as previously.

We now have for every original trajectory $c^n$, a set of $c^m$ possible ongoing trajectories. From the combination of these trajectories we can create a the set of states that represents the outcomes of all trajectories of $n+m$ steps, and label this $S_{t+n+m}$.

We can form a channel from these states and actions; traditional empowerment's channel would be $p(s_{t+n}|a_t^n)$, corresponding colloquially to 'what is the probability of ending up in a certain state given the player performed a certain action'. With the two trajectories we could form the channel $p(s_{t+n+m}|a_t^{n+m})$, which would be equivalent to if we had simply increased $n$ by $m$ additional steps.

Instead, we create a channel $p(s_{t+n+m}|a_t^n)$, corresponding colloquially to 'what is the probability of ending up in a certain state in $n+m$ steps time if the player performs a given action sequence in the first $n$ steps'. Essentially we are forecasting the potential ongoing future that would follow from starting with a given $n$-step action sequence.

To do this we need to aggregate and normalise the various distributions of $S_{t+n+m}$ for those which stem from the same original $n$-step action sequence, $a_t^n$ (their common 'ancestor sequence'). We can calculate this channel:

$$p(s_{t+n+m}|a_t^n) = \frac{\sum_{A_{t+n}^m} p(s_{t+n+m}|a_t^n, a_{t+n}^m)}{\left|A_{t+n}^m\right|} \tag{8}$$

where

$$p(s_{t+n+m}|a_t^{n+m}) \equiv p(s_{t+n+m}|a_t^n, a_{t+n}^m) \tag{9}$$

The result of this 'folding back' to ancestor sequences is that the channel now incorporates two important aspects of the initial $n$-step sequences:

1) each value for $a_t^n$ now has a rough forecast of its future which can be used to approximate a 'future empowerment' value, i.e what is a players empowerment likely to be after completing the given n-step action sequence, $a_t^n$.
2) the distribution of potential future states, $S_{t+n+m}|a_t^n$, for different values of $a_t^n$ can be used to compare the potential overlap (the similarity, if you will) in the possible futures that follow from those values of $a_t^n$. This corresponds to how similar they are in terms of strategy, which we call *strategic affinity*.

Point 1 empowers us to differentiate between possible action sequences in terms of utility; naive empowerment is simply counting states whereas this model acknowledges that some potential states won't be as empowered as others. We show how to calculate this forecast of ongoing empowerment in section V-B.

The overlap between the potential futures of each $n$-step sequence of actions causes them to be grouped together when this channel is fed into the impoverishment algorithm outlined in section III-B, which brings about the emergence of strategies instead of arbitrary groups which would have previously been seen.

The effect of this clustering of action sequences, by their *strategic affinity*, can be illustrated easily in a gridworld as in such an scenario it corresponds closely to geographically close states (see Fig. 7); with more complex worlds such a visualization breaks down but the effect of clustering 'nearby' states remains. An example of such a mapping can be seen in Fig. 6. For many games, this grouping already gives an insight into how tasks may be simplified; either by acting as a coarse representation of the problem or as a tool to identify separate local sub-problems that could be dealt with separately.

### B. Reducing strategies to actions

We wish to retain only a subset of actions and will use this clustering of action sequences into strategy groups to select a subset of our original action sequences to form the new action policy for the player.

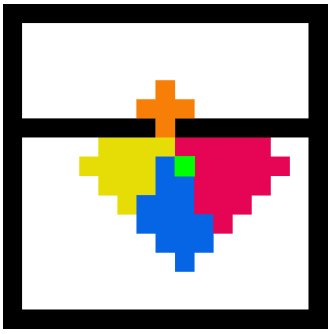

Fig. 6. An example grouping of action sequences, shown here by the colouring of their final states. The example here is a simple one so the states appear end up grouped geographically; the same principle applies to more complex scenarios with many more dimensions. The clustering is an effect of the potential 'shared futures' that similar action sequences may share; a measure of their strategic affinity.

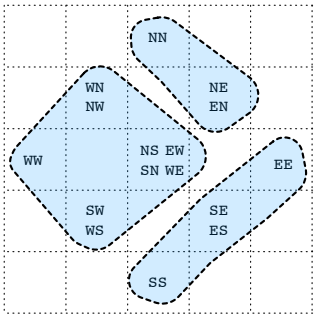


Fig. 7. Visualization of the action sequence grouping from Fig. 4 when the grouping is performed with open-ended empowerment. It can be seen that now the action sequence cluster together into 'strategies' formed of similar action sequences that could potentially lead to the same future states. An example of this grouping in a scenario is shown in Fig. 6.

We use the forecast of the future now contained within the channel to select the action sequence from each strategy group that we forecast will lead to the most empowered state. We will roughly approximate this forecasted empowerment without actually fully calculating the channel capacity that follows each action sequence.

Earlier we formed the channel ($p(s_{t+n+m}|a_t^n, a_{t+n}^m)$) from two separate stages of action sequences, the initial $n$-steps and the subsequent $m$-steps. We now break this channel up into separate channels based on all those where $a_t^n$ is identical, i.e one channel for each case in which the first $n$ steps are identical.

We now have a set of channels, corresponding to each set of $m$-step sequences that stem from common ancestor sequences. For each, we assume an equidistribution over the action sequences, and then measure the mutual information; this is equivalent to testing $I(a_t^{n+m}; S_{t+n+m}|a_t^n)$ for each sequence, $a_t^n$.

The equidistribution and mutual information gives us an approximation of the $(n+m)$-step empowerment for each sequence of $n$-steps; and we can now select, from within each strategy group, those that are most empowered.

As before we must weight this by their likelihood to map to that $g$ (i.e the highest value of $p(g|a_t^n)$ for the given $g$), although once again usually these mappings are deterministic so this is unnecessary.

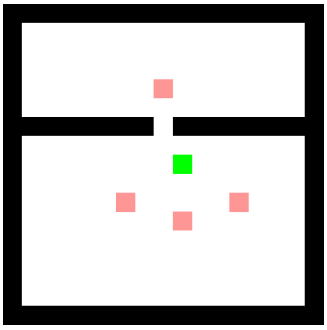

Fig. 8. End states of the 4 action sequences selected to represent the 'strategies' seen in Fig. 6. Each representative state is distant from the walls to improve their individual ongoing empowerment.

For the mapping shown in Fig. 6 this leads to the selection of action sequences with the end states shown in Fig. 8.

It can be seen that this results in collapsing strategies to the action sequences which are forecast to lead to the most empowered states. Without any explicit goal or reward, we are able to identify action sequences which represent different strategies, and that are forecast to have future utility.

We should note that no regard is given to which action sequence was selected from the other group's and so it is possible that action sequences on the borders of the strategy groups may be selected and these might share some possible future with actions selecting from other strategy groups.

For example, in a simple grid world two action sequences that lead to neighbouring states in a large world may sometimes be selected to represent two different strategies. A more complex decomposition of strategy groups to action sequences could address this, however in the present scenarios the results are sufficiently strong without introducing these additional complications and we prefer to keep the approach more general.

### C. Optimizing: Single-step iterations to approximate full empowerment

One major problem of the traditional empowerment computation is the necessity to calculate the channel capacity in view of a whole set of $n-step$ actions. Their number grows exponentially with $n$ and this computation thus becomes infeasible for large $n$.

In (33), we therefore introduced a model whereby we iteratively carried out empowerment computations for short-ranged action sequences, followed by impoverishment phases. While this method was able to extend the empowerment horizon by more than an order of magnitude and managed to locate salient locations in the environment, it turned out to be still more limited than the open-ended empowerment model introduced in the present paper, and we will not present the full model and results here. However, we will briefly introduce a special case of this iterative procedure, which can be used as an optimization when calculating open-ended empowerment.

Imagining we are content for a player to just retain as many action sequences as it needs to reach all possible states it can reach within its empowerment horizon, then it is still in our interests to only retain a single action sequence that takes us to each of these states (in a deterministic world; in a stochastic world we are concerned with the reliability of actions also, and empowerment model handles this too without any adjustment necessary). Note now that in many gridworld scenarios the level of redundancy is extremely high, and given the Markovian nature of the world, nothing is lost by this reduction; the action sequence that leads you to a given state gives no regard to the future possibilities achievable from that state. (We would especially note that games such as Go, Arimaa and Havannah display this same property, with many of the possible move sequences leading to identical states.)

We can now apply the information bottleneck method detailed before with the cardinality of $G$ set to equal the number of states observed; this would usually mean that each group,$G$, would contain all actions leading to a single state. It is therefore sufficient to select a single action sequence from each $G$ and would have a single action sequence leading to each state.

However, with long action sequences the number of sequences usually increases exponentially, and so it is actually more efficient to do the impoverished extension process iteratively, as follows:

1) Generate a list of 1-step action sequences.
2) Calculate empowerment using the current action sequences list as our available actions.
3) Apply a bandwidth constraint using the information bottleneck, with a cardinality equal to the number of states observed when each action sequence is tried, in turn, from the current state.
4) Use the resulting grouping to select a subset of action sequences from the list (as per previous section).
5) Extend each of the retained sequences by 1-step for every possible next action.
6) Return to step 2 until the desired total sequence length is reached.

This process will calculate an identical value for empowerment, in deterministic scenarios, as achieved via the standard empowerment method. If we introduce noise into the scenario, then this method still correlates with the standard empowerment algorithm, and the information bottleneck step of the above process actually eliminates the 'noisier' paths when there are redundant paths to the same state, as they are less empowered. In this case the noise is a probability that in performing any single action a player actually performs another, with some probability. Fig. 9 shows some example data of the correlation between the two methods, including in various scenarios with varying boxes, mazes and noise.

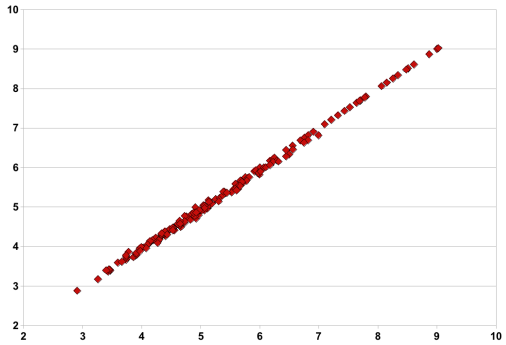


Fig. 9. The correlation between standard empowerment, and single-step iterative empowerment, for 160 data points in 5 different gridworlds, for various levels of noise (from 0% to 16%) and various empowerment horizons (from 1-6 steps). This demonstrates that in a wide range of scenarios empowerment can be cheaply approximated using the single-step iterations method.

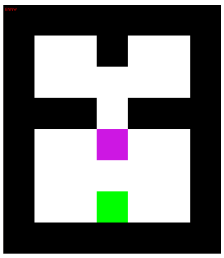

Fig. 10. A Sokoban inspired gridworld scenario. Green represents the player, and purple represents the pushable box, which is blocking the door.

## VI. Game Scenarios and Results

### A. 'Sokoban'

Many puzzle games concern themselves with arranging objects in a small space to clear a path, towards a 'good' configuration, or even to interact with other players. Strategy games often are concerned with route finding and similar such algorithms, and heuristics for these often have to be crafted carefully for a particular game's dynamics.

As an example for such games, we examine a simplified box-pushing scenario inspired by Sokoban. In the original incarnation, each level has a variety of boxes which needed to be pushed (never pulled) by the player into some designated configuration; when this was completed, the player completed the level and progressed to the next. Sokoban has received some attention for the planning problems it introduces (40; 41), and most pertinent approaches to it are explicitly search-based and tuned towards the particular problem.

We are changing the original Sokoban problem insofar, as that in our scenario there are *no* target positions for the boxes, and in fact there is *no* goal or target at all. As stated, we postulate a critical element for General Game Playing is self-motivation that precedes the concretization of tasks, thus we adapt the game accordingly by removing the concept of target spaces for the boxes.

Fig. 10 shows the basic scenario, with a player and a pushable box. Similarly to the earlier gridworlds, the player can move North, South, East and West in any timestep (there is no 'stand still' option). Should the player move into a neighbouring cell occupied by the box, then the box is pushed in the same direction into the next cell; should the destination

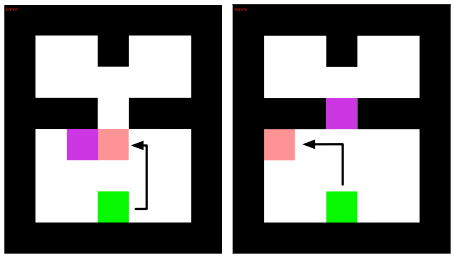

Fig. 11. The 4-step action sequences selected by an AI player constrained to 2 action sequences, selected by open-ended empowerment. The extended horizon in this case is $m$=5. The green cell represents the player's starting position, the pink cell the player's ending position, and the purple the box.

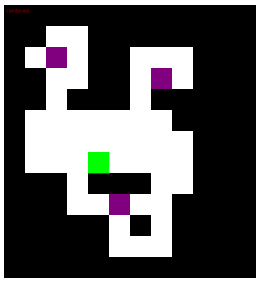

Fig. 12. A second Sokoban scenario, with multiple boxes.

Fig. 13. The 14-step action sequences selected by an AI player constrained to 4 action sequences, selected by open-ended empowerment. The extended horizon in this case is $m$=5. There are 3 boxes 'trapped' within puzzles, each of which has at maximum 6 action sequences that retrieves box to the main room without permanently trapping it. Despite there being *268 million action sequences* to select from and *no explicit goal*, all 3 boxes are retrieved successfully.

cell for the box be blocked then neither the player or the box move and the time step passes without any change in the world.

Our intuition would be that most human players, if presented with this scenario and given 4 time steps to perform a sequence of actions on the understanding that an unknown task will follow in the subsequent time steps, would first consider moves that move the box away from blocking the doorway. Humans, we believe, would understand instinctively from observing the setup of the environment, that the box blocks us from the other room and thus moving it gives us more degrees of freedom.

However, it is not obvious how to enable AI players, without explicit goals and no hand-coded knowledge of their environment, to perform such basic tasks as making this identification. Here we approach this task with the fully generic open-ended empowerment concept. We use the following parameters: $n$=4 and $m$=5, and a bandwidth constraint limiting the player to selecting 2 action sequences from amongst the $4^n = 4^4 = 256$ possible. This constraint was originally selected to see if the AI player would identify both choices for clearing a pathway to the door, and otherwise to allow for a possible contrast in strategies.

In Fig. 11 we can see two action sequences; the thin lines show the path the player took, and the shaded cells the start (green) and end (pink) position of the player subsequent to performing the action sequence (box is purple). The two options for clearing a path to the doorway (box pushed left or right of the door) are sometimes clustered as being the same strategy; when they are not then they are the two actions selected, otherwise the other action shown (or its mirror) is selected.

In Fig. 12 we can see a more complex scenario, with multiple boxes in the world, all of which are 'trapped' in a simple puzzle. Again, there is no explicitly formulated goal in this scenario. There are 3 boxes within puzzles, and for each box there exists a single unique trajectory that will recover the box from the puzzle without leaving it permanently trapped. One should now observe that trapping any box immediately reduces the player's ability to control its environment and costs it any degrees of freedom afforded by being able to manipulate the box.

Fig. 13 shows a typical set of results when allowing the AI player to select 4 action sequences of 14-steps with an extended horizon of 5 steps. An explicit count of the different paths to the doorway for each box's puzzle room reveals that there are only 6 action sequences that 'retrieve' the box to the main room for each of the top 2 boxes, and only one still for the bottom box.

Among the 268 million possible action sequences of length 14 available for the task, open-ended empowerment, with a selection bandwidth of 4 action sequences, will consistently select sequences required for all three puzzles to be 'solved' and the box to be retrieved to the main room in each case. In these solutions, the final action selected is one where the box is pushed into an empty cell before the player returns to the main room.

### *B. Gambler's Problem*

The Gambler's Problem, which was, to our knowledge, first studied in (42), has been addressed several times before, notably in (43) where Reinforcement Learning is used to solve it. The scenario, as presented here, is one in which a gambler, with some current balance of money given by a

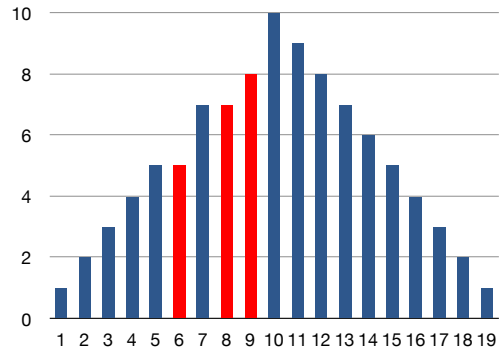


Fig. 14. A typical betting policy produced by open-ended empowerment where $n$=1, $m$=4, and the probability to in a bet $p$=0.4, with a target balance $t = 20$. Other than the 3 highlighted values, the betting policy equates to the bold play strategy proved optimal in (42).

positive integer, $b$, with $b < t$ where $t$ is a target balance, must repeatedly make bets until he is either bankrupt or reaches (or exceeds) the target balance, $t$. In each betting round, the gambler wagers some positive integer value $w$ where $w \leq b$ and has a probability of winning given by $p$; in this paper we only examine the case $p$=0.4 (as in (43)). If the bet is won, then the gambler's balance is increased by the amount wagered, and if the bet is lost, then their balance is decreased accordingly. Thus, the new balance becomes:

$$b' = b + w \text{ with probability } p$$
$$b' = b - w \text{ with probability } 1 - p$$

The objective is to find the optimal betting policy: the best bet possible for all possible balances (i.e those that maximize the probability of reaching the target balance). A general property of the problem is that for $p > 0.5$, the gambler will try to prolong the game as much as possible; our case, with $p < 0.5$ is more interesting, and the optimal strategies (which are not unique in general) will strive to make as few bets as possible since each bet is more likely to lose than to win and so prolonging the betting is a bad proposition.

This problem is an exception to the others in this paper, as the scenario necessitates an explicit goal state. It demonstrates how an explicit goal state, if necessary, can be incorporated into the open-ended empowerment formalism. Here, we handle this by creating a 'paradise' state where any bets placed whilst having a balance above the target threshold would be a winning bet. This 'paradise' state does not explicitly define higher balances to be better, but by allowing a player to exactly predict the outcome of any move the player's empowerment is increased. Furthermore, an increased budget provides for an increased set of possible moves (you have more options what to bet) which also increases empowerment. So the attraction to this paradise state comes from the elimination of stochasticity, and the increase in the number of available moves.

Fig. 14 shows the results of applying open-ended empowerment to the Gambler's Problem using open-ended empowerment, where $n$=1 and $m$=4. The algorithm was run for each starting balance, and then a bandwidth constraint was applied to select a single action, which would be the action contributing the most to the empowerment. This process produced a betting policy for what amount, $w$, to bet for each starting balance, $b$, shown by the graph.

The channel $p(s_{t+1+m}|a)$ gives a probability of ending in each of the possible states based on an approximation of the capacity achieving distribution across the future $m$ actions that would follow for the bet $a$.

Key states to pay attention to are the *pivot* balance values of 5, 10, and 15; these are points where there is an obvious best bet to make. At 15, it is clear to see you should bet 5, and you either win, or fall back to the next pivot point of 10. At 10, where the player's balance matches exactly the amount it needs yet to win, it seems clear that a single bet of the entire balance is the best strategy (there is no advantageous 'fall back' position as with case of 15). The case of 5 can be thought of, in this instances, as a 'sub game' where the player first needs to get to the immediate goal of 10; again there is no fallback and betting our entire budget makes sense and in the case of success, one reaches $b = 10$ with the opportunity to win with the next move.

In particular, for the Reinforcement Learning Gambler's problem solution from (43), it is the pivot points that stand out in the policy space. Moving to empowerment we can see it also performs optimally at the pivot points, betting the correct values.

The results show the betting strategy can be seen to be a sensible betting strategy that can be discovered with an approximation of the future and a limited horizon. The policy equates almost perfectly to the *bold play* policy that was proven to be (non-uniquely) optimal in (42). Repeated runs of the problem using open-ended empowerment all follow the same typical pattern with a near optimal betting policy.

This example shows that the principles presented of maximising control, 'folding back' future states and trying to maintain future control extend beyond the realm of deterministic gridworld scenarios and into non-deterministic scenarios in other spaces. It also demonstrates possible strategies to incorporate explicit goals seamlessly into the empowerment framework when this is required.

### C. 'Pacman'

Pacman and its variants have been studied previously, included using a tree search approach (44), but the aim of the current paper is not to attempt to achieve the performance of these methods but rather to demonstrate that, notwithstanding their genericness, self-motivation concepts such as open-ended empowerment are capable of identifying sensible goals of operation in these scenarios on their own and use these to perform the scenario tasks to a good level.

Thus, the final scenario we present is a simplified version of Pacman; rather than having pills to collect, and any score, the game is simplified to having a set of ghosts that hunt Pacman and kill him should they catch him. The 'score', which we will measure, is given simply by the time-steps that Pacman survives before he is killed; however it is important to note

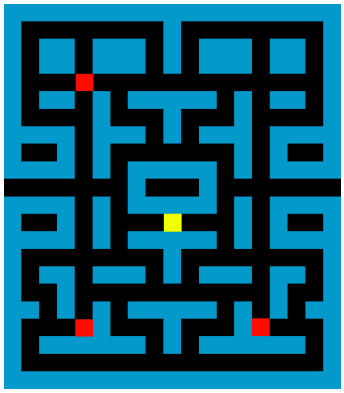

Fig. 15. Pacman scenario, showing the starting position of the player (yellow) in the center, and of 3 ghosts (red).

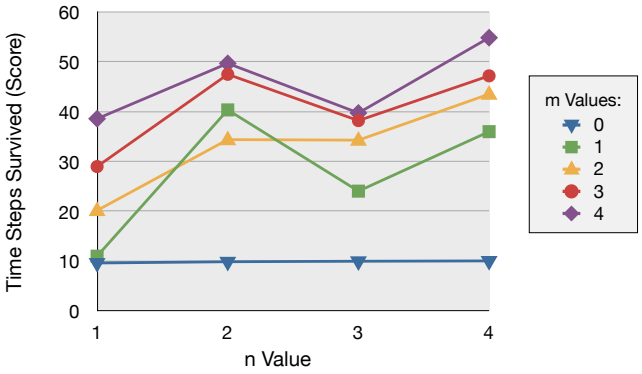


Fig. 16. Pacman's 'score' for different parameter sets, averaged over 3 different starting positions and 1500-3000 games per data point.

that our algorithm will not be given an explicit goal, rather the implicit aim is simply survival. If humans play the game for a few times, it is plausible to assume (and we would also claim some anecdotal evidence for that) that they will quickly decide that survival is the goal of the game without being told. Choosing survival as your strategy is a perfectly natural decision; assuming no further knowledge/constraints beyond the game dynamics, and a single-player game, anything that may or may not happen later has your survival in the present as its precondition.

In the original Pacman game, each ghost uses a unique strategy (to add variation and improve the gameplay) and they were not designed to be ruthlessly efficient; the ghosts in our scenario are far more efficient and all use the same algorithm. Here, in each timestep, Pacman makes a move (there is no 'do nothing' action, but he can indirectly achieve it by moving towards a neighbouring wall), and then the ghosts, in turn, calculate the shortest path to his new location and move. Should multiple routes have the same distance, then the ghosts randomly decide between them. They penalise a route which has another ghost already on it by adding $d$ extra steps to that route; setting $d = 0$ results in the ghosts dumbly following one another in a chain which is easy for Pacman. Increasing the value makes the ghosts swarm Pacman more efficiently. For the results presented here we use $d = 8$ which is a good compromise between ghost efficiency and giving Pacman sufficient chance to survive for long enough to allow different values for $n$ and $m$ to differentiate.

The maze setup we used is shown in Fig. 15, and the location of the 3 ghosts can be seen. Having only 3 ghosts is another compromise for the same reasons as above; using 4 ghosts did not result in Pacman surviving long enough to get meaningful variance in the results generated with different parameter sets.

Pacman has a model of the ghosts' algorithm and thus can predict their paths with some accuracy, and is being allowed 4 samples of their possible future positions (which are stochastic given the possibility that for one or more ghosts the path lengths coincide) for a given move of his. However, once no equal routes are present then 1 sample is perfect information, but once one or more ghosts has one or more equal length paths, then the sampling becomes less accurate and may lose information about the possible future ghost moves.

The game begins with Pacman's first move, and continues until he is caught by any of the ghosts; at this point Pacman is 'dead' and is no longer allowed to move. However, there is no special 'death' state in our model; once caught, Pacman is no longer allowed to move but can still observe the game state (which quickly stops changing anyway, as the ghosts no longer move having swarmed Pacman).

We ran Pacman using the above algorithm, and the result is that Pacman flees from the ghosts once they come into his horizon; this result from the fact that his future control over the state of the game would drop to zero should he be caught. Death translates directly into the empowerment concept by a vanishing empowerment level. Figure 16 shows the results for various parameter sets, for 3 different starting positions; for each combination of starting position, $n$ value and $m$ value we ran 500-1000 games, then averaged the number of time steps survived over the starting positions for a final average 'score' for each combination of $n$ and $m$.

Firstly, it can be seen that for $m = 0$, which is equivalent to 'standard' empowerment (25; 26) and does not make use of any features of the algorithm presented that increasing the value of $n$ has no impact on Pacman's performance. Without a second horizon and thus some measure of his control over the game in the future there is no pressure to maintain that control. Colloquially, we could say Pacman only cares about his empowerment in the present moment, not at all about the future. Being able to reach a future state in which he is dead or trapped seems just as good as being able to reach a future state in which he still has a high empowerment; the result is he does not even try to avoid the ghosts and is easily caught.

Once Pacman has even a small ongoing horizon with $m = 1$ it is easy to see the increase in performance, and with each increase in $m$ performance improves further as Pacman is better able to predict his future state beyond what action sequence he plans to perform next. For all cases where $m > 0$ it can be seen there is a general trend that increasing $n$ is matched with increasing performance, which would be expected; planning further ahead improves your chances to avoiding the ghosts

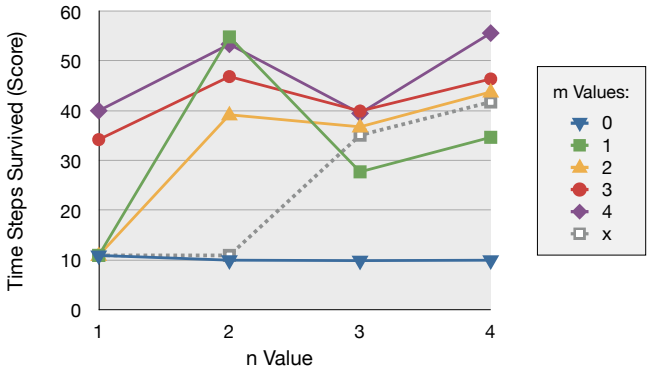


Fig. 17. Pacman's 'score' for his traditional starting position as seen in Fig. 15. The dotted line labelled 'x' corresponds to using standard greedy empowerment hill-climbing, which is far more expensive computationally.

and finding areas of continued high empowerment.

However, it can be noted in Fig. 17 that $n = 2$, $m = 1$ performs extremely well and outside of the fit of the other results; this seems, from inspection of individual runs, to be an artifact of the design of the world, combined with the properties of this specific starting position, and does not persist that strongly when the starting position is changed. This highlights how a given structure or precondition in a world, which is not immediately observable, could be exploited by specific, hand-crafted AI approaches unique to that exact situation but would be difficult to transfer to other scenarios.

One interesting, non-trivial, behaviour that was consistently consistently emerged from the open-ended empowerment algorithm applied to Pacman was a technique he would use to 'pull ghosts in'; his employed strategy favoured having a ghost in the immediate cell behind him (this makes that particular ghosts behaviour completely predictable and not only reduces Pacman's uncertainty about the future but also increases his ability to control it). Therefore, Pacman could often be observed moving back and forth between two cells waiting for a nearby ghost to get to such a position; however, in our observations this did not happen when other ghosts are nearby which would result in the danger of Pacman being surrounded. This behaviour is not something that would seem intuitive to a human player, and whether skirting danger in such a way is desirable in other scenarios is hard to predict.

Another example of an interesting behaviour that could be consistently observed for very select combinations of $n$, $m$ and $d$ was that Pacman would always start out moving in an eastwards direction (even with the ghost at the top left removed), rather than randomly choosing between the two seemingly symmetric options. After detailed analysis, this was attributed to the fact that the ghost to the East had a higher index in the software code and would take its move before that in the West, which had the result that at a pivotal moment Pacman could use the Eastern ghost to block the path that the Western ghost was following, forcing it to double back. This would lead to a extremely small advantage for Pacman travelling East to begin instead of West which was exploited by open-ended empowerment.

Finally, we also compared the Pacman scenario with a greedy hill-climbing implementation of standard empowerment which has been used in (28, 26) to select favoured moves without external reward function. With this approach Pacman tries each of his 4 available moves and then measures the empowerment (with varying horizon values) and then selects the move which leads to the highest empowerment. This is a considerably more computation-intensive process than using open-ended empowerment, and as can be seen in Fig. 16, the results do not perform as well as the open-ended empowerment method. This is because the overall forecast of the future is shorter; with $m = 4$ and $n = 4$, the horizon into the future is 8 steps, which would be computationally inaccessible with standard empowerment. Furthermore, greedy hill-climbing can only be used to select a single action, so in other scenarios (such as the Sokoban scenario presented) it cannot be used effectively as it cannot determine different strategies as open-ended empowerment does.

## VII. Discussion

The presented open-ended empowerment method exhibits primarily two powerful features, both of which require no hand-coded heuristic:

- the ability to assign sensible 'anticipated utility' values to states where, as of yet, no task or goal has been explicitly specified.
- accounting for the strategic affinity between potential action sequences, as implicitly measured by the overlap in the distribution of their potential future states (naively this can be thought of as how many states that are reachable from A are also reachable from B within the same horizon). With this the player is able to select a subset of action sequences that fit within a specific bandwidth limit whilst ensuring that they represent a diversity of strategies.

This clustering of action-sequences allows strategic problems to be approached with a coarser grain; by grouping sets of actions together into a common strategy different strategies can be explored without requiring that every possible action sequence is explored. We believe that such a grouping moves towards a 'cognitively' more plausible perspective which groups strategies a priori according to classes of strategic relevance rather than blindly evaluating an extremely large number of possible moves. Furthermore, by modifying the bandwidth, the concept of having strategies with differing granularities (i.e 'attack' versus 'attack from the north' and 'attack from the south' etc.) emerges. As an aside, it has been shown that there is a strong urge to compress environments and tasks in such a way (45), with the ability to compress one situation better than another making it more attractive.

Before, however, we go into a more detailed discussion of the approach in the context of games, some comments are in place why a heuristic which is based only on the structure of a game and does not take the ultimate game goal into account, can work at all. It is not obvious that this should work and seems, on first sight, to contradict the rich body of

work on reward-based action selection (e.g. grounded in utility theory/reinforcement learning etc.).

To resolve this apparent paradox, one should note that for many games, the structure of the game rules already implicitly encodes partial aspects of the ultimate tasks to a significant degree. For instance, Pacman by its very nature is a survival game. Empowerment is immediately reflecting survival, as a 'dead' player loses all empowerment.

### A. Application to Games

In the context of tree search, the ability to cluster action-sequences into strategies introduces the opportunity to imbue game states and corresponding actions with a relatedness which derives from the intrinsic structure of the game and is not externally imposed by human analysis and introspection of the game.

The game tree could now be looked at from a higher level, where the branches represent strategies, and the nodes represent groups of similar states. It is possible to foresee pruning a tree at the level of thus determined strategies rather than individual actions, incurring massive efficiency gains. More importantly, however, these strategies emerge purely from the structure of the game rather than from an externally imposed or assumed semantics.

In complex or subtle game states, the tree could be pruned initially at the level of strategies and then either run again in order to break the strategies into more refined groups, or examined at the individual actions level.

In many games, it is reasonable to assume having perfect knowledge of transitions in the game state given a move. However, note that the above model is fully robust to the introduction of probabilistic transitions, be it through noise, incomplete information or simultaneous selection of moves by the opponent. The only precondition is is the assumption that one can build a probabilistic model of the dynamics of the system. Such opponent or environment models can be learned adaptively (46; 35; 47). The quality of the model will determine the quality of the generated dynamics, however, we do not investigate this here further. We limit ourselves to discuss how this could look by sketching two viable methods to generate an 'opponent' policy in the current context.

Designate the AI player as the 'hero' and the opponent player as the 'antagonist'. Then the model presented in this paper itself can be used to create a policy for our antagonist by operating with the naive assumption that hero moves either randomly or to maximize its empowerment. It is unlikely that this will provide a perfect approximation of the antagonist policy, but similar dependencies and layered interaction between player models have already been studied in a information-theoretic context and shown to produce non-trivial and intricate behaviours which encompass various instances of behaviours ranging between antagonistic and cooperative (48; 49; 50).

Alternatively, access to any prior observations of the environment, and an ability to distinguish between the antagonist and the rest of the environment (which is usually the case for games), would allow the construction of a policy based on previously observed actions of the antagonist in certain states.

Either of these two methods for predicting an opponent policy could form part of an evolving strategy, where policies are evolved and updated as more observations are made. Furthermore, they lend themselves to working together with a method of building heuristics more specific to this game or scenario, whilst allowing the player to act sensibly and survive long enough while still learning and refining these more specific heuristics.

We have illustrated the efficacy of this approach using three scenarios. Importantly, in all three scenarios we use the same algorithm. In other words, the algorithm was not specifically crafted to suit the particular scenario, but is generic and transfers directly to other examples. For reasons of argument, we review the scenarios in the reverse order of appearance in the paper.

In the Pacman scenario, we demonstrated that acting to maintain anticipated future empowerment is sufficient to provide a strong generic strategy for Pacman. More precisely, the player, without being set an explicit goal, made the 'natural' decision to flee the ghosts. This behaviour derives from the fact that empowerment is by its very nature a 'survival-type' measure, with death being a 'zero-empowerment' state. By the use of the second horizon's approximate forecast of the future, the player was able to use the essential basic empowerment principle to successfully evade capture for extended periods of time.

The Gambler's Problem served to illustrate that the empowerment generalizes outside of grid scenarios, and can handle stochastic tasks, which traditionally have required specific hand-coded or reinforcement learning approaches. Whilst the policy generated in our studies does not correspond to the perfectly optimal policy obtained via usual reinforcement learning methods, it is very close to it, which is striking considering the generality of the method. We introduced the 'paradise state' in this scenario because this scenario terminates the game on success, which currently this open-ended empowerment model cannot account for. This addition was made to the game, and the algorithm was unchanged.

The most illuminative example, with respect to the power of the method in identifying strategy classes, is the Sokoban scenario. The scenario presented 3 trapped boxes each requiring a 14-step action sequence to 'retrieve' from the trap. A human introspecting the problem could deduce the desirable target states for each box (freeing them), and identify there exists only 13 action sequences which would lead to one of these states. With a total of $268 \times 10^6$ possible action sequences to choose from, and lacking the *a priori* knowledge determining which states should be target states, the algorithm reliably selects a set of action sequences which includes an action sequence for retrieving each of the boxes. Not only are the target states identified as being important but the possible action sequences to recover each of the different boxes are identified as belonging to a different strategy.

The importance of this result lies in the fact that, while again, the approach used is fully generic, it nevertheless gives rise to distinct strategies which would be preferred also based on human inspection. This result is also important for the practical relevance of the approach. The above relevant

solutions are found consistently, notwithstanding the quite considerable number and depth of possible action sequences. We suggest that this may shed additional light on how to construct cognitively plausible mechanisms which would allow AI agents to preselect candidates for viable mid-term strategies without requiring full exploration of the space.

Implicit for this consideration to work is that the typical game scenario is not a worst-case scenario (in terms of structural predictability of the game, not in terms of antagonistic opponents, of course). In other words, there is significant relation between local and global structure which, in typical games, makes the predictions (on which empowerment relies) meaningful. Of course, it is possible to specifically design structurally "worst-case" games which have no structural predictability, and in particular no heuristics short of a full expansion of the game tree (51). However, this is not typical for many strategy games and puzzles which humans prefer and which seem to exhibit quite a significant amount of structural coherence (52). We thus feel that it is legitimate to assume the existence of prediction-based algorithms that, by virtue of such structural coherence, can filter out relevant action sequences in an effective manner - the experiments in the present paper suggest that open-ended empowerment is a suitable candidate for such an approach. Of course, we expect that there will be different ways to achieve that, in which case the success of the still quite straightforward open-ended empowerment formalism opens a very promising line for future research.

### *B. Relation to Monte-Carlo Tree Search*

The presented formalism consists essentially of just dividing a transition table into two halves and using the knowledge of the probability of encountering each of the states in the second half of the table, forecast by the distribution of actions considered for each state in the first half, to differentiate those states in the first half and assign them some estimated utility. The information-theoretic approach allows this to be quantifiable and easily accessible to analysis. However, we believe the technique presented would work using other methodologies and could be combined with other techniques in the mid-term. One important example of where it could be applied would be in conjunction with a Monte-Carlo Tree Search approach, and we would like to discuss below how the formalism presented in this paper may provide pathways to address a number of weaknesses with MCTS.

MCTS has been seen to struggle with being a global search in problems with a lot of 'local structure' (53). An example for this is a weakness seen in the Go progam Fuego, which is identified as having territorially weak play (54) because of this problem. Some method which clusters of action sequences into strategies, where the stategic affinity 'distance' between the subsequent states is low, might allow for the tree search to partially operate at the level the strategies instead of single actions and this could help in addressing the problem.

The second aspect of MCTS which has led to some criticism is that it relies on evaluating states to the depth of the terminal nodes they lead to in order to evaluate a state. It is possible that the 'folding back' model of empowerment presented in this paper could be used as a method to evaluate states in an MCTS, which may operate within a defined horizon when no terminal states appear within that horizon. In this way the search could be done without this terminal state requirement, and this might allow a better balance between the depth of the search versus its sparsity. This, of course, would make use of the implicit assumption of structural predictability underlying our formalism.

## VIII. Conclusion

We have proposed open-ended empowerment as a candidate for solving implicit 'problems' which are defined by the environment's dynamics without imposing an externally defined reward. We argued that these cases are of a type that are intuitively noticed by human players when first exploring a new game, but which computers struggle to identify.

Importantly, it is seen that this clustering of action sequences into strategies determined by their strategic affinity, combined with aiming to maintain a high level of empowerment (or naive mobility in simpler scenarios) brings about a form of 'self-motivation'. It seems that setting out to maximize the agent's future control over a scenario produces action policies which are intuitively preferred by humans. In addition, the grouping of action sequences into strategies ensures that the 'solutions' produced are diverse in nature, offering a wider selection of options instead of all converging to micro-solutions in the same part of the problem at the expense of other parts. The philosophy of the approach is akin to best preparing the scenario for the agent to maximize its influence so as to react most effectively to an as yet to emerge goal.

In the context of general game-playing, to create an AI that can play new or previously un-encountered games, it is critical to shed its reliance on externally created heuristics (e.g. by humans) and enable it to discover its own. In order to do this, we propose that it will need a level of self-motivation and a general method for assigning preference to states as well as for identifying which actions should be grouped into similar strategies. Open-ended empowerment provides a starting point into how we may begin going about this.

## APPENDIX A
## OPEN-ENDED EMPOWERMENT COMPLETE ALGORITHM

The open-ended empowerment algorithm consists of two main phases. This appendix presents the complete algorithm. In order to make it somewhat independent and concise, it introduces some notation not used in the main paper.

Phase 1 is not strictly necessary, but acts as a powerful optimization by vastly reducing the number of action sequences that need to be analysed. The main contributions presented in this paper are within phase 2.

### A. Setup

1) Begin with an empty set containing a list of action sequences, $\mathcal{A}$. Set $n = 0$.
2) Define a set $\mathcal{P}$ as the list of all possible single-step actions available to the player.

### B. Phase 1

Phase 1 serves to create a set of action sequences $\mathcal{A}$ that, most likely, will reach all possible states within $n$-steps, but will have very few (0 in a deterministic scenario) redundant sequences. In stochastic scenarios that have heterogeneous noise in the environment it may be that those areas are avoided in preference to staying within more stable states, and in these cases you will find there may be some redundancy in terms of multiple action sequences to the same state.

In a deterministic scenario phase 1 can be entirely skipped; the same optimization can be achieved by selecting at random a single action sequence for each state reachable within $n$-steps (i.e for each state $s$ select any single action sequence where $p(s|a_t^n) = 1$).

3) Produce an extended list of action sequences by forming each possible extension for every action sequence in $\mathcal{A}$ using every action in $\mathcal{P}$; the number of resultant action sequences should equal $|\mathcal{A}| \cdot |\mathcal{P}|$. Replace $\mathcal{A}$ with this new list and increment $n$ by 1.
    i. Using the channel/transition table, $p(s_{t+n}|a_t^n)$, note the number of unique states reachable using the action sequences in $\mathcal{A}$, always starting from the current state. This will be labelled $\sigma$.
4) Produce $p(a_t^n)$ from $\mathcal{A}$, assuming an equi-distribution on $\mathcal{A}$. Using this, combined with $p(s_{t+n}|a_t^n)$, as inputs to the Information Bottleneck algorithm (we recommend the implementation at (39), pp. 41). For the cardinality of $U$, our groups of actions (labelled $T$ in (39)), use $\sigma$ such that the number of groups matches the number of observed states. This will produce a mapping, $p(g|a_t^n)$, which will typically be a hard mapping in game scenarios. Select a random value of $a$ from each $U$ (choosing $argmax_{a_t^n} p(u|a_t^n)$ in cases where it is not a hard mapping). Form a set from these selected values, and use this set to replace $\mathcal{A}$.
5) Loop over steps 3 and 4 until $n$ reaches the desired length.

### C. Phase 2

Phase 2 will extend these base $n$-step sequences to extended sequences of $n + m$-steps, before collapsing them again such that we retain only a set of n-step sequences which can forecast their own futures in the following $m$-steps available to them.

6) Produce a list of action sequences, $\mathcal{M}$, by forming every possible $m$-step sequence of actions from the actions in $\mathcal{P}$.
7) Produce an extended list of action sequences by forming each possible extension for every action sequence in $\mathcal{A}$ using every action sequence in $\mathcal{M}$. Replace $\mathcal{A}$ with this new list.
8) Create a channel $p(s_{t+n+m}|a_t^{n+m})$ (where $a_t^{n+m} \in \mathcal{A}$) by sampling from the environmental dynamics for our current state (using the game's transition table). For environments with other players, one can use any approximation of their behaviour available and sample over multiple runs or, lacking that, model them with greedy empowerment maximisation based on a small horizon.
9) Now collapse $p(s_{t+n+m}|a_t^{n+m})$ to $p(s_{t+n+m}|a_t^n)$ by marginalizing over the equally distributed extension of the action sequences:

$$p(s_{t+n+m}|a_t^n) = \frac{\sum\limits_{a_{t+n}^m} p(s_{t+n+m}|a_t^n, a_{t+n}^m)}{\left|A_{t+n}^m\right|}$$

where

$$p(s_{t+n+m}|a_t^{n+m}) \equiv p(s_{t+n+m}|a_t^n, a_{t+n}^m)$$

10) Apply the Information Bottleneck as in (39), pp. 41 to reduce this to a mapping of action sequences to strategies, $p(g|a_t^n)$ where $G$ are our groups of action sequences grouped into strategies. Cardinality of $G$ sets how many strategy groups you wish to select.
11) We now need to select a representative action from each group, $g$; for this we assume an equi-distribution of future actions $p(a_{t+n}^m)$, then select from each group the representative action $a^{(\text{rep})}$ which maximises approximated future empowerment (and weight this on how well the action represents the strategy, $p(g|a_t^n)$, which is relevant if $p(g|a_t^n)$ is not deterministic):

$$a^{(\text{rep})}\big(p(s_{t+n+m}|a_t^n, a_{t+n}^m), g\big) =$$
$$\arg\max_{a_t^n} \left( p(g|a_t^n) \cdot \sum_{a_{t+n}^m \in A_{t+n}^m} I(a_{t+n}^m; S_{t+n+m}) \right)$$

We can now form a distribution of $n$-step action sequences from the value $a^{(\text{rep})}$ for each action group; these represent a variety of strategies whilst aiming to maximise future empowerment within those strategies.

## REFERENCES


[1] S. Margulies, "Principles of Beauty," *Psychological Reports*, vol. 41, pp. 3–11, 1977.

[2] J. von Neumann, "Zur Theorie der Gesellschaftsspiele," *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[3] C. E. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, vol. 41, no. 314, p. 256–275, 1950.

[4] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal*, vol. 11, pp. 601–617, 1967.

[5] B. Weber, "Mean Chess-Playing Computer Tears at Meaning of Thought," *New York Times*, Feburary 1996.

[6] J. McCarthy, "What is Artificial Intelligence?" 2007. [Online]. Available: http://www-formal.stanford.edu/jmc/whatisai/

[7] O. Syed and A. Syed, "Arimaa - a new game designed to be difficult for computers," *International Computer Games Association Journal*, vol. 26, pp. 138–139, 2003.

[8] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck, "Monte-Carlo Tree Search: A New Framework for Game AI," in *AIIDE*, 2008.

[9] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo Planning," in *In: ECML-06. Number 4212 in LNCS*. Springer, 2006, pp. 282–293.

[10] R. Pfeifer and J. C. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press, 2006.

[11] F. J. Varela, E. T. Thompson, and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*, new edition ed. The MIT Press, Nov. 1992.

[12] T. Quick, K. Dautenhahn, C. L. Nehaniv, and G. Roberts, "On Bots and Bacteria: Ontology Independent Embodiment," in *Proc. of 5th European Conference on Artificial Life (ECAL)*, 1999, pp. 339–343.

[13] H. A. Simon, *Models of man: social and rational; mathematical essays on rational human behavior in a social setting*. New York: Wiley, 1957.

[14] O. E. Williamson, "The Economics of Organization: The Transaction Cost Approach," *The American Journal of Sociology*, vol. 87, no. 3, pp. 548–577, 1981.

[15] C. A. Tisdell, *Bounded rationality and economic evolution : a contribution to decision making, economics, and management*. Edward Elgar, Cheltenham, UK, 1996.

[16] D. A. McAllester, "PAC-Bayesian Model Averaging," in *In Proceedings of the Twelfth Annual Conference on Computational Learning Theory*. ACM Press, 1999, pp. 164–170.

[17] N. Tishby and D. Polani, "Information Theory of Decisions and Actions." in *Perception-Reason-Action Cycle: Models, Algorithms and Systems*, V. Cutsuridis, A. Hussain, and J. Taylor, Eds. Springer, 2010 (in press).

[18] F. Attneave, "Some Informational Aspects of Visual Perception," *Psychological Review*, vol. 61, no. 3, pp. 183–193, 1954.

[19] H. B. Barlow, "Possible Principles Underlying the Transformations of Sensory Messages," in *Sensory Communication: Contributions to the Symposium on Principles of Sensory Communication*, W. A. Rosenblith, Ed. The M.I.T. Press, 1959, pp. 217–234.

[20] ——, "Redundancy Reduction Revisited," *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 241–253, 2001.

[21] J. J. Atick, "Could Information Theory Provide an Ecological Theory of Sensory Processing," *Network: Computation in Neural Systems*, vol. 3, no. 2, pp. 213–251, May 1992.

[22] M. Prokopenko, V. Gerasimov, and I. Tanev, "Evolving Spatiotemporal Coordination in a Modular Robotic System," in *From Animals to Animats 9: 9th International Conference on the Simulation of Adaptive Behavior (SAB 2006), Rome, Italy, September 25-29 2006*, ser. Lecture Notes in Computer Science, S. Nolfi, G. Baldassarre, R. Calabretta, J. Hallam, D. Marocco, J.-A. Meyer, and D. Parisi, Eds., vol. 4095. Springer, 2006, pp. 558–569.

[23] W. Bialek, I. Nemenman, and N. Tishby, "Predictability, Complexity, and Learning," *Neural Comp.*, vol. 13, no. 11, pp. 2409–2463, 2001. [Online]. Available: http://neco.mitpress.org/cgi/content/abstract/13/11/2409

[24] N. Ay, N. Bertschinger, R. Der, F. Guettler, and E. Olbrich, "Predictive Information and Explorative Behavior of Autonomous Robots," *European Physical Journal B*, 2008.

[25] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Empowerment: A Universal Agent-Centric Measure of Control," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 1. IEEE Press, 2005, pp. 128–135.

[26] ——, "All Else Being Equal Be Empowered," in *Advances in Artificial Life: Proceedings of the 8th European Conference on Artificial Life*, ser. Lecture Notes in Artificial Intelligence, M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, Eds., vol. 3630. Springer, Sep 2005, pp. 744–753.

[27] E. Slater, "Statistics for the chess computer and the factor of mobility," *Information Theory, IRE Professional Group on*, vol. 1, no. 1, pp. 150 –152, Feb 1953.

[28] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems," *PLoS ONE*, vol. 3, no. 12, 12 2008.

[29] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[30] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[31] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Organization of the Information Flow in the Perception-Action Loop of Evolved Agents," in *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, R. S. Zebulum, D. Gwaltney, G. Hornby, D. Keymeulen, J. Lohn, and A. Stoica, Eds. IEEE Computer Society, 2004, pp. 177–180.

[32] R. Blahut, "Computation of Channel Capacity and Rate Distortion Functions," *IEEE Transactions on Information*

*Theory*, vol. 18, no. 4, pp. 460–473, Jul 1972.

[33] T. Anthony, D. Polani, and C. L. Nehaniv, "Impoverished Empowerment: 'Meaningful' Action Sequence Generation through Bandwidth Limitation," in *Proc. European Conference on Artificial Life 2009*. Springer, 2009.

[34] ——, "On Preferred States of Agents: how Global Structure is reflected in Local Structure," in *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, S. Bullock, J. Noble, R. Watson, and M. A. Bedau, Eds. MIT Press, Cambridge, MA, 2008, pp. 25–32.

[35] T. Jung, D. Polani, and P. Stone, "Empowerment for continuous agent-environment systems," *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, vol. 19, pp. 16–39, February 2011.

[36] D. Polani, "Information: currency of life?" *HFSP journal*, vol. 3, no. 5, pp. 307–16, 2009.

[37] S. B. Laughlin, R. R. De Ruyter Van Steveninck, and J. C. Anderson, "The metabolic cost of neural information." *Nature Neuroscience*, vol. 1, no. 1, pp. 36–41, 1998.

[38] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999, pp. 368–377.

[39] N. Slonim, "The Information Bottleneck: Theory And Applications," Ph.D. dissertation, The Hebrew University, 2003.

[40] A. Junghanns and J. Schaeffer, "Sokoban: A Case-Study in the Application of Domain Knowledge in General Search Enhancements to Increase Efficiency in Single-Agent Search," *Artificial Intelligence, special issue on search*, 2000.

[41] D. Dor and U. Zwick, "SOKOBAN and other motion planning problems," *Comput. Geom. Theory Appl.*, vol. 13, no. 4, pp. 215–228, 1999.

[42] L. E. Dubins and L. J. Savage, *Inequalities for Stochastic Processes: How to Gamble If You Must*. Dover, New York, 1976.

[43] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[44] D. Robles and S. M. Lucas, "A simple tree search method for playing Ms. Pac-Man," in *CIG'09: Proceedings of the 5th International Conference on Computational Intelligence and Games*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 249–255.

[45] J. Schmidhuber, "Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes," *CoRR*, vol. abs/0812.4360, 2008.

[46] M. Harder, D. Polani, and C. L. Nehaniv, "Two Agents Acting as One," in *Artificial Life XII: The 12th International Conference on the Synthesis and Simulation of Living Systems*, H. Fellermann, M. Dörr, M. Hanczyc, L. L. Ladegaard, S. Maurer, D. Merkle, P.-A. Monnard, K. Støoy, and S. Rasmussen, Eds. MIT Press, Cambridge, MA, 2010, pp. 599–606.

[47] P. Capdepuy, D. Polani, and C. L. Nehaniv, "Constructing the Basic Umwelt of Artificial Agents: An Information-Theoretic Approach," in *Proceedings of the Ninth European Conference on Artificial Life*, ser. LNCS/LNAI, F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, and A. Coutinho, Eds., vol. 4648. Springer, 2007, pp. 375–383.

[48] ——, "Maximization of Potential Information Flow as a Universal Utility for Collective Behaviour," in *Proceedings of the First IEEE Symposium on Artificial Life*, 2007.

[49] ——, "Perception-Action Loops of Multiple Agents: Informational Aspects and the Impact of Coordination," *Special Issue of Theory in Biosciences on Guided Self-Organisation*, 2011, accepted.

[50] C. Salge and D. Polani, "Digested Information as an Information Theoretic Motivation for Social Interaction," *Journal of Artificial Societies and Social Simulation*, vol. 14, no. 1, p. 5, 2011.

[51] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

[52] C. Salge, C. Lipski, T. Mahlmann, and B. Mathiak, "Using genetically optimized artificial intelligence to improve gameplaying fun for strategical games," in *Sandbox '08: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*. New York, NY, USA: ACM, 2008, pp. 7–14.

[53] M. Müller, "Challenges in Monte-Carlo Tree Search," 2010, unpublished. [Online]. Available: http://www.aigamesnetwork.org/_media/main:events:london2010-mcts-challenges.pdf

[54] ——, "Fuego-GB Prototype at the Human machine competition in Barcelona 2010: A Tournament Report and Analysis," University of Alberta, Tech. Rep. TR10-08, 2010.