Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Granular Computing in Actor-Critic Learning

James F. Peters
Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba R3T 5V6, Canada
Email: jfpeters@ee.umanitoba.ca

Abstract— **The problem considered in this paper is how to guide actor-critic learning based on information granules that reflect knowledge about acceptable behavior patterns. The solution to this problem stems from approximation spaces, which were introduced by Zdzisław Pawlak starting in the early 1980s and which provide a basis for perception of objects that are imperfectly known. It was also observed by Ewa Orłowska in 1982 that approximation spaces serve as a formal counterpart of perception, or observation. In our case, approximation spaces provide a ground for deriving pattern-based behaviours as well as information granules that can be used to influence the policy structure of an actor in a beneficial way. This paper includes the results of a recent study of swarm behavior by collections of biologically-inspired bots carried out in the context of an artificial ecosystem. This ecosystem has an ethological basis that makes it possible to observe and explain the behavior of biological organisms that carries over into the study of actor-critic learning by interacting robotic devices. The contribution of this article is a framework for actor-critic learning defined in the context of approximation spaces and information granulation.**

I. Introduction

The problem considered in this paper is how to guide actor-critic learning [1] based on information granules that reflect knowledge about acceptable behavior patterns. The solution to this problem stems from approximation spaces, which were introduced by Zdzisław Pawlak [12] starting in the early 1980s and which provide a basis for set approximations (see, *e.g.*, [6], [7], [8]). A basic granular computing architecture for actor-critic learning is shown in Fig. 1.

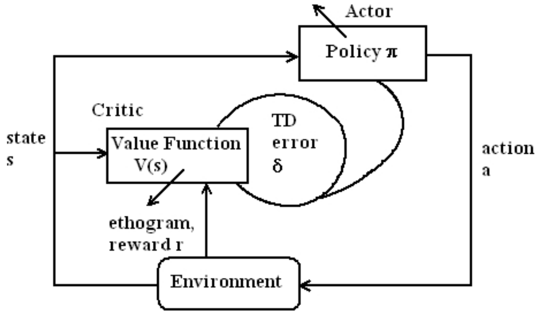


Fig. 1. Basic Structures in Actor-critic Learning

The conventional actor-critic method evaluates whether things have gotten better or worse than expected as a result of an action-selection in the previous state. A temporal difference (TD) error term δ is computed by the critic to evaluate an action previously selected. An estimated action preference in the current state is then determined by an actor using δ. Swarm actions are generated by a policy that is influenced by action preferences. In the study of swarm behaviour of multi-agent systems such as systems of cooperating bots, it is helpful to consider ethological methods (see, *e.g.*, [35]), where each proximate cause (stimulus) usually has more than one possible response. Swarm actions with positive TD error tend to be favored. A second form of actor-critic method is defined in the context of an approximation space (see, *e.g.*, [15], [16], [17], [21], [18], [20], [30], [33]), and which is an extension of recent work with reinforcement comparison (see, *e.g.* [17], [19], [22], [18]). This form of actor-critic method utilizes what is known as a reference reward, which is pattern-based and action-specific. The contribution of this article is a framework for actor-critic learning defined in the context of approximation spaces and information granulation.

This paper is organized as follows. A brief introduction to actor-critic learning is given in Sect. II. The distinction between features and probe functions is given in Sect. III. Approximation spaces are briefly presented in Sect. IV. Information granulation that starts with neighborhoods of objects that indiscernible from each other (*i.e.*, atoms) is briefly introduced in Sect. IV-C. A comparison of two forms of actor-critic learning is given in Sect. V.

II. Actor-Critic Learning

Actor-critic (AC) methods are temporal difference (TD) learning methods with a separate memory structure to represent policy independent of the value function used (see Fig. 1). The AC method considered in this section is an extension of reinforcement comparison in [34]. The following notation is needed (here and in subsequent sections). Let S be a set of possible states, let s denote a (current) state and for each $s \in S$, let $A(s)$ denote the set of actions available in state s. Put $A = \cup_{s \in S} A(s)$, the collection of all possible actions. Let a denote a possible action in the current state; let s' denote the subsequent state after action a (that is, s' is the state in the next time step); let $p(s, a)$ denote an action-preference (for action a in state s); let r denote the reward for an action while in state s.

Begin by fixing a number $\gamma \in (0, 1]$, called a *discount rate*, a number picked that diminishes the estimated value of the next state; in a sense, γ captures the confidence in the expected

1-4244-0703-6/07/$20.00 ©2007 IEEE

value of the next state. Let $C(s)$ denote the number of times the actor has observed state s. As is common (*e.g.*, see [34]), define the estimated value function $V(s)$ to be the average of the rewards received while in state s. This average may be calculated by (1).

$$V(s) = \frac{n-1}{n} V_{n-1}(s) + \frac{1}{n} \cdot r, \tag{1}$$

where $V_{n-1}(s)$ denotes $V(s)$ for the previous occurrence of state s. After each action selection, the critic (represented as δ) evaluates the quality of the selected action using

$$\delta \longleftarrow r + \gamma V(s') - V(s),$$

which is the error (labelled the TD error) between successive estimates of the expected value of a state. If $\delta > 0$ then it can be said that the expected return received from taking action a at time t is larger than the expected return in state s resulting in an increase to action preference $p(s, a)$. Conversely, if $\delta < 0$, the action a produced a return that is worse than expected and $p(s, a)$ is decreased [38].

The preferred action a in state s is calculated using

$$p(s, a) \leftarrow p(s, a) + \beta\delta,$$

where β is the actor's learning rate. The policy $\pi(s, a)$ is employed by an actor to choose actions stochastically using the Gibbs softmax method [2] (see also [34], 30–31)

$$\pi(s, a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}.$$

Algorithm 1 gives the actor-critic method that is an extension of the reinforcement comparison method given in [34]. It is assumed that the behaviour represented by Algorithm 1 is episodic (with length T_m, an abuse of notation used [28] for terminal state, the last state in an episode) and the while loop in the algorithm is executed continually over the entire learning period, not just for a fixed number of episodes.

III. Features and Measurements

It was Zdzisław Pawlak who proposed classifying objects by means of their attributes (features) considered in the context of an approximation space [9]. Explicit in the original work of Pawlak is a distinction between features of objects and knowledge about objects. The knowledge about an object is represented by a measurement associated with each feature of an object. In general, a feature is an invariant property of objects belonging to a class [37]. The distinction between features and corresponding measurements associated with features is usually made in the study of pattern recognition (see, *e.g.*, [3], [5]). In this article, the practice begun by Pawlak [9] is represented in the following way. Let A denote a set of features for objects in a set X. For each $a \in A$, we associate a function f_a that maps X to some set V_{f_a} (range of f_a). The value of $f_a(x)$ is a measurement associated with feature a of an object $x \in X$. The function f_a is called a *probe* [5]. By $Inf_B(x)$, where $B \subseteq A$ and $x \in U$ we denote the *signature* of x, i.e., the set $\{(a, f_a(x)) : a \in B\}$. If $B = \{a_1, \ldots, a_m\}$, then Inf_B is identified with a vector $(f_{a_1}(x), \ldots, f_{a_m}(x))$ of probe function values for features in B.

Algorithm 1: Actor-critic Method

```
Input  : States $s \in S$, Actions $a \in A$, Initialized $\gamma$, $\beta$.
Output: Policy $\pi(s, a)$.
for (all  $s \in S, a \in A(s)$) do
    $p(s, a) \longleftarrow 0$; $\pi(s, a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$; $C(s) \longleftarrow 0$;
end
while True do
    Initialize $s$;
    for ($t = 0; t < T_m; t = t + 1$) do
        Choose $a$ from $s = s_t$ using $\pi(s, a)$;
        Take action $a$, observe $r, s'$;
        $C(s) \longleftarrow C(s) + 1$;
        $V(s) \longleftarrow V(s) + \frac{1}{(s)} [r - V(s)]$;
        $\delta = r + \gamma V(s') - V(s)$;
        $p(s, a) \longleftarrow p(s, a) + \beta\delta$;
        $\pi(s, a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$ ;
        $s \longleftarrow s'$;
    end
end
```

In what follows, the term *feature* is used instead of the term *property* in [32]. It is assumed that variables x, X denote concrete objects. That is, it is understood that an object is something external to us (i.e., something subject to spatial and temporal constraints). An object can either be molecular (i.e., with the structure defined by the *part relation*, *e.g.*, a set with the only parts being its elements (objects)) or atomic (i.e., an object with no parts). An atom is always part of some molecular object. We freely use the terms *set* and *element* interchangeably with *molecular object* and *atom*, respectively. The notation X, Y and x, y denotes also sets and elements of sets, respectively. It is also understood that *classifying* an object by means of its features is not the same thing as *defining* an object.

IV. Approximation spaces

This section briefly presents some fundamental concepts in rough set theory that provide a foundation for a new approach to reinforcement learning by collections of cooperating agents. The rough set approach introduced by Zdzisław Pawlak [10], [11] provides a ground for concluding to what degree a set of equivalent behaviours are *covered* by a set of behaviours representing a standard. The term "coverage" is used relative to the extent that a given set is contained in a standard set. Approximation spaces were introduced by Zdzisław Pawlak during the early 1980s [9], elaborated in [4], [11], [6], [7], [8], and generalized in [30], [33]. The motivation for considering approximation spaces as an aid to reinforcement learning stems from the fact that it becomes possible to derive pattern-based rewards (see, *e.g.*, [22]).

An overview of rough set theory and applications is given in [27]. For computational reasons, a syntactic representation of knowledge in rough set theory is provided in the form of *data tables*.

A. Rough sets

Let U be a non-empty finite set (called a *universe*) and let $\mathcal{P}(U)$ denote the power set of U, *i.e.*, the family of all subsets of U. Elements of U may be, for example, objects, behaviours, or perhaps states. A *feature* $\mathcal{F}$ of elements in U is measured by an associated probe function $f = f_{\mathcal{F}}$ whose range is denoted by $\mathcal{V}_f$, called the *value set* of f; that is, $f : U \to \mathcal{V}_f$. There may be more than one probe function for each feature. For example, a feature of an object may be its weight, and different probe functions for weight are found by different weighing methods; or a feature might be colour, with probe functions measuring, *e.g.*, red, green, blue, hue, intensity, and saturation. The similarity or equivalence of objects can be investigated quantitatively by comparing a sufficient number of object features by means of probes [5]. For present purposes, to each feature there is only one probe function associated and its value set is taken to be a finite set (usually of real numbers). Thus one can identify the set of features with the set of associated probe functions, and hence we use f rather than $f_{\mathcal{F}}$ and call $\mathcal{V}_f = \mathcal{V}_{\mathcal{F}}$ a set of feature values. If F is a finite set of probe functions for features of elements in U, the pair (U, F) is called a *data table*, or *information system* (IS).

For each subset $B \subseteq F$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $\sim_B$ is an equivalence relation, for $B \subset F$ and $x \in U$ let $[x]_B$ denote the equivalence class, or *block*, containing x, that is,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$) then x and x' are said to be *indiscernible* with respect to all feature probe functions in B, or simply, B-*indiscernible*.

Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a B-lower approximation

$$B_*X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

and a B-upper approximation

$$B^*X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

The B-lower approximation B_*X is a collection of blocks of sample elements that can be classified with full certainty as members of X using the knowledge represented by features in B. By contrast, the B-upper approximation B^*X is a collection of blocks of sample elements representing both certain and possibly uncertain knowledge about X. Whenever $B_*X \subsetneq B^*X$, the sample X has been classified imperfectly, and is considered a rough set. In this paper, only B-lower approximations are used.

B. Generalized approximation spaces

This section gives a brief introduction to approximation spaces. The basic model for an approximation space was introduced by Pawlak in 1981 [9], elaborated in [4], [11], [6], generalized in [30], [33], and applied in a number of ways (see, *e.g.*, [15], [16], [24], [17], [18], [19], [22], [23], [25], [31]). An approximation space serves as a formal counterpart of perception or observation [4], and provides a framework for approximate reasoning about vague concepts.

To be precise about what "approximation space" means, some definitions are required. A *neigbourhood function* on a set U is a function $N : U \to \mathcal{P}(U)$ that assigns to each $x \in U$ some subset of U containing x. A particular kind of neigbourhood function on U is determined by any partition $\xi : U = U_1 \cup \cdots \cup U_d$, where for each $x \in U$, the ξ-neigbourhood of x, denoted $N_\xi(x)$, is the U_i that contains x. In terms of equivalence relations in Section IV-A, for some fixed $B \subset F$ and any $x \in U$, $[x]_B = N_B(x)$ naturally defines a neigbourhood function N_B. In effect, the neigbourhood function N_B defines an *indiscernibility* relation, which defines for every object x a set of like-wise defined objects, that is objects whose value sets agree precisely (see, *e.g.*, [20]). An *overlap function* ν on U is any function $\nu : \mathcal{P}(U) \times \mathcal{P}(U) \to [0, 1]$ that reflects the degree of overlap between two subsets of U.

A *generalized approximation space* (GAS) is a triple (U, N, ν), where U is a non-empty set of objects, N is a neigbourhood function on U, and ν is an overlap function on U. In this work, only indiscernibility relations determine N.

A set $X \subseteq U$ is *definable* in a GAS if, and only if X is the union of some values of the neigbourhood function. Specifically, any information system (U, F) and any $B \subseteq F$ naturally defines parameterized approximation spaces $AS_B = (U, N_B, \nu)$, where $N_B = [x]_B$, a B-indiscernibility class in a partition of U.

A standard example (see, *e.g.*, [30]) of an overlap function is *standard rough inclusion*, defined by $\nu_{SRI}(X, Y) = \frac{|X \cap Y|}{|X|}$ for non-empty X. Then $\nu_{SRI}(X, Y)$ measures the portion of X that is included in Y. An analogous notion is used in this work. If $U = U_{beh}$ is a set of behaviours, let $Y \subseteq U$ represent a kind of "standard" for evaluating sets of similar behaviours. For any $X \subset U$, we are interested in how well X "covers" Y, and so we consider another form of overlap function, namely, *standard rough coverage* ν_{SRC}, defined by

$$\nu_{SRC}(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|} & \text{if } Y \neq \emptyset, \\ 1 & \text{if } Y = \emptyset. \end{cases} \tag{2}$$

In other words, $\nu_{SRC}(X, Y)$ returns the fraction of Y that is covered by X. In the case where $X = Y$, $\nu_{\mathrm{SRC}}(X, Y) = 1$. The minimum coverage value $\nu_{\mathrm{SRC}}(X, Y) = 0$ is obtained when $X \cap Y = \emptyset$. One might note that for non-empty sets, $\nu_{SRC}(X, Y) = \nu_{SRI}(Y, X)$

C. Information Granules

Information granulation makes it possible to achieve a form of data compression [7]. A neighborhood of an object

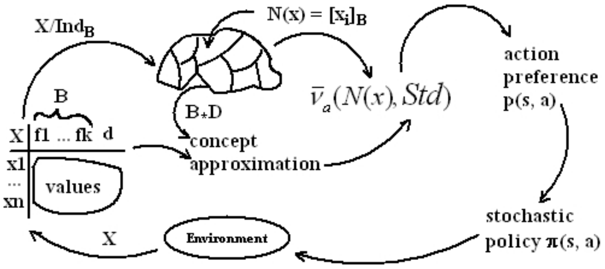


Fig. 2. Basic Structures in Approximation Space-Based Learning

TABLE I
SAMPLE ETHOGRAM

x_i	s	a	$p(s,a)$	r	d
x_0	k	h	0.0	0.75	1
x_1	k	i	0.0	0.75	0
x_2	ℓ	i	0.0	0.1	0
x_3	ℓ	j	0.0	0.1	1
x_4	k	h	0.0	0.75	1
x_5	k	i	0.0	0.75	0
x_6	ℓ	i	0.010	0.9	1
x_7	ℓ	j	0.025	0.9	0
x_8	k	h	0.01	0.75	1
x_9	k	i	0.056	0.75	0

constitutes a set of indiscernible (similar) objects. Neighborhoods form the basic granules (atoms) of knowledge about the universe [7]. Neighborhoods can also be construed as a basis for our perception of objects. An example of a granule at the atomic level is an equivalence class $[x]_B$ (also known as a block or elementary set). Neighborhoods provide the basic building blocks for approximations of particular sets of objects and, in particular, approximations of concepts [4]. The lower approximation B_*D of a concept D is an example of a molecular or *compound information granule* (*i.e.*, composition of elementary granules), which is the set of all objects which can for *certainly* classified as D in view of B [7]. In what follows, the lower approximation provides a standard of behaviour during actor-critic learning. To see this, we first introduce what is known as average rough coverage.

D. *Average rough coverage*

An *ethogram* is a decision system representing observations of the behavior of an agent interacting with an environment, and provides a basis for the construction of a particular approximation space, which can be used to influence action preferences and a stochastic policy during reinforcement learning (see Fig. 2).

This section illustrates how to derive average rough coverage using an ethogram. An *episode* is a sequence of states that terminates. During an episode, an ethogram is constructed, which provides the basis for an approximation space and the derivation of the degree that a block of equivalent behaviours covers a set of behaviours representing a standard (see, *e.g.*, [17], [19], [35]).

Define a *behaviour* to be a tuple $(s, a, p(s, a), r)$ at any one time t, and let d denote a decision (1 = choose action, 0 = reject action) for acceptance of a behaviour. Let $U_{beh} = \{x_0, x_1, x_2, \ldots\}$ denote a set of behaviours. Decisions to accept or reject an action are made by the actor during the learning process; let d denote a decision (0=reject, 1=accept). Often ethograms also include a column for "proximate cause" (see [35]), however in the following example, this is considered as constant, and so such a column is redundant and does not appear.

Let B be the set of probe functions for state, action, action-preference, and reward. The probe functions are suppressed, so identifying probe functions with features, write $B = \{s, a, p(s, a), r\}$. For each possible feature value j of a, (that is, $j \in \mathcal{V}_a$), and $x \in U_{beh}$, put $B_j(x) = [x]_B$ if, and only if, $a(x) = j$, and call $B_j(x)$ an *action block*.

Put $\mathcal{B} = \{B_j(x) : j \in \mathcal{V}_a, x \in U_{beh}\}$, a set of blocks that "represent" actions in a set E of sample behaviours. Setting $\nu = \nu_{SRC}$, define the *average* (lower) *rough coverage*

$$\overline{\nu} = \frac{1}{|\mathcal{B}|} \sum_{B_j(x) \in \mathcal{B}} \nu\left(B_j(x), B_*E\right).$$

Computing the average rough coverage value for action blocks extracted from an ethogram implicitly measures the extent that past actions have been rewarded.

Sample approximation space

What follows is a simple example of how to set up a lower approximation space relative to an ethogram.

Let $S = \{k, \ell\}$ be the collection of two states, and let $A = \{i, j, k\}$ be the set of possible actions, with $A(k) = \{h, i\}$, $A(\ell) = \{i, j\}$.

The calculations are performed on the feature values shown in the first four columns of Table I. Put $B = \{s, a, p(s, a), r\}$. Let $U_{beh} = \{x_0, x_1, \ldots, x_9\}$ and let $D = \{x \in U : d(x) = 1\} = \{x_0, x_3, x_4, x_6, x_8\}$ be the decision class, and consider the following equivalence classes:

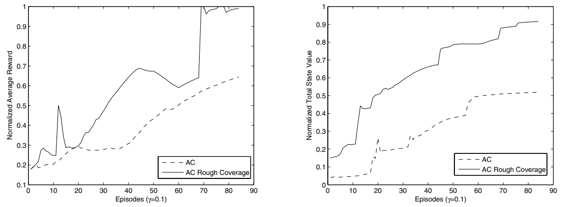
$[x_0]_B = \{x_0, x_4\}$, $[x_1]_B = \{x_1, x_5, x_9\}$,
$[x_2]_B = \{x_2\}$, $[x_3]_B = \{x_3\}$,
$[x_6]_B = \{x_6\}$, $[x_7]_B = \{x_7\}$,
$[x_8]_B = \{x_8\}$.

Then letting $B_*D = \{x_0, x_3, x_4, x_6, x_8\}$ play the role of Y in equation (2) and each equivalence class $[x]_B$ play the role of X, one obtains:

$\nu([x_0]_B, B_*D) = \frac{2}{5}$, $\nu([x_1]_B, B_*D) = 0$,
$\nu([x_2]_B, B_*D) = 0$ $\nu([x_3]_B, B_*D) = \frac{1}{5}$,
$\nu([x_4]_B, B_*D) = 0$ $\nu([x_5]_B, B_*D) = 0$ $\nu([x_6]_B, B_*D) = \frac{1}{5}$,
$\nu([x_7]_B, B_*D) = 0$ $\nu([x_8]_B, B_*D) = \frac{1}{5}$.

Finally,

$\overline{\nu}_h = (\frac{2}{5})\frac{1}{3} = \frac{3}{10}$, $\overline{\nu}_i = (\frac{1}{5} + 0 + 0)\frac{1}{3} = \frac{1}{15}$, and $\overline{\nu}_j = (\frac{1}{5} + 0)\frac{1}{2} = \frac{1}{10}$.



(a) Average Rewards ($\gamma = 0.1$) (b) State Values ($\gamma = 0.1$)

Fig. 3. Actor-critic Method Test Results, $\gamma = 0.1$

E. Actor-Critic Learning and Information Granulation

In this work, each instance of a twiddle (effort to improve its behavior) by an organism leads to granulation of available information, which is stored in an ethogram. The granules derived from an ethogram are in the form of neighborhoods (blocks of equivalent behaviors). The basic idea is to measure the degree of overlap of each neighborhood with the objects in a decision class containing objects that have been judged to be acceptable.

F. Actor-critic methods using rough coverage

This section introduces what is known as a rough-coverage actor critic (RAC) method. The preceding section is just one example of actor-critic methods [34]. In fact, common variations include additional factors that vary the amount of credit assigned to selected actions. This is most commonly seen in calculating the preference, $p(s, a)$. The rough coverage form of the actor-critic method calculates preference values as shown in (3).

$$p(s, a) \leftarrow p(s, a) + \beta \left[\delta - \bar{\nu}_a\right], \tag{3}$$

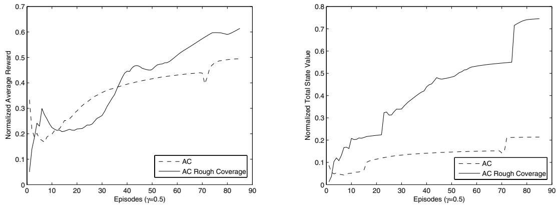
where $\bar{\nu}_a$ (average rough coverage relative to action a-blocks) is reminiscent of the idea of a reference reward used during reinforcement comparison. Recall that incremental reinforcement comparison uses an incremental average of all recently received rewards as suggested in [34]. By contrast, rough coverage reinforcement comparison (RCRC) uses average rough coverage of selected blocks in the lower approximation of a set [19]. Intuitively, this means action probabilities are now governed by the coverage of an action by a set of equivalent actions which represent a standard. Rough coverage values are defined within a lower approximation space. Alg. 2 is the RAC learning algorithm used in the ecosystem for actor-critic methods using lower rough coverage.

V. Conclusion

Test results for two forms of actor-critic learning are given in Figures 3 and 4, which suggests that the RT AC method does better than the AC method in adjusting the action policy to yield favorable results. The details about the design of various testbeds and construction of ethograms that provide a basis for the experimental results reported in this paper, can be found in [14], [17], [18], [19], [25], [24].

Algorithm 2: Rough Coverage Actor Critic Method

```
Input  : States $s \in S$, Actions $a \in A(s)$, Initialized $\gamma$, $\beta$,
         $\bar{\nu}$.
Output: Policy $\pi(s, a)$ //where $\pi(s, a)$ is a policy in
         state $s$ that controls the selection of a particular
         action in state $s$.
for $(all\ s \in S, a \in A(s))$ do
    $p(s, a) \longleftarrow 0$;
    $\pi(s, a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$;
    $Count(s) \longleftarrow 0$;
end
while True do
    Initialize $s$;
    for $(t = 0; t < T_m; t = t + 1)$ do
        Choose $a$ from $s$ using $\pi(s, a)$;
        Take action $a$, observe $r, s'$;
        $C(s) \longleftarrow C(s) + 1$;
        $V(s) \longleftarrow V(s) + \frac{1}{C(s)}\left[r - V(s)\right]$;
        $\delta = r + \gamma V(s') - V(s)$;
        $p(s, a) \leftarrow p(s, a) + \beta\left[\delta - \bar{\nu}\right]$;
        $\pi(s, a) \longleftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$ ;
        $s \longleftarrow s'$;
    end
    Extract ethogram table $IS_{swarm} = (U_{beh},\ A,\ d)$ ;
    Discretize feature values in $IS_{swarm}$ ;
    Compute $\bar{\nu}$ using $IS_{swarm}$;
end
```



(a) Average Rewards ($\gamma = 0.5$) (b) State Values ($\gamma = 0.5$)

Fig. 4. Actor-critic Method Test Results, $\gamma = 0.5$

The concept of selecting the highest reward for an equivalence class comes back to this idea of perception. We are interested in representing perceptions about the behaviour of an ecosystem. In trying to observe patterns in an ethogram, usually it would be difficult to observe meaning from individual elements, so instead we look to classes of them [4]. Furthermore, the way we see the world is reflected in the assembly of neighbourhoods that we perceive and measurements of overlap obtained in some manner. This is facilitated through the use of the neighbourhood function (relative to an equivalence relation defined for objects with specified features), and the overlap function (rough coverage). This method appears to lead to improved ecosystem behaviour because we are drawing on perceptions of the behavior of ecosystem members represented in ethograms, which enable us to induce improvements in learning by ecosystem members.

Future work will include other forms of granulation (*e.g.*, families of neighborhoods) in approximation spaces that provide a basis for actor-critic learning.

ACKNOWLEDGEMENTS

The author gratefully acknowledges comments and suggestions concerning various parts of this paper by the anonymous referees, David Gunderson, Christopher Henry, Dan Lockery, and Andrzej Skowron. The author also extends his thanks to Christopher Henry, who set up the experimental testbeds useds to test each of the algorithms presented in this paper. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and Manitoba Hydro grant T277.

REFERENCES

[1] H.R. Berenji, "A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters," *IEEE Trans. on Fuzzy Systems*, vol. 11, no. 4, pp. 478-485, 2003.

[2] Gibbs, J.W.: *Elementary Principles in Statistical Mechanics*, Dover, NY, 1960.

[3] Mendel, J.M., Fu, K.S. (Eds.): Adaptive, Learning and Pattern Recognition Systems. Theory and Applications. Academic Press, London (1970).

[4] Orłowska, E.: Semantics of Vague Concepts, *Applications of Rough Sets*, Institute for Computer Science, Polish Academy of Sciences, Report 469, March 1982.

[5] Pavel, M.: *Fundamentals of Pattern Recognition, 2nd Edition*, Marcel Dekker, Inc., NY, 1993.

[6] Z. Pawlak, A. Skowron, "Rudiments of rough sets," *Information Sciences*, vol. 177, pp. 3-27, 2006.

[7] Z. Pawlak, A. Skowron, "Rough sets: Some extensions," *Information Sciences*, vol. 177, pp. 28-40, 2006.

[8] Z. Pawlak, A. Skowron, "Rough sets and Boolean reasoning," *Information Sciences*, vol. 177, pp. 41-73, 2006.

[9] Pawlak Z.: *Classification of Objects by Means of Attributes*, Institute for Computer Science, Polish Academy of Sciences, Report 429, March (1981).

[10] Pawlak Z.: *Rough Sets*, Institute for Computer Science, Polish Academy of Sciences, Report 431, March (1981).

[11] Pawlak Z.: "Rough sets," *International J. Comp. Inform. Science*, **11**, pp. 341–356, 1982.

[12] J.F. Peters, A. Skowron, "Zdzisław Pawlak: Life and work," *Transactions on Rough Sets*, vol. V, pp. 1-24, 2006.

[13] Peters, J.F., Borkowski, M., Henry, C., Lockery, D., Gunderson, D.S.: Line-Crawling Bots that Inspect Electric Power Transmission Line Equipment. In: Proc. Third Int. Conference on Autonomous Robots and Agents (ICARA 2006), Palmerston North, New Zealand, 14 Dec. 2006.

[14] Peters, J.F., Henry, C.: "Approximation spaces in off-policy Monte Carlo learning." *Engineering Application of Artificial Intelligence*, 2007, *in press*.

[15] Peters, J.F.: "Approximation space for intelligent system design patterns," *Engineering Applications of Artificial Intelligence*, **17(4)**, pp. 1–8, 2004.

[16] Peters, J.F.: "Approximation spaces for hierarchical intelligent behavioural system models," In: B.D.-Kepliçz, A. Jankowski, A. Skowron, M. Szczuka (Eds.), *Monitoring, Security and Rescue Techniques in Multiagent Systems*, Advances in Soft Computing, Physica-Verlag, Heidelberg, pp. 13–30, 2004.

[17] Peters, J.F.: "Rough ethology: Towards a Biologically-Inspired Study of Collective behaviour in Intelligent Systems with Approximation Spaces." *Transactions on Rough Sets*, **III**, LNCS 3400, pp. 153-174, 2005.

[18] Peters, J.F., Henry, C.: "Reinforcement learning with approximation spaces," *Fundamenta Informaticae* **71** (2-3), pp. 323-349, 2006.

[19] Peters, J.F., Henry, C., Ramanna, S.: "Rough Ethograms: Study of Intelligent System behaviour." In: M.A. Kłopotek, S. Wierzchoń, K. Trojanowski (Eds.), *New Trends in Intelligent Information Processing and Web Mining (IIS05)*, Gdańsk, Poland, pp. 117-126, 2005.

[20] Peters, J.F., Skowron, A., Synak, P., Ramanna, S.: "Rough sets and information granulation," in: Bilgic, T., Baets, D., Kaynak, O. (Eds.), Tenth Int. Fuzzy Systems Assoc. World Congress IFSA, Instanbul, Turkey, *Lecture Notes in Artificial Intelligence* **2715**, Physica-Verlag, Heidelberg, pp. 370–377, 2003.

[21] Peters, J.F., Ramanna, S.: "Measuring acceptance of intelligent system models," In: M. Gh. Negoita et al. (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, *Lecture Notes in Artificial Intelligence*, **3213**, Part I, pp. 764–771, 2004.

[22] Peters, J.F., Henry, C., Ramanna, S.: "Reinforcement learning with pattern-based rewards," *Proc. Fourth Int. IASTED Conf. Computational Intelligence (CI 2005)*, Calgary, Alberta, Canada, pp. 267-272, 2005.

[23] Peters, J.F., Henry, C., Ramanna, S.: "Reinforcement Learning in Swarms that Learn." In: Proc. 2005 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology (IAT 2005), Compiegne Univ. of Technology, France, pp. 400-406, 2005.

[24] Peters, J.F.: "Approximation spaces in off-policy Monte Carlo learning," Plenary paper in T. Burczynski, W. Cholewa, W. Moczulski (Eds.), *Recent Methods in Artificial Intelligence Methods*, AI-METH Series, Gliwice, pp. 139-144, 2005.

[25] Peters, J.F., Lockery, D.,Ramanna, S.: "Monte Carlo off-policy reinforcement learning: A rough set approach," *Proc. Fifth Int. Conf. on Hybrid Intelligent Systems*, Rio de Janeiro, Brazil, pp. 187-192, 2005.

[26] Polkowski, L., Skowron, A. (Eds.): Rough Sets in Knowledge Discovery 2, *Studies in Fuzziness and Soft Computing* **19**, Springer-Verlag,Heidelberg (1998).

[27] Polkowski, L.: *Rough Sets. Mathematical Foundations*, Springer-Verlag,Heidelberg (2002).

[28] Precup, D.: *Temporal abstraction in reinforcement learning*, Ph. D. dissertation, University of Massachusetts Amherst, May 2000.

[29] Schattschneider, D.J.: The Taxicab Group. *American Mathematical Monthly* **91(7)**, pp. 423-428, 1984.

[30] Skowron, A., Stepaniuk, J.: "Generalized approximation spaces," in: Lin, T.Y.,Wildberger, A.M. (Eds.), *Soft Computing, Simulation Councils*, San Diego, pp. 18–21, 1995.

[31] Skowron, A., Swiniarski, R., Synak, P.: "Approximation spaces and information granulation," *Transactions on Rough Sets III*, pp. 175-189, 2005.

[32] Słupecki, J.: "Towards a generalized mereology of Leśniewski." *Studia Logia* VIII, pp. 131-155, 1958.

[33] Stepaniuk, J.: "Approximation spaces, reducts and representatives," in [26], 109–126.

[34] Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*, Cambridge, MA: The MIT Press, 1998.

[35] Tinbergen, N.: "On aims and methods of ethology," *Zeitschrift für Tierpsychologie* **20**, pp. 410–433, 1963.

[36] Watkins, C.J.C.H.: *Learning from Delayed Rewards*, Ph.D. Thesis, supervisor: Richard Young, King's College, University of Cambridge, UK, May, 1989.

[37] Watanabe, S.: *Pattern Recognition: Human and Mechanical*. John Wiley & Sons, Cishester, UK (1985).

[38] Wawrzyński, P.: *Intensive Reinforcement Learning*, Ph.D. dissertation, supervisor: Andrzej Pacut, Institute of Control and Computational Engineering, Warsaw University of Technology, May 2005.