Evolving Interactive Autonomous Agents to Understand Contextual Meaning of User's Messages

Mario Antonio Regin Gutierrez
Computer Science and Engineering
University of Nevada
Reno, USA
regin@nevada.unr.edu

Sushil Louis
Evolutionary Computing Systems Lab
Computer Science and Engineering
University of Nevada
Reno, USA
sushil@cse.unr.edu

***Abstract*— This work presents a novel approach to evolve "interactive autonomous agents" that are able to: a) complete the goals of a given task. b) receive messages from a user. c) extract symbolic meaning from these encoded messages. d) plan their next action base on these messages while fulfilling the task's goals. Genetic algorithms, coevolution, and reinforcement learning methodologies are used in combination with an abstraction of the categorical representation presented in natural languages to create a general framework that trains these interactive autonomous agents. The greatest strength of the framework is its ability to evolve agents that learn to recognize symbolic meaning in users' messages and modify their behavior accordingly. Although the evolution of agents' communication mechanism is an active area of research, most the previous work focus on evolving communication mechanism between agents, while this work focuses on evolving user-agent communication mechanisms. This work presents arguments to claim that, currently, genetic algorithms present the only practical way to train interactive autonomous agents because of the capability of genetic algorithms to explore vast search spaces and the ability to coevolve agents and simulated users at the same time.**

Keywords*— *evolutionary computing, genetic algorithms, reinforcement learning, neuroevolution, coevolution, interactive autonomous agents

I. INTRODUCTION

Research on autonomous agents has been a proliferous area in recent years. The availability of multicore processors, reinforcement learning environments, and specialized deep learning libraries have diminished the requirements to experiment with autonomous agents and training new models. However, these autonomous agents are normally trained to complete certain goals, and once they start executing the task, the agents disregard user intentions. The modification of the agent behavior usually requires retraining the agent with a new reward/punishment structure. This paper proposes a new category of autonomous agents, the "interactive autonomous agent." This agent is capable of: a) complete the goals of a given task. b) receive messages from a user. c) extract symbolic meaning from these encoded messages. d) plan their next action base on these messages while fulfilling the task's goals. In other words, the agent is capable of adjusting its behavior on the fly base on the user's intentions. To train such an agent, this work presents a general training framework that evolves agents that are able to extract contextual meaning in users' messages and modify their behavior accordingly to the agent's internal logic, environment's state, and user's intentions.

II. BACKGROUND

A. *Communication Mechanisms*

Human natural language is a symbolic system based on words. As Harnad [4] details, "words originated as the names of perceptual categories of two forms of representation underlying perceptual categorization -- iconic and categorical representations." Iconic representations are the perceptions of the objects themselves. In the human case, this perception is given by our senses. Categorical representation groups similar objects together. These two representations give rise to the third representation form: symbolic. Symbolic representations associate objects with symbols. Harnad [4] mentions that: "The third form of representation made it possible to name and describe our environment, chiefly in terms of categories, their memberships, and their invariant features." The most powerful capability of symbolic representations for this works is that it can be shared across communication systems.

It is possible to communicate without symbolic representation. Such communication mechanics link signals to objects in a direct manner. These communication systems are certainly more limited than symbolic systems but are useful in many scenarios. Cangelosi [1] mentions that regardless of the multiple studios in animal communications, no simple symbolic animal language has been found, and all animal communication is done by signaling. The complex coordination of animals with signaling communication languages is proof of the power of even simple communication systems.

B. *Autonomous Agents*

As defined by Russell [9], an "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators." An autonomous agent is an agent that acts rationally in order to complete a set of tasks in the environment while fulfilling a set of constraints. An agent acts rationally when it performs the best valid action given the information available to the agent.

C. *Logic and Autonomous Agents*

The most resilience method to implement an autonomous agent is using a planning mechanism. If the environment,

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

actions, and instructions of the autonomous agent can be described as a set of logical predicates in the knowledge base KB, then the agent can receive instructions and use a logical planner, such as STRIPS, to complete its task. However, the implementation of this agent would require complete knowledge of the domain, a set of coded logical rules for each possible case, and very low uncertainty in the agent's actions.

D. *Searching and Autonomous Agents*

An autonomous agent decision process can be model as a search problem. If the environment and agent states are discrete, then a search tree can be constructed by mapping valid actions to new discrete states. A search algorithm operates upon the tree to get a set of actions to complete the agent's task. This approach is very computationally expensive since even apparently simple spaces produce impractical large trees. "Go" has a 19x19 board and 361 possible actions; Nevertheless, the game has 3.72×10^{79} valid states. The typical game length of 150 movements would require searching a tree of 4.2×10^{383} nodes. A simple search is not a viable implementation option in almost all scenarios in which an autonomous agent would need to operate.

E. *Deep Reinforcement Learning and Autonomous Agents*

Reinforcement learning is a proliferous area of research in autonomous agents' implementation. In Reinforcement learning, the agent is free to explore the environment. The agent acts upon the environment base on its current policy. At the beginning of training, the policy encourages exploration. The agent improves its policy by interacting with the environment and receiving rewards. Those actions that ended in a reward are encouraged. State of the art methods employ deep neural networks that learn to evaluate the expected reward of each valid action for the current state.

F. *Neuroevolution and Autonomous Agents*

Neuroevolution has been a continuous area of research. The introduction of NEAT by Kenneth [7] permits to evolve the structure of the neural network guiding the agent. Multiple successful additions to NEAT have been made, like the modular behavior of MM-NEAT created by Schrum [5]. NEAT like approaches as all evolutionary approaches require to define an appropriate fitness function to evaluate the agent. In the autonomous agent case, the fitness function is based on the rewards that the environment provides to the agent.

G. *Autonomous Agents and Communication*

Research on evolving agent communication systems seeks to create agents that can coordinate between themselves to collectively perform a task. Cangelosi [1] summarizes various experiments to evolve agents that are able to evolve communication systems based on signals and symbols. The methodology involves agents with fixed topology neural networks that learn to communicate with others in their environment to increase their fitness evaluation. The methodology employs reinforcement learning methodologies to evaluate agents in a virtual environment.

On the other hand, the evolution of Human-Agent communication in a reinforcement learning environment has not seen the proliferation of research that it deserves.

III. Methodology

As mentioned before, an interactive autonomous is capable to: a) complete the goals of a given task. b) receive messages from a user. c) extract symbolic meaning from these encoded messages. d) plan their next action base on these messages while fulfilling the task's goals. messages. The creation of such an interactive autonomous agent would be challenging with current methodologies since they are limited in their ability to incorporate users' messages/meaning/intention while the agent is operating.

a) Logic base agents could incorporate communication with the user by re-planning its action each time a user message is received. However, implementing a complete knowledge base is impractical for all but the simplest domains.

b) Search base agents are impractical given current hardware capabilities since they will have to search the state tree each time a message is received.

c) Reinforcement learning appears as a viable solution, but training deep neural networks through backpropagation requires to properly evaluate the loss function of the output units. The required label samples are impractical to obtain when the agent must not only complete its task but also try to follow the human intention/meaning relayed through the messages it receives.

Evolutionary methodologies offer the best procedure to implement this interactive autonomous agent.

A. *General Procedure*

The presented method presents a general framework to evolve interactive autonomous agents that incorporate user intentions by learning contextual meaning in the message they receive. There are four interfaces that each domain must implement in accordance with its requirements. The environment model, the agent model, the user model, and the meaning function. Once implemented, the general framework uses a genetic algorithm to coevolve two populations. An agent population that seeks to maximize fitness in an environment while incorporating user intentions and a user population that seeks to test the agents. There four main components are:

B. *The Environment Model*

The environment model is the same as in any other reinforcement learning algorithm. The environment assigns rewards and punishments to the agent's actions. Determines when an episode has finished and presents sensorial information to the agent.

C. *The Agent Model*

The agent model receives the sensor state from the environment. The agent can keep track of its internal state, although this part is optional. A user message arrives along with the sensors state. The message can be empty. The agent takes those inputs and executes an action in the environment. The agent must be able to be represented as an abstraction that can mutate and reproduce with another agent. There is no additional constraint in the learning model of the agent. It can use a neuroevolution model, a neural network with fixed topology, a perceptron, a genetic program, or any other learning model that can be used with evolutionary techniques. However, the learning model must be expressive enough to capture the domain solution. Simple models will only work with simple domains.

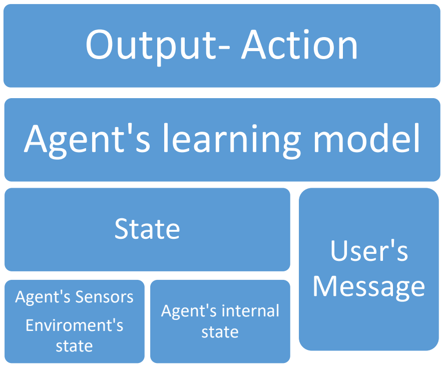

Fig. 1. Agent Model.

D. *The User Model*

The user model must capture a set of actions a user would like to take given the current environment state. The user model rewards the agent each time it follows the intentions contain in the message. These intentions are evaluated by the meaning function. The user model is not a solution but a test. It seeks to test the agent with different messages and see if the agents follow the user's intention. The environment's rewards and punishments guide the agent to complete the task's goals and fulfill the environment's constraints while the user model guides the agent to interpret and follow the user's intentions. In other words, the user model tests the agent's understanding of the user's intentions. The user model receives as input the same information as the agent sensor input the environment state at time t and outputs a message to the agent for time t+1. The user model does not require the internal agent's state since the agent is autonomous.

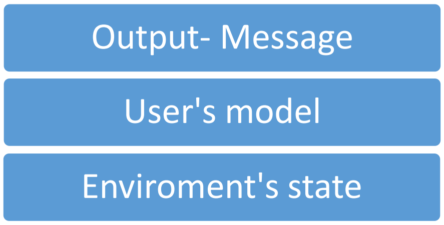

Fig. 2. User Model.

A good way to implement the user model is a probability distribution, especially a joint probability distribution of the environment state and a message. The user messages can be coded in any way as long as they can be represented in the meaning function.

E. *The Meaning Function*

To evolve interactive autonomous agents that recognize symbolic meaning to a user encoded message, the framework uses an abstraction of the categorical representation in natural languages. A categorical representation is a group of similar objects together, in this case, the abstraction represents groups of states that fulfill the meaning encoded in the user message. Consequently, this abstraction manifest as a meaning function that assigns the environment's states to messages. The conjecture is as follows: The message sent by the user is a symbolic representation of its intentions. Each valid state that fulfills the intention of the user is an iconic representation of that intention. The set of all states that fulfill at least one of the intentions of the user is a categorical representation of those intentions. The selective pressure of the user rewards will encourage the reproduction of those agents that evolve to associate sets of states to categorical representations. As the generations continue to increase the agents will learn the symbolic meaning of the messages sent by the users regardless of the encoding used in the message.

Specifically, the meaning function must take the old environment's state and user message and output a set of states that are included in the categorical representation of this message.

Meaning function (old state, message) → *Set of all environment's states in the categorical representations*

In practice, the meaning function will be impossible to compute since it needs to enumerate all the states for each categorical representation. Consequently, a simplified meaning function that accomplishes the same effects must be used. This new meaning function receives the old state, user message and the new state given the agent's selected actions. The simplified meaning function returns true if the new state is part of the categorical representation. This function is easier to implement for a set of countable states and still rewards agents whose actions send them to the state in the correct categorical representation.

Simplified Meaning function (old state, message, new state) → *True if the new state is in the set of all environment's states in the categorical representation*

The state is domain-specific but should usually contain the environment's variables and the agent's external variables like position and rotation.

F. *Coevolution*

The coevolution follows the guidance presented by Rosin, et al [8], "A perfect solution to the testing problem is a set of minimal and extremely hard test cases". Rosin, et al [8], present an infinite population model for coevolution through the use of fitness sharing, shared sampling, and the hall of fame. In this case, such a view is appropriate for the user population, which seeks to represent the space of all possible user's tests to the agent. For the moment, only the hall of fame was implemented in the user's population. The hardest test each generation kept in the hall of fame. N user individuals are selected from the hall of fame each generation.

For the agent's population, a multi-objective fitness is used. Each agent possesses one fitness equal to the rewards gained for its actions in the environment and one fitness gained as compensation for following the user intentions. Agent selection for reproduction is made by following NSGA-II as described by Deb et al [6]. This selection will favor individuals in the Pareto frontier over the two fitness' values.

IV. Experiments

A. *Experiment One: Basic Experiment*

a) Environment: The base experiment uses the classic control problem of balancing a pole in a mobile cart. The problem depends on the weight variables of the objects, the gravitational force and maximum force to apply to the cart. The environment provides the cart velocity and position, and the pole angle and velocity. There are two constraints on the problems, the cart can not go beyond 2.5 meters from the starting positions and the pole angle can not be greater than twelve degrees. The system can be solved by finding the k1, k2, k3, k4 coefficients.

$$F = Fmax * sign(k1*x + k2* \dot{x} + k3*\theta + k4*\theta)$$

A one-layer network is enough to represent this equation. The environment is taken from the popular open ai gym implementation.

b) User population. The user model is a simple one-layer network where weights are in codec binary string format from -100 to 100 with 15 bits per value. A total of 75 bits where used. The mutation operation is binary over the chromosome and reproduction is done with one point cross over. The fitness of each individual is by testing the hamming distance between this individual and the others. Roulette wheel selection is used. The objective, in this case, is to create many different test cases for the agents. The user sends a three-bit message with the meaning of moving left, moving right or staying there. The encoding of the message was randomly assigned for each trail. Mutation, crossover, roulette wheel selection, and coding are implemented as described by Goldberg [3].

c) Agent population. The objective of this experiment is to test the basic concept of this work and assert that this interactive reinforcement learning would not require a vastly more complex learning model. Therefore, the agent model uses a one-layer network with a logarithmic activation function for the output. The model receives 7 inputs, 4 inputs from the environment and a 3-bit message from a user. It outputs 1 variable from -1 to 1, interpreted as the percentage of max force applied to the cart. Weights are encoded binary string format from -100 to 100 with 15 bits per value. A total of 120 bits where used. The mutation operation is binary over the chromosome and reproduction is done with one point cross over. Selection is made using the NSGA-II method to favor individuals in the Pareto frontier. Mutation, crossover, and encoding are implemented as described by Goldberg [3].

d) Simplified meaning function. In this case, the agents must evolve lo learn the meaning of right, left and stay. The simplified meaning function is presented in the following table:

TABLE I. Simplified Meaning function

User's Message	True If
Left	Cart's new x coordinate is less than the cart's x old coordinate
Right	Cart's new x coordinate is more than the cart's x old coordinate
Stay	Cart's new x coordinate is within .01range of cart's x old coordinate

e) Coevolution. Agent's and user's populations where evaluate each generation. Agents are tested multiple times. They act in a randomly initialize environment and receive new messages from a user each delta interval. Each test runs for three minutes. The agent's first fitness gains +1 for each second the agent can balance the pole. The second fitness is given +1 each time the simplified meaning function returns true. Fitnesses are normalized to one for comparison porpuses.

f) Control agent. The normal cart pole problem without user messaging was used as a control experiment. The problem uses the same environment. There is only one fitness, +1 for each second balancing the pole. The problem is solved using NEAT.

B. *Experiment Two: A two-word sentence*

a) Environment: A modified game of tag is used to test is the agents are capable of extracting the meaning of a sentence composed of two words. In this environment, there are four agents. A black agent that represents "it". Three "players" red, green and blue. Each color represents a different agent. "It" must touch the players. If a player is touched by "it", the player is eliminated from that game. "It" is implemented as a simple agent that chase after the nearest player. The red and green players are evolved to avoid "it" using NEAT. The objective of the experiment is to evolve a blue player that avoids "it" but can also receive a user's message consisting of a two-word sentence and a five-word vocabulary: avoid, follow, red, green, nothing. In this case, the blue player can alter its behavior on the fly if the user message the agent to "avoid green" or "follow red" for example.

b) User population. The user model is once again a simple one layer network similar to the one used in experiment one but it receives the different black, red and green agent relative position and distance as input and outputs two three-bit words to form a two words sentence. The first word can be "follow" or "avoid" and the second word can be "red", "green" or "nothing". The encoding of the words was randomly assigned in each trial. Mutation, crossover, roulette wheel selection and coding are implemented as described by Goldberg [3].

c) Agent population. Each agent is an evolved neural network using the NEAT methodology presented by Kenneth [7]. The model receives 8 inputs, 6 inputs from the environment, the directions and distances relatively from the agent to the black, green and red agents, and a two-word message from a user. It generates one output, an angle interpreted as the direction in which to move. Selection is made using the NSGA-II method to favor individuals in the Pareto frontier. Mutation, crossover, and encoding are implemented as described by Goldberg [3].

d) Simplified meaning function. In this case, the agents must evolve lo learn the meaning of the verbs follow and avoid nad how they can be combined to the name of the other agents "red", and "green". It also must learn that nothing signals no particular restrictions. The detailed meaning of “follow” or “avoid” depends on the desired behavior for the domain. In this case, an agent is following another if it is within two meters of the target. Consequently, the agent is avoiding others if it is more than two meters away from the target. The simplified meaning function that satisfies these requirements is presented in the following table:

TABLE II. SIMPLIFIED MEANING FUNCTION

User’s Message	True If
follow green	agent’s new distance from green is less than 2 meters
avoid green	agent’s new distance from green is more than 2 meters
follow red	agent’s new distance from red is less than 2 meters
avoid red	agent’s new distance from red is more than 2 meters
Follow nothing	Always true
Avoid nothing	Always true

e) Coevolution. Agent's and user's populations where evaluate each generation. Agents are tested multiple times. They act in a randomly initialize environment and receive new messages from a user each delta interval. Each test runs for three minutes. The agent's first fitness gains +1 for each second the agent is in the game, meaning it has successfully avoided it. The second fitness is given +1 each time the simplified meaning function returns true. Fitnesses are normalized to one for comparison porpuses.

f) Control agent. A simple agent that avoid “it” was evolved using NEAT. The agent is given +1 fitness each second it can avoid it.

V. RESULTS AND ANALYSIS

The empirical results show the viability of the framework presented in this paper. First, the final agent population was able to assign contextual meaning to the messages presented to them. The agent was also able to evolve to understand the two words sentence structure. Second, computational requirements did not scale out of control. Third, the learning model in the interactive cart pole problem was only as complex as the one needed to solve the no interactive cart pole problem and the best NEAT topology for the interactive blue agent used only one more node than the best no interactive green/red agent, 15 nodes instead of 14.

A summary of the results is shown in table I. Results show that at least for simple problems the method can reliably find an autonomous agent that successfully fulfills the environment’s constraints while extracting message meaning and incorporating the desired user intention.

TABLE III. NEAT VS INTERACTIVE AUTONOMOUS AGENT FRAMEWORK

Method	**Summary**

	Population	*Number of trials*	*Percentage of trials that found an agent that fulfills the task's goals*	*Percentage of trials that found an agent that follows user intention 90% of the time*
Cart Pole No interactive NEAT	150	30	100%	N/A
CartPole Interactive Autonomous Agent	150 AGENTS 10 users	30 using different 3-bits encoding	100%	100%
Tag No interactive Red/green NEAT	150	30	100%	N/A
Tag Interactive Autonomous Agent blue	150 AGENTS 10 USERS	30 using different 3-bits encoding	100%	100%

Figures 3 shows a comparison of the evolution of the NEAT agent in the normal cart pole environment versus the interactive autonomous agent. Fitness was reduced to one value and normalized to one. The x coordinate represents the number of generations. Each generation represents 1,350,000 state evaluations for the NEAT agent and 13,500,000 for the interactive autonomous agent.

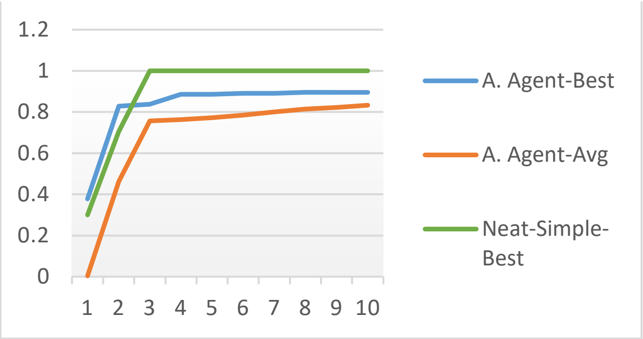

Fig. 3. NEAT and Interactive A. Agent. Average of 30 trials

As shown in the graph, the normal cart pole problem is easily solved with NEAT in 3 generations on average. On the other hand, the interactive agent appears to never reach the theoretical max fitness of one. However, the interactive agent is correctly balancing the pole, but the environment's constraints, cart position and pole angle, disallows it to always follow user intention. This exactly the desired behavior. An interactive agent that understands the meaning in the user’s messages but completes its task while respecting the environment's constraints.

Figures 4 shows a comparison of the evolution of the no interactive NEAT agent in the tag environment versus the interactive autonomous agent. Fitness was reduced to one value and normalized to one. The x coordinate represents the number of generations. Each generation of the interactive autonomous agent requires 10 times more computations since each agent is tested against 8 users in the population and 2 users in the hall of fame.

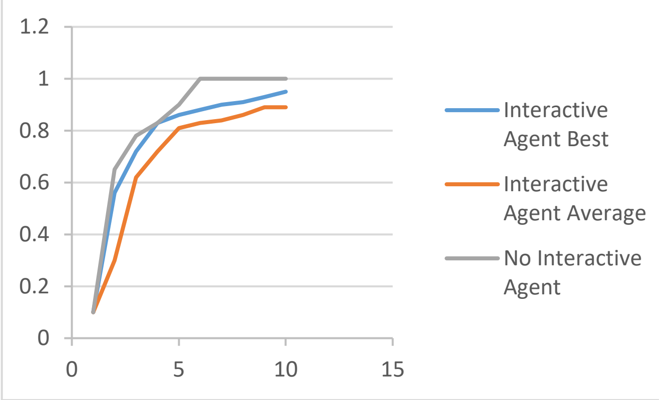

Fig. 4. NEAT and Interactive A. Agent. Average of 30 trials

Results are satisfactory as the interactive agent follows the user intentions most of the time with the exception of cases where "it" is to close and must continually avoid it.

Figure 5 shows the best interactive agent of its generation in the cart pole problem and the relationship between the agent's two objectives. The x-axis shows the normalized fitness, max fitness is 1, the agent is receiving from the environment. Each delta time step in the simulation that the agent is able to keep balancing the pole increments its environmental fitness +1. We expected to always reach values near max fitness in this axis. The y-axis shows the interactive fitness, gaining by understanding the meaning of the user's messages and following such intention. The interactive fitness increases by +1 each time the agent does what the user message conveys. This axis has a theoretical max fitness. Nevertheless, such value will be unattainable because of the constraints placed upon the agent. For example, an agent cannot continue moving forward beyond the 2.4 meters of distance from the center, or the agent may choose to ignore a "move" message because it is busy trying to rebalance the pole after an abrupt movement. Overall figure 4 presents results in line with the expected performance.

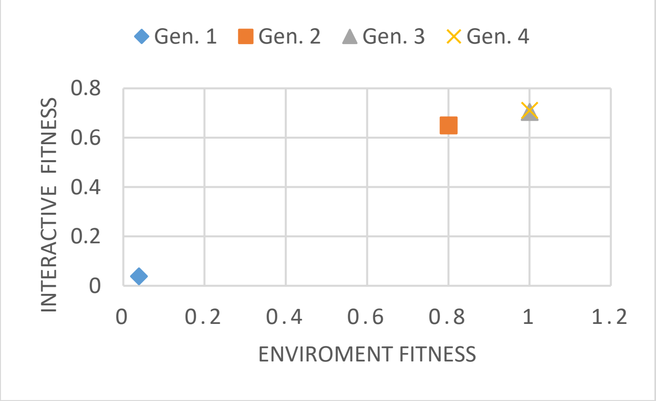

Fig. 5. Interactive autonomous agent best per generation

Figure 6 represents similar results for the interactive agent in the tag problem. The agent is once again able to complete its task, avoid "it" constantly during all the game, while modifying its behavior base on the user message, following or avoiding the other two agents.

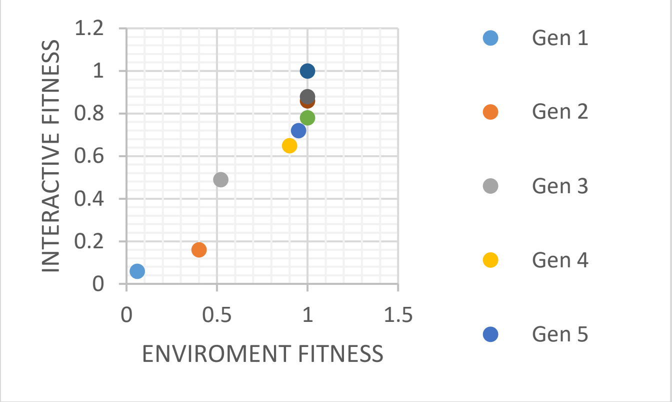

Fig. 6. Interactive autonomous agent best per generation

Directional meaning appears to easily transfer between the categorical and symbolic representations. In this case, the cart pole experiment evolves agents that extract directional meaning of contextual direction, left, right, stay, but absolute directions should not represent any extra difficulty. Furthermore, the tag experiment proves that the framework is capable of evolving agents that understand multi-words sentences related to the environment.

VI. Conclusions

The framework presented in this work demonstrates the ability of evolutionary methods to impart capabilities to autonomous agents beyond those that the current state of the reinforcement methods can. The general usefulness of this framework may be debatable but further research is justifiable because of two reasons. A) This is a novel methodology that may useful to specialized domains, especially autonomous agents that deal with navigation where assigning meaning to a symbolic message that represents direction, ordering, and position is important. B) The training methodology of symbolic meaning presented in this paper, categorical to symbolic meaning, could be generalized to be used in other methodologies.

The initial results show promise for the methodology to work with many different categorical representations like signaling to stay in a user-selected side of the road, avoid and go near objects on demand, explore in a certain order. Although, one can design an environment where one particular task is encouraged, like avoiding some particular object, the methodology is unique in the fact that behavior can be changed on the fly according to user communication, because the agent has learned to associate meaning in its environment to the user messages. This is the main advantage of an interactive autonomous agent.

Future Work

There is work in progress to apply the same framework presented in this work to different contexts. The first one will be to train a driving agent on the popular TORCS environment. The objective of the experiment will be to send a message that corresponded to the user preferred “lane” and “speed” on the road while the agent controls driving.

REFERENCES

[1] Cangelosi A., "Evolution of communication and language using signals, symbols, and words," in IEEE Transactions on Evolutionary Computation, vol. 5, no. 2, pp. 93-101, April 2001.

[2] Deacon T.W. (1997). The Symbolic Species: The coevolution of language and human brain, London: Penguin.

[3] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc..

[4] Harnad S. (1996) The origin of words: A psychophysical hypothesis. In B.M. Velichkovsky & D.M. Rumbaugh (eds), Communicating meaning: The evolution and development of language, Mahwah NJ: LEA Publishers.

[5] Schrum Jacob, and Miikkulainen Risto, (2014). Evolving Multimodal Behavior With Modular Neural Networks in Ms. Pac-Man. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014) (pp. 325–332).

[6] Deb K., Pratap A., Agarwal S., and Meyarivan T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.

[7] Kenneth O. Stanley, & Risto Miikkulainen (2002). Evolving Neural Networks Through Augmenting TopologiesEvolutionary Computation, 10(2), 99-127.

[8] Rosin, C., & Belew, R. (1997). New Methods for Competitive Coevolution*Evolutionary Computation*, *5*(1), 1-29.

[9] Russell, S. J., Norvig, P., & Davis, E. (2010). Artificial intelligence: A modern approach (Third ed.). Upper Saddle River: Prenti