# Fuzzy Behavior Learning for Sony Legged Robots

**Dongbing Gu and Huosheng Hu**
Department of Computer Science
University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
Email:dgu@essex.ac.uk, hhu@essex.ac.uk

## Abstract

This paper presents a learning reactive control scheme for the Sony legged robots to play soccer. We built a behavior-based architecture that consists of a set of reactive behaviors and a deliberative reasoning component to select or co-ordinate the reactive behaviors. The major reactive behavior is for a Sony robot to move towards the ball, guided by on-board sensors. Fuzzy logic is used to encode the robot behavior. The learning of a FLC is conducted in two stages:learning parameters by GA and learnign its structure by Q-learning.
**Keywords:** robot behavior, fuzzy logic controller, reinforcement learning, reactive control.

## 1 Introduction

Behavior-based robotic control is a strategy that decomposes complex robot tasks into a collection of behaviors. These behaviors tightly couple perception and motor responses and work in a co-ordinated manner to complete complex tasks. The behavior decomposition also leads to partitioning underlying physical space into a set of stable regions and decreasing in dimensions of input space for different behaviors. There is an increasing tendency to build up behaviors by using fuzzy logic controller (FLC) which makes robots be reactive and adaptive [1]. In a FLC, uncertainty is represented by fuzzy sets and an action is generated co-operatively by several rules, each one triggering to some degree to produce smooth, reasonable and robust control effect.

Problems in the design of a FLC are parameter setting of membership functions and composition of rules. They are classified into two catalogues: structure identification and parameter identification [18]. The structure identification of a FLC includes the partition of its input space, the selection of antecedent and consequent variables, the determination of the number of IF-THEN rules, and the initial position of membership functions. In contrast, the parameter identification determines parameters of membership functions. Many learning approaches have been proposed to model a FLC, including neural network (NN) based, reinforcement learning (RL) based [1] [2] [3] [6] [7], and genetic algorithm (GA) based [4] [9] [14] [12] [15] [16] [17]. The NN-based FLC can automatically determine or modify its structure and parameters by representing it in a connectionist way. The problem for the NN-based learning is that a large number of data pairs have to be provided to train the neural networks. Both GA-based and RL-based learning are two equivalent learning schemes, which need a scalar response from real world by interacting with it to evaluate action performance [13] [19]. Such a response can be a scalar value that is more easy to collect than the desired input and output data pairs in robot applications and can be any forms without the differentiable limitation.

Many GA-based learning schemes represent a FLC as an individual and the parameters of

membership function as its genes [9] [14]. In [4] and [12], authors had different evolution strategy where an individual represents a rule, not a FLC. Rules in one generation compete with each other in order to be selected. The credit to individuals was assigned according to their contribution and eligibility traces similar to temporal difference learning. In order to reduce the search space, the rules with same antecedents were grouped into a sub-population where the competence occurred [12]. The final FLC was constructed by selecting a rule from each sub-population.

The RL-based learning includes two similar learning schemes: fuzzy Actor-Critic learning (FACL) and fuzzy Q-learning (FQL). FACL is used for parameter learning in [1] [3] [11] [16] and its difficulty is that it needs both an actor network and a critic network to converge. FQL is also viewed as a kind of Q-learning that uses fuzzy logic to generalize the mapping between Q values and sensory-action pairs. In [2] [6] [7], it was used to learn conclusion actions of a FLC.

In this paper, we are investigating the problem of playing soccer by Sony legged robots in an environment where different objects can be recognized by different colors. The aim is to represent interactions between a robot and its environment as a set of local models by constraining specific behaviors to specific situations. Therefore, dimensions of sensory input and action output are deduced and behavioral design can become relative easy. In each local model, individual behavior copes with local uncertainty appeared in the current sensory information and provides real-time control response. In this way, behavior learning can also benefit from the reduced state space. We use FLC to encode the behaviors. The learning of a FLC will be conducted in two stages: learning parameters by GA and leaning structure by Q-leaning.

The paper is organized as follows. Section 2 models sensory information. Section 3 presents a layered representation of action space. The learning strategy is implemented in Section 4. The Experiment and simulation are given in Section 5. Section 6 concludes the paper.

## 2 State space

### 2.1 Sensors

The main sensors embedded in each Sony legged robot include eighteen optical encoders for motion control of eighteen motors, a color CCD camera, an infrared range sensor, and three gyros for posture measurement (roll, pitch, yaw)[5]. The environment for Sony Legged Robot League in RoboCup is a playing field with the dimension of 3m in length and 2m in width. The goals are centered on both ends of the field, with a size of 60cm wide and 30cm high. Six unique colored landmarks are placed around edges of the field. The ball, walls, goals, landmarks and robot labels are painted with eight different colors distributed in the color space so that a robot can easily distinguish them. Object identification is a color-based processing that filters images based on a Color Detection Table(CDT) [10].

### 2.2 Spatial state representation

One way to gather spatial features from an dynamic and uncertain environment to stimulate behaviors is to continuously maintain a local map that includes relative position of objects in the view of robot on the playing field. The local information is also necessary to most of robot behaviors, such as obstacle avoiding, wall following, goal achieving, docking, wandering, etc.
The objects in a local map for soccer playing could be a ball, goals, beacons, ground, team-mates, opponents, and white edges depending on what the robot will see. Spatial state information in a local map for each object $B$ is defined as:

$$B(\theta, d, s, CV)$$

- $\theta$ is an angle reference to robot's orientation. It is calculated by pan angle $p$ and tilt angle $t$ reading from the head's encoders (see figure 1).

- $d$ is a distance between the robot and an object. It is read from infrared range sensor.

- $s$ is object's pixel size in the image. It is calculated by a morphology filter and the run length encoding (RLC) [8].
- $CV$ is a certainty value. It acts as a memory for what is seen previously and will decay with time exponentially.

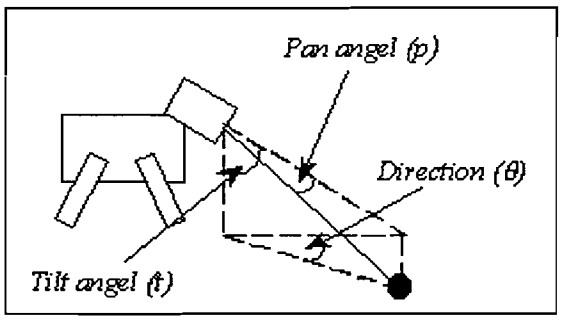


Figure 1: Spatial state space

## 3 Action space

### 3.1 Primitive behaviors

Sony legged robots are quadruped walking robots. Its neck and four legs have three degree of freedoms for looking and walking respectively. A set of primitive behaviors is build up as a basis for the legged robot movement. Figure 2 illustrates eight primitive behaviors. The reactive behaviors used to play soccer are based on this set of primitive behaviors and encoded by FLCs. On top of the action space, a temporal reasoning component is developed to select and co-ordinate the reactive behaviors to drive the robot to achieve the task.

### 3.2 Reactive behaviors for an attacker

To select and design robot behaviors depends upon robot's task and its environment. For a given environment like a paying field described previously, situated activity-based design is reasonable since robot's action can be predicated based on the situation where the robot finds itself. For example, the robot needs to find the ball when the ball disappears from its view. If it finds the ball, the robot needs to approach the ball until it can manipulate the ball. This design method actually endows the behaviors with temporal and

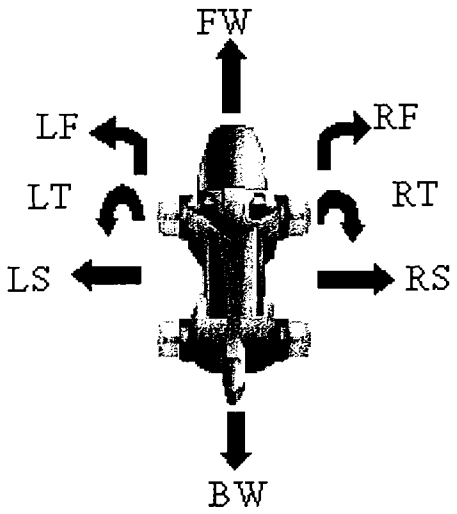


Figure 2: Action space

spatial constrains. The spatial constrains can be represented in the local map on which the robot has to depends. The order for the robot's activities forms behavioral temporal constrains. For an attacker in our team, a set of behaviors is defined as follows:

- *Standing up(SU)*: The robot should try to stand up from the beginning or when it falls down.
- *Looking for ball(LB)*: This behavior will be triggered when the ball is lost from its view for long enough.
- *Approaching ball(AB)*: The robot can move toward the ball when the ball is within its view.
- *Looking for goal(LG)*: This behavior will end when the robot finds the goal or it knows the goal's location.
- *Aligning with goal(AG)*: The robot tries to walk behind the ball in order to let the ball's position lie between the robot and the goal.
- *Kicking ball(KB)*: The robot uses fast walking command to move the ball into the goal.
- *Backward looking(BL)*: This behavior just lets the robot move back for a few steps while

its head looks for the ball since the robot beliefs the ball is in somewhere nearby when it just loses the ball.

### 3.3 Temporal reasoning

Each of the behaviors described above can be regarded as a state in a time series process. The attacker's deliberative reasoning in temporal order for trying to get a score can be modelled as a discrete event system. A discrete event system can be represented by a finite state machine(FSM) by which implementation and verification of an architecture are easy to achieve. There are four elements in designing a FSM:

- *Allowable states*: There are seven states for the attacker described above. They can be labelled as *SU*, *LB*, *AB*, *LG*, *AG*, *KB*, and *BL* respectively.
- *State outputs*: They indicate the corresponding behavioral execution.
- *State inputs*: They are predications based on the spatial information and heuristic rules for an attacker's role.
- *State transition*: It is expressed in figure 3 where the circle represents the state, the arc with arrow represents transition direction, and the symbol near the arc represents state input.

## 4 FLC Learning scheme

### 4.1 Behavior fuzzy encoding

A quadruple of $(a, b, c, d)$ is used to represent the triangle or trapezoid membership function of a fuzzy set. The output action $o$ is the primitive behavior that can be viewed as fuzzy singletons $c^m (m = 1, \ldots, M)$ in a FLC. There are $N$ rules for each behavior. The membership degree of a fuzzy set $F_n^i$ of the $nth$ input state variable $s_n$ in the $ith$ rule is expressed as $\mu_{F_n^i}(s_n)$. The true value of the $ith$ rule activated by an input state vector $S$ is calculated by Mamdani's minimum fuzzy implication:

$$\alpha_i(S) = \min(\mu_{F_1^i}(s_1), \ldots, \mu_{F_N^i}(s_N)) \qquad (1)$$

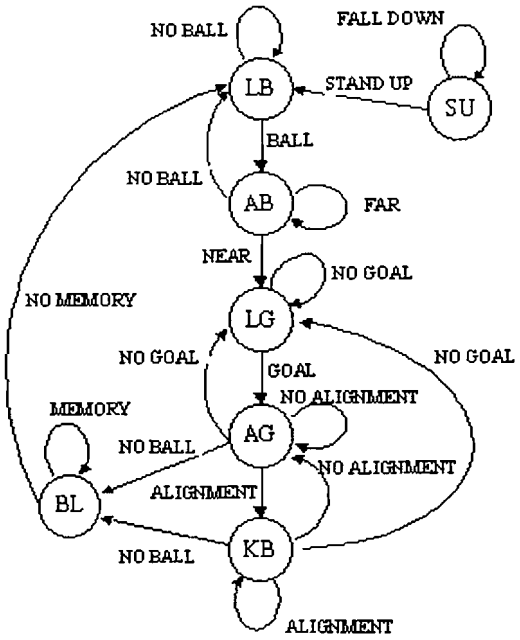


Figure 3: Finite state machine for the temporal reasoning

The crisp output $o$ stimulated by the input state $S$ after the fuzzy reasoning is calculated by the center of area method (COA):

$$o(S) = \sum_{i=1}^{N} \alpha_i(S) \times c^{m_i} / \sum_{i=1}^{N} \alpha_i(S) \qquad (2)$$

### 4.2 Learning Reward

The aim of the *Approaching-ball* behavior is to move towards the ball with an appropriate orientation to the goal. Reward should be designed to guide the evolution of the FLCs to achieve this purpose. The learning payoff is defined as:

$$r = w_1 \cdot \frac{\sum s_i}{L} + w_2 \cdot L + w_3 \cdot \theta_L^{ball} + w_4 \cdot \theta_L^{goal} + w_5 \cdot s_L \qquad (3)$$

where $w_i$ $(i = 1, ..., 5)$ is the weight, $(\theta, s)$ are state variables, $L$ is the number of the walking steps used in one trial. The first term in (3) indicates payoffs received during the whole movement. The second term rewards those FLCs that have fewer steps. The last three terms evaluate the final position of the robot.

### 4.3 Structure learning

In the structure learning, the rule base of a FLC is learned by Q-learning using similar idea proposed in [6][8]. More specifically, the actions in consequences of a FLC are learned while the number of fuzzy rule is fixed. We divide sensory state space into different fuzzy sets and assign initial values for their membership parameters by rule of thumb. Then, the rule base is learned incrementally through backup updating with the initial parameter values. There are several candidates for output action in a rule, each of which is assigned a Q-value that will be learned by Q-Learning. When the learning is finished, the action with the highest Q-value in a rule is selected as consequence for the rule.

Since the sensory space is divided into grids according to the definition of input fuzzy sets, the rules that will be fired in a grid are formed as a local control model. The backup updating learning starts from the grids near to the ball and will continue from them to those far from the ball. In this way, the search space is explored by backup updating instead of randomly exploration like GA learning since the size of grids is relative limited. The learned grids will be exploited, which make the learning works more effectively since the robot becomes increasingly aware of better control rules and capable of interact with its environment via these better control rules as learning progresses.

### 4.4 Parameter learning

In the parameter learning, further fine-tuning of the learned FLC will be carried out. For Sony legged robots, output actions are discrete commands, each of which can make a robot move a single step in different directions. In this research, the parameter learning for input membership functions is investigated by GA learning. A real value GA learning is adopted to optimize membership parameters.

A FLC is encoded as one individual. In each generation, a collection of individuals is maintained to compete with each other for survival. By evolving through genetic operators, the best one will be selected as the optimal FLC for the behavior control. There are three genetic operators employed in this process, namely reproduction, crossover, and mutation. The parameter learning procedure starts from random formation of initial generation of a structured FLC. After completing the trial of one generation, GA evolves into a new generation and the learning repeats again until GA terminal condition is met.

## 5 Experiment results

Experiments for the real robot consists of the primitive behaviors, handcrafted reactive behaviors and temporal reasoning. Figure 4(a) shows the results where an attacker is positioned near the goal initially and the ball is placed at the other side of the field. The robot first goes into the LB state where *Looking-for-ball* behavior is executed. Upon finding the ball, the robot transits into the AB state from the LB state 4(b). Figure 4(c) show how the robot tries to approach the ball in the AB state. Then, the robot starts to look for the goal 4(d) and align itself with the goal until the directions of the ball and the goal fall in the given range 4(e). Finally, the robot kicks the ball into the goal by a fast-walking command 4(f).

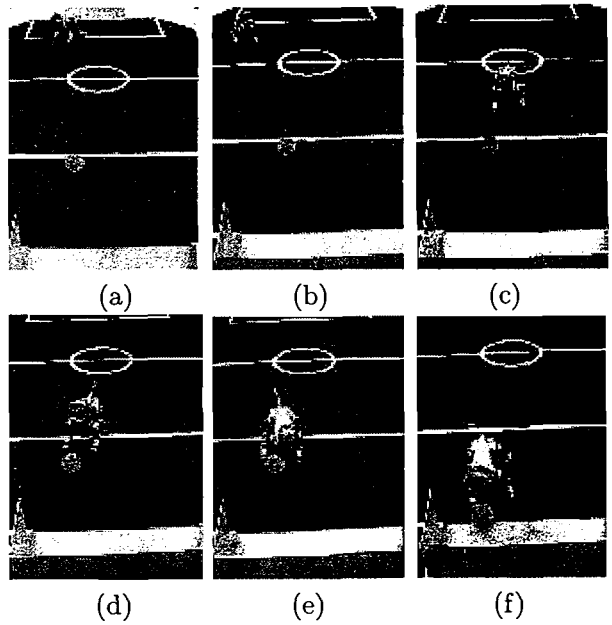


Figure 4: real robot experiment

Fuzzy learning experiments are tested in a simulation environment where the exact size of the filed and the ball is built up. The dynamics of Sony legged robots is identified by measuring

the consequences of actual execution of each walking commands. In order to reflect real-world uncertainty, Gaussian noise is added in both motion dynamics and state variables during the simulation. One step delay is used in the command execution.

The rule base is incrementally build up by the backup updating approach. Figure 5 shows the results for the *Approaching-ball* behaviour controlled by the FLC with the learned rule base. It can be seen that the robot can move to the ball although the robot do not exactly face the goal at final positions.

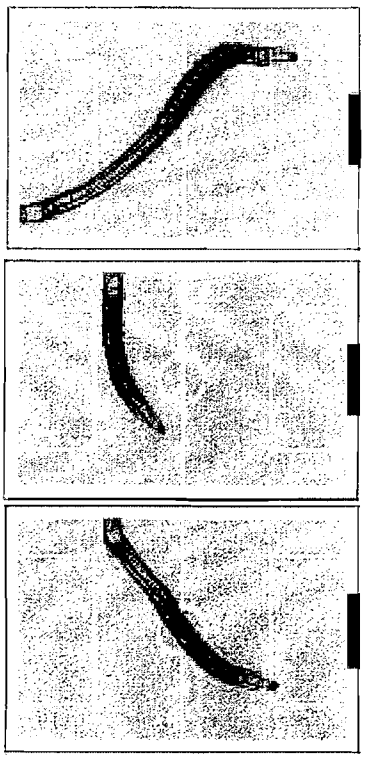

Figure 5: The behavior after structure learning

In parameter learning, the probability of crossover and mutation are chosen both as 0.2. The size of population in one generation is 50. The GA learning process is shown in figure 6 after evolving 300 generations. The upper curve is the maximal fitness values in each generation. The low curve is the average fitness values in each generation. It is shown that the average fitness values converge to the maximum as the generation increases. The quadruple (a,b,c,d) for each input fuzzy label should be constrained by their geometric shapes ($a \leq b \leq c \leq d$) during GA learning. A validation process is employed to check the constraints for each FLC before it is used. Invalid FLCs are given up. Therefore, most of FLCs are not executed due to their invalidation at early evolving stage and the fitness values are unchanged before the 50th generation.

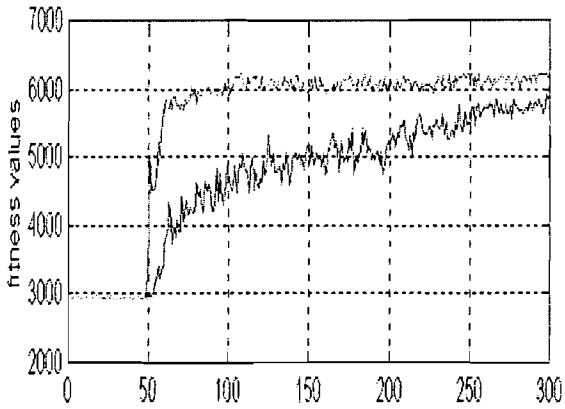


Figure 6: GA learning process

After the parameter learning by GA, the same situations as in figure 5 are tested for the robot with the evolved FLC. The results are shown in figure 7 where the improvement at final positions can be seen.

## 6 Further Work

The next step of our work is to transfer the learned FLC in the simulation into a real robot for further learning in real world. The learned knowledge in simulation will be reused to increase the learning rate. Investigation will also be conducted to the comparison between two-stage learning proposed in this paper (learning parameters and structure independently) and symbiotic learning.

There is no global visual system to provide relative positions among robots, ball and goal. Only information provided from environment is the ball's local sensory information relative to robot's localization.

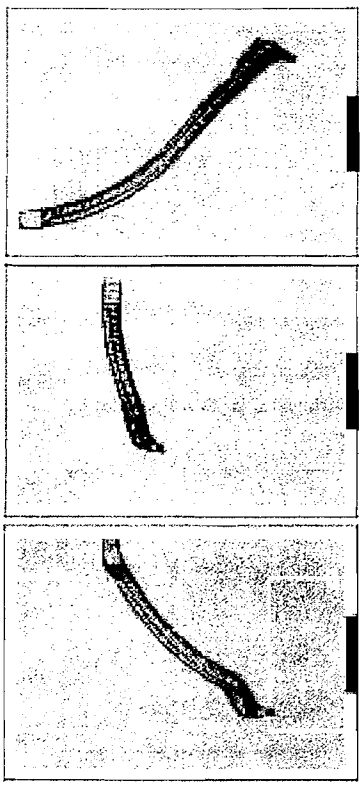

Figure 7: The behavior after parameter learning

## References


[1] Beom, H. R. and Cho, H. S., A Sensor-based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning, IEEE Transactions on SMC, Vol. 25, No. 3, March 1995, pages 464-477.

[2] Berenji, H. R., Fuzzy Q-Learning: A New Approach for Fuzzy Dynamic Programming, Proc. of the 3rd IEEE Int. Conf. on Fuzzy Systems, Orlando, June, 1994.

[3] Berenji, H. R., Khedkar, P. and Malkani, A., Refining Linear Fuzzy Rules by Reinforcement Learning, Proc. of the 5th IEEE Int. Conf. on Fuzzy Systems, 1996, pages 1750-1756.

[4] Bonarini, A., Evolutionary Learning of Fuzzy rules: competition and cooperation, In Pedrycz, W. (Ed.), Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Norwell, MA, 1996, pages 265-284.

[5] Fujita, M. and Kitano, H., Development of an Autonomous Quadruped Robot for Robot Entertainment, Autonomous Robots, Vol. 5, 1998, pages 7-20.

[6] Glorennec, P. Y., Fuzzy Q-Learning and Dynamical Fuzzy Q-Learning, Proc. of the 3rd IEEE Int. Conf. on Fuzzy Systems, Orlando, June, 1994.

[7] Glorennec, P. Y. and Jouffe, L., Fuzzy Q-Learning, Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems, 1997, pages 659-662.

[8] Gu, D. and Hu, H., Learning and Evolving of Sony Legged Robots, Proc. Int. Workshop – Recent Advances in Mobile Robots, Leicester, UK, June 2000, pages 52-59.

[9] Homaifar, A. and McCormick, E., Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms, IEEE Trans. on Fuzzy Systems, Vol. 3, April 1995, pages 129-139.

[10] Hu, H. and Gu, D., Reactive Behaviours and Agent Architecture for Sony Legged Robots to Play Football, International Journal of Industrial Robot, Vol. 28, No. 1, January 2001, pages 45-53.

[11] Jouffe, L., Fuzzy Inference System Learning by Reinforcement Methods, IEEE Transactions On SMC-Part B, Vol. 28, No. 3, August 1998, pages 338-355.

[12] Juang, C. F., Lin, J.Y. and Lin, C. T., Genetic Reinforcement learning through Symbiotic Evolution for Fuzzy Controller Design, IEEE Transactions on SMC-Part B, Vol. 30, No. 2, April 2000, pages 290-301.

[13] Kaelbling, L. P. and Moor, A. W., Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research, Vol. 4, 1996, pages 237-285.

[14] Karr, C. L. and Gentry, E. J., Fuzzy Control of pH Using Genetic Algorithms, IEEE Transactions on Fuzzy Systems, Vol. 3, Jan. 1993, pages 129-139.

[15] Leitch, D. and Probert, P., New Techniques for Genetic Development of a Class of Fuzzy Controllers, IEEE Transactions on SMC-Part C, Vol. 28, No. 1, 1998, pages 112-123.

[16] Lin, C. T. and Jou, C. P., GA-Based Fuzzy Reinforcement Learning for Control of a Magnetic Bearing System, IEEE Transactions on SMC-Part B, Vol. 30, No. 2, April 2000, pages 276-289.

[17] Matellan, V., Fernandez, C. and Molina, J. M., Genetic Learning of Fuzzy Reactive Controllers, Robotics and Autonomous Systems 25, 1998, pages 33-41.

[18] Mitra, S. and Hayashi, Y., Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework, IEEE Transactions on Neural Networks, Vol. 11, No. 3, May 2000, pages 748-768.

[19] Moriarty, D. E., Schultz, A. C. and Grefenstette, J. J., Evolutionary Algorithms for Reinforcement Learning, Journal of Artificial Intelligent Research, 11, 1999, pages 241-276.