A Biologically-Inspired Platform for the Evolution of Communication in Multi-Agent Systems

Sam McKennoch*, John-Michael McNew, and Linda G. Bushnell
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195-2500 USA
E-Mail: skennoch@u.washington.edu

Abstract - This research effort seeks to address the problem of limited communication in multi-agent systems by the use biologically-inspired implementations of intelligent control systems. A detailed simulation platform is built and tested. The platform contains many user-configurable parameters and is intended to be used as a general purpose research tool for the study of the evolution of communication in multi-agent systems. The user-configurable parameters include those that modify the agent environment, the structure of agent communication abilities, and various simulated biological selection forces and influences including natural selection, cultural transmission, and sexual selection. The simulation involves a predator-prey (pursuer-evader) environment in which evolving predators seek to capture prey and thus increase their fitness. Agents are given the ability to communicate, but are not forced to do so. Simulation results are presented to demonstrate the usefulness and abilities of this tool.

1. Introduction

In recent years, attention has been growing in the area of distributed robotics. Under this paradigm, multiple robots or agents solve a task in parallel either through explicit cooperation, competition or some combination thereof [28]. One of the primary factors limiting the application of such systems is the scalability of communications needed for group action. For an individual robot, communications can represent a significant drain on power and computing. Obviously, as scalability explodes, time, space and power usage will quickly swamp the available resources. Thus efficient communication is an important current topic of research in the application of multi-agent systems to a variety of engineering tasks including distributed space exploration and search and rescue tasks. Another recent research effort from the fields of computer science, and linguistics is the evolution of language [9], [17]. This paper combines ideas from these two areas and describes new research exploring the way that communication and cooperative action among autonomous agents interact. The goal is to evolve task environment specific language whose communications overhead is minimized for a maximized performance measure. The overall objective of this research effort is to demonstrate that in task environments whose constraints can be modeled as evolutionary pressures, a group of autonomous agents may evolve a task-specific communication system and simple language that will increase their overall effectiveness. Moreover in environments that penalize communication efforts such as surveillance tasks or noisy communication networks, a task-specific communication system including only those speech acts found necessary to complete the task will minimize inter-agent communications, yielding significant group problem solving improvement. As a step towards this overall research goal this paper focuses on the development of a software simulation tool, called *CommEvolve*, for investigating these issues using evolutionary algorithms and artificial neural networks.

The authors take the view that all communication takes place within the context of language. Language has two components, structure (or syntax) and vocabulary. The combination of vocabulary and syntax define the expressiveness of a language or the size of the ideas which can be expressed. The authors propose that for a given task environment consisting of a performance measure, environment, and a number of agents, and methods to act upon that environment, there exists a supremum of that performance measure. Given the resource usage of communications, it makes sense to include the expressivity of the communications into the performance measure, since a more expressive communications structure requires more resources which will impinge on other control and decision making methods. Intuitively, once communication overhead is included in the performance measure, there is a trade-off between expressivity and performance. Initially as the expressivity of the language is reduced, the performance measure increases since less communication overhead is required. However, as the size of the language is reduced, the ability of agents to communicate and hence to coordinate is reduced. At some point the trade-off balances. Thus at the maximal performance, there exists a minimized language size. The size and structure of such a language should be such that the required resources generate as little unneeded expressivity as possible. Thus language size becomes another aspect of group control to be optimized. This paper discusses new research in this area. Specifically, this paper presents a simulation tool that utilizes genetic algorithms and neural networks to combine the decision making and language structure problems into a common framework, where language structure can be developed that is specific to the task environment.

More formally, let P_u be the set of pursuers in the unknown environment, *Env*, and E_v be the set of evaders. *L* is the set of all possible speech acts contained within a given syntax and vocabulary size. Let $S(P_u, E_v, Env)$ be the set of all possible sensor states for each robot, where *S* is a function of the pursers and evaders, and environment. Let M_p be the set of all predetermined methods available to the pursuer robot to operate on the environment. The evader methods, M_e are initially unknown and predetermined. Finally let $P(M_p, L, M_e, Env)$ be the performance metric. The metric is a function of the methods used by the pursuers, the methods used by the evaders, the speech acts used by the robots, and the environment.

The current research can then be stated as follows: given a language structure *L*, an initially unknown environment, *Env*, and a set of pursuers and evaders with unknown methods, find a mapping $F: (S,L) \rightarrow L$ and a mapping $G: (S,L) \rightarrow M$ such that $P(M_p, L, M_e, Env)$ is maximized. Future research will take this one step further by also optimizing *L* in order to maximize $P(M_p, L, M_e, Env)$. Thus the problem requires a mechanism for these mappings and a learning mechanism to improve the mappings as the environment and methods of the other agent are sensed.

This paper is organized as follows. Section 2 presents the basics of the *CommEvolve* tool and the predator-prey environment. Section 3 and Section 4 present various aspects of the tool, namely the communication system and the learning methods (cultural transmission and agent evolution). Section 5 presents results from representative simulations. Section 6 contains conclusions and future research directions. References to the state of the art in the literature are given throughout the paper.

2. Simulation Environment

CommEvolve is a simulation tool designed in Matlab (Copyright 1984-2002, The MathWorks, Inc., Version 6.5, Release 13). The tool includes a graphic user interface for ease in use. Given the modularity of the tool, and the community's familiarity with Matlab, the intent is that the tool will be used not only to explore the theoretical aspects of evolved communication, but also to model real world multi-agent systems

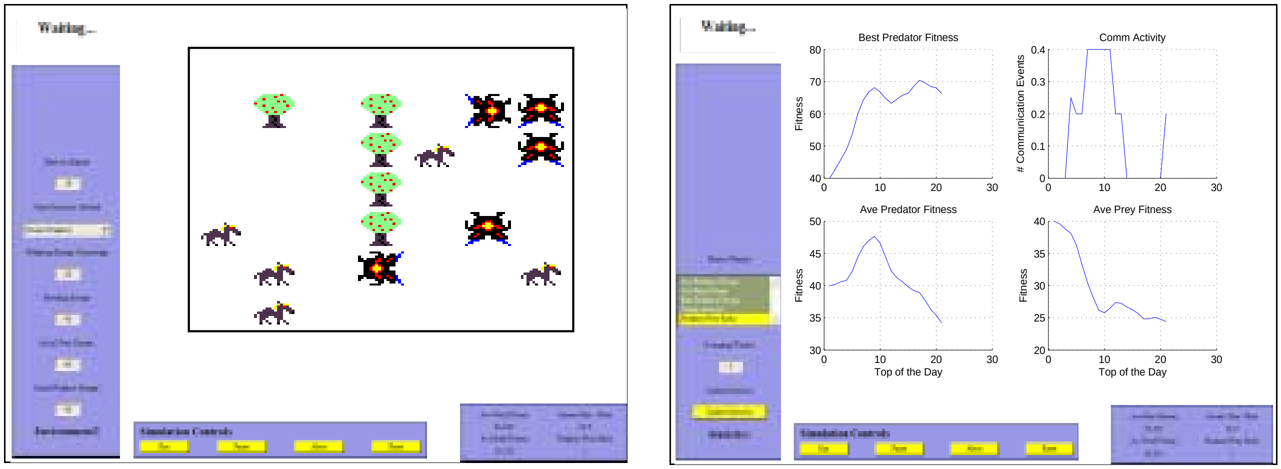
The task environment, *Env*, chosen is a simulated predator- prey domain. This domain is chosen because it is well documented. Also because the complexity of decisions increases exponentially with the number of given choices, in the absence of a good model of the prey, predators are forced to "learn" their likely moves. The prey do not necessarily obey game-theoretic concepts such as a rational computation of rewards. In the face of such illogic, the pursuers must adapt specifically to the methods of the evaders. Another reason this domain is chosen, is that the simulation can also be used to explore ideas of evolved search methods, since search simply requires stationary prey.

The tool is developed modularly with predators, prey, and world as structures that can be modified to represent different engineering problems. What follows is a description of current default parameters. The world structure is a toroidal manhattan grid of variable dimension in which each space can be occupied by a predator agent, a prey agent, or obstacle object. Each agent has a sensor suit, *S* that yields information about the environment. Current defaults include directional sight, hearing, and contact sensors. Each agent also has methods, *M*, corresponding to actuators that effect the environment. Thus, for real engineering problems, the methods involved in different problems can be included to reflect the specifics of the problem. For instance, agents can be given mass parameters that simulate actual robot dynamics. Heterogeneity can also be added in order to give different predators different sensing or actuation abilities. Thus heterogeneous agents may be more likely to have to cooperate and communicate in order increase their collective fitness.

Obstacles are objects that interfere either with the agents sensing, actuation, or communication. They can be used to simulate real environmental effects such as signal scattering, etc. The current default obstacle is a tree which blocks sight sensors and motion methods, but does not affect communication. As discussed by MacLennan, although a task can require cooperation, it only requires communication when agents possess private knowledge of the environment necessary for successful cooperation [17]. The sensor blockage created by the trees creates this private domain knowledge in our simulation and increases the need for agent communication. Figure 1 shows a sample screen shot of a two-dimensional world and resulting statistics from a sample simulation using this world.

The strategy by which prey move can be chosen from a stationary one, to moving away from closest predator, to a number of standard pre-formulated strategies based on predator capture strategies [14]. In order for successful predator evolution to take place it is best that the predator agents neither dominate nor be too weak (otherwise prey extinction or overpopulation will take place). Also, allowing the prey to be stationary at times allows this tool to perform simulations similar to that in [7] for comparison in which agents seek to find mushrooms to eat, but must also chose between mushrooms of different nutritional levels.

Providing a rich environment in which agents may act helps to solve the problem of grounding in a communication system. This problem occurs when agents exchange symbols back and forth in an evolved language. However, in the absence of environment, although agents may under-



(a) (b)

Fig. 1. Sample Screen Shots of *CommEvolve* in Display (a) and Statistics (b) Modes.

(a) Demonstrates the ability to observe the simulation in real-time as agents interact with the environment. (b) Demonstrates the modularity of the tool in allowing for observation of different simulation statistics.

stand each other, the symbols refer only to other symbols and therefore there is no starting point in language through which to establish meaning [16]. Grounding language in the environment also encourages the development of unexpected and original semantics [7].

Simulation time is divided into "hours" and larger units of "days". Every hour a random number of predators and prey are chosen to move simultaneously. This asynchronous method of updating agents eliminates many of the problems associated with synchronous updating [3]. It is therefore possible that an agent may respond in two consecutive hours, or that it may not be activated at all. Conflicts in movement are settled probabilistically via agent fitness level. Fitness for all agents is updated at the end of each hour, and followed by the removal of any agents that may have died. Every day agents are removed from the current environment and exposed to a new one in order to ensure that the agents are not merely training on the initial environmental set-up [14]. Upon the completion of a day, agents with a fitness level over the user-defined breeding threshold are able to be bred if this option is enabled. A mate is chosen based on a user-defined parameter relative to the breeding agent (female choice). New agents are added to the environment without the removal of the parent agents. Each agent's genotype undergoes crossover and mutation to produce a new agent. The parents remain in the environment with the new agent placed spatially near to the mother agent. This arrangement is meant primarily to simulate sexual selection. Sexual selection provides another force of evolution which can work with or against natural selection within limits [12]. Any agent is able to mate when their fitness exceeds a certain level, and likewise any agent can be mated with, thus the population is on some level hermaphroditic. Breeding levies a fitness penalty on the mother.

Subsequently, agent's neural networks are trained, as discussed in Section 4.1, if this option is enabled. Prey also multiply in this environment at a fixed user-defined rate (which can be negative or positive) and without any intelligence. Possible termination criteria for this simulation occur when the prey go extinct or overrun the predators.

The current effort tackles the objective of finding a mapping which combines both decision, sensing, and speech acts for a language of a given structure and expressivity through the development of the *CommEvolve* tool.

The performance metric $P(M_p, L, M_e, Env)$ is referred to as "fitness" in genetic applications and is the measure by which an agent's ability to mate and survive is determined. In terms of engineering applications, the exact form of the fitness function is crucial to the success or failure of the intelligent controls approach to finding a solution. For example, in [25], the author's perform path planning. The fitness is defined as a function of the time it takes an agent to reach its goal and the weight of the cargo it carries with it to the goal.

The exact form of the fitness formula in this effort is user-defined, but in general fitness is a function of agent age, movement, amount of communication performed, feeding success, and mating. The amount of fitness subtracted per hour may be set to be uniform or to increase exponentially to simulate the compounding effects of age. In the pursuer-evader domain conflict ensues when a pursuer agent and either another pursuer agent or an evader agent come in contact. In both cases the losing agent in a conflict loses an amount of fitness related to the participant's fitness levels. As determined in [20] the penalty in the case of pursuer-evader contact should be an order of magnitude greater than other costs per the effects of serious injury tend to have a much greater impact on fitness than other energy expenditures. In order to come in contact, a pursuer need only be next to an evader, while for the case of two pursuers, conflicts occurs when they both attempt to occupy the same space. Also, there is a penalty for acts of communication committed (Section 3).

The mechanism for an increase in agent fitness is feeding. When an agent feeds on trees, fitness increases but at a much slower rate compared to when they feed on moving prey. The predator agents are thus omnivores. When an agent mates, the parent agents incurs a fitness penalty in order to provide the new agent with an initial fitness and still conserve the fitness of the entire environment. In sum then the fitness function may be described as:

$$P_a = P_a - \alpha(t) - \beta \cdot \gamma + \delta \cdot \varepsilon + \zeta \cdot \eta(N_p) - \theta(\vec{P}) - \kappa(\vec{P}, \vec{L}) \quad (1)$$

where P_a is the fitness of agent a; α is a uniform or exponential function of time to account for aging; β is the probability of performing a communication act as determined by the agent's neural network; γ is the communication penalty; δ is probability of being next to a tree; ε is the energy gain for feeding on a tree; ζ is probability of being next to a prey; η is the payoff for a prey-predator conflict and is a function of N_p, the number of predators surrounding the prey; θ is a function to determine the change in fitness when two predators come in conflict by trying to move to same location and is a function of $\vec{P}$, the vector of fitnesses for the entire predator population; κ is a function describing the probability of mating and is a function of the population fitnesses and locations of the current agent and of agents in proximity to the current agent.

Rather than guessing other agent's goals, interactions are based on external, observable, time-varying behavior [19]. A time filter is used as a behavior-based method of measuring survival success which is inclusive of the use of communication. Currently, such a measure that can characterize and judge agent behavior and therefore communication success in a way that is not completely subjective or qualitative does not exist in the literature [13] [26]. Other methods of measuring communication success do not in general apply to this simulation. For example, the denotation matrix used in [8], [11], [17], and [18] is not applicable since it assumes that the communication system is externally known a priori.

3. Communication System

Successful communication results in conferred individual or group advantage via the transfer of information. In purely biologically inspired systems, the actual cost of signalling is negligible [20]. However, in the realm of engineering if the signalling cost was negligible there would be no reason to optimize such a system, which is clearly not the case. This section discusses some background theory in inter-agent communication, and explains how *CommEvolve* relates language, L, and environment, *Env*.

Many research efforts to date have attempted to develop communication from the perspective of meaning to signal mappings, largely because many of these efforts were non-grounded thus eliminating the environment as a consideration. The signal mapping for *CommEvolve* is shown in Figure 2. The meaning to signal mapping exists only internally within the neural network. An explicit mapping between meanings and signals is not necessary because fitness is based on behavior not on a forced pre-defined mapping metric. At least three output communication channels are available as representative of a frequency decomposition of the output communication signal. The actual number of communication channels available will be modulated in different experiments to help determine the number of channels needed to maximize agent fitness. Indeed the number of channels available may be incrementally increased as a simulation proceeds to enable the incremental learning of more complex syntaxes as done in [1]. Communication channel outputs are real-valued as opposed to other efforts that use only binary values [8], [17]. Real-valued communication systems have been shown to make communication systems less sensitive to noise, allow more variability, and produce a closer model to nature [24].

As defined in [27], a conversation is a particular language game, in this case the communication of information from an agent to other agents. A possible series of steps for a language game is as follows [6].

- Establish contact with the another agent
- Identify the communication topic
- Categorize the surrounding world
- Speaker's encoding of the communicative signal
- Listener's decoding of the signal
- Feedback from the listener to the speaker

Feedback in this simulation is allowed but not necessary because learning training sets are developed based on agent observed agent behavior. The arbitrary choice then of choosing to award the emitter or receiver for successful communication alone as in [17], and thus guide communication into developing as a purely cooperative mechanism is not necessary. A series of conversations is called a dialog. Iterated dialogs between agents may help in narrowing down the intended meaning to be transferred.

Agents receive a communicated signal according to their spatial distance to the communicating agent. As this distance increases the strength of the communicated signal is

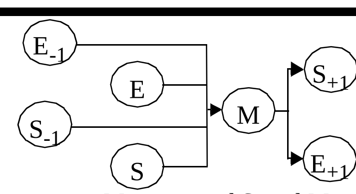


Fig. 2. Environment, Meaning, and Signal Mapping.

Meanings (M) consist of the current, (E), and past (E_{-1}) environment states in addition to current (S) and past (S_{-1}) signals from other agents. Meanings are then communicated and/or acted upon.

scaled downward according as [24]:

$$S_{IN}(a, i) = \sum \frac{\sigma(S_{OUT}(b, i) + U[-u_i, u_i])}{d^2(a, b)} \qquad (2)$$

S_{IN} is the signal that agent *a* receives; S_{OUT} is the signal that agent *b* sends; σ is a bounding function to maintain a useful numerical range, *d* is the euclidean distance between agents *a* and *b*; *U* is a noise generating function that generates a random number between $-u_i$ and u_i. An important test of the communication system once established is to add noise to the environment as represented by *U* in equation (2) and see if the communication system can be maintained, i.e., how robust the communication system is to external interference. σ is a bounding function used to ensure that communicated signals remain within a reasonable range. Signals communicated simultaneously from multiple agents are summed together and last for a single agent activation cycle. *CommEvolve* allows this equation to be modulated in order to test out various communication neighborhood sizes. Communication allows for the transfer of signals to agents other than the nearest neighbors as would be the case in a purely cooperative system, however there is an optimal neighborhood size beyond which sending information to more distance agents adds unneeded complexity.

Communication's effect on a population's fitness can be measured by removing communication abilities and observing the relative metrics compared to when communication is permitted as in [18]. The evolved communication system can also be assessed by providing an externally chosen non-evolving communication language for comparison [6].

Over time it is expected that *CommEvolve* will enable the agent communication language to evolve by memorization, generalization, and invention [4]. An optional feature in *CommEvolve* is to enable a "follow communication gradient" primitive for each communication channel. This primitive has been shown to aid in communication development [24]. In [26], communication systems are defined as unambiguous, partially ambiguous, and fully ambiguous depending the uniqueness of the signal mappings. Unambiguous systems are seen as most efficient as they are the least redundant. CommEvolve allows users to examine typical environmental and signal inputs in respect to their outputs in order to determine the nature of the communication system that might develop using different parameters. In [26] an obverter (basically a neural network run backwards) is used to encourage the development of an unambiguous communication systems. However it is the authors' view that a degree of ambiguity may add to the robustness of the system and so should be explored rather than shunted.

4. Learning Methods

CommEvolve's mapping of sensed environment and communication to outputted communication, F: $\{S, L\} \rightarrow L$, and its mapping to actuation methods, G: $\{S,L\} \rightarrow M$ is achieved via neural networks. Thus all learning methods which improve the performance metric are methods which alter the structure of the neural networks. The two methods used for learning are cultural transmission and genetic evolution. Both are described in this section.

4.1 Cultural Transmission

Cultural transmission models the learning ability of humans to acquire language information that is not innate. Agent aging helps to modulate the amount of cultural transmission. More specifically, agents all have a genotype that is composed of the weights and biases in the neural network that they are born with. Upon birth, an agent's phenotype is identical to its genotype, but soon begins to diverge as the agent learns from its surroundings as in [18]. Because new agents are placed spatially near their mothers, it is expected that a large amount of the initial cultural development will take place as a result of observing the mother agent. Based on the agent's observations of its environment and of adults communicating, a training set is produced that is representative of the observations that resulted in maximum fitness payoff over time to the participants. At the end of every day this training set is used to train the agent's neural network via back-propagation. By generating a training set based on observation rather than participation, the simulation more closely follows the method by which children learn language [16] and which has previously been successfully demonstrated in simulation [7]. When the agent matures and becomes ready to breed, the agent's genotype and not phenotype are used, thus maintaining Darwinian over LaMarkian evolution. As with the other selection forces, culture may be permitted or suppressed. When culture is suppressed, no training of the neural network is allowed.

Regarding how the training set is developed more specifically, a time filter is applied over a number of hours to determine what inputs and outputs are to be put in its training set. The time filter helps to determine the longer-term effects of a move as it may initially cause a decrease in fitness, only to later on cause an overall increase in fitness. The exact nature of this filter is user-defined. Typically, signal processing windows such as modified Hamming windows are used in *CommEvolve* because they provide a good balance of long-term and short-term effects.

Predator neural networks have a user-defined number of layers and nodes per layer. Input and hidden layers use sigmoid squashing functions, while output nodes are also user-defined. Neural networks are a good choice for agent structure because they are not domain dependant and therefore do not constrain or direct in a biased way the direction of the simulation [13]. A number of other efforts have used neural networks to successfully evolve levels of communication as listed in [17], [24], [26]. Other structures such as finite state machines are unable to generalize in the way that neural networks can [17]. The *CommEvolve* tool allows the user to choose between a feed-forward neural network and an Elman neural network. The Elman neural network contains recurrent weights that allow the network to learn behaviors based on a sequence of past and present environment and communication inputs.

Each agent's genotype is made up of an encoding of the weights in this network. The inputs to the neural network include a representation of the agent's sensible environment including communication inputs. Spatial relationships such as front, side, behind, left, straight, and right (in the case of two dimensions) help to define the inputs as was successfully demonstrated in [27]. The outputs of the network include a behavior output as well communication outputs. Also kept track of within each agent are its position, orientation, a record of previous environments and moves.

Following from [3], agents have three primary senses at their disposal: vision, hearing, and touch. Vision is limited to the direction in which the agent is oriented out to some maximum distance. Agents are able to hear other agents in all directions, although the strength of communicated signal drops off with distance. Tactile sensation allows agents to know precisely what occupies the grid spaces immediately adjacent. Unlike homogenous agents, heterogenous agents may have varying sensing and actuation abilities. Thus communication could provide full sensory information and a subsequent sensible movement response.

There is a balance to be had in developing the level of complexity of behavior primitives. Extremely high-level behavior primitives restrict the freedom of the evolutionary process, but provide a set of behaviors that can be used as a building blocks to the overall behavior required. The problem is the overall required behavior is not known a priori in this case, and so care must be taken in designing appropriately sized building blocks. Likewise, low-level behavioral primitives offer the maximum scope for ingenuity, but can result in prohibitively large evolutionary times prior to the emergence of meaningful high-level behavior [2]. This platform includes by default primitives such as orient toward nearest predator, move forward, random move, do nothing. From these behaviors, higher-level behaviors are able to emerge, such as in the case of conflicts retreat, hold-ground, or attack [20]. With minimal knowledge of the underlying code, users may add or delete primitives in order to test various hypotheses.

4.2 Agent Genetic Evolution

CommEvolve uses a genetic algorithm (GA) to simulate biological evolution on agent's neural network weights and biases. GAs have been shown to provide an optimal allocation of trials to substrings, as well as the ability to evaluate an exponential number of string patterns (schemas) with only a linear number of string evaluations [14]. Agent information is encoded into chromosomes. GAs do not scale well to the use of long chromosomes since the search space increases exponentially and so the size of the neural network that the GA acts upon is kept to a minimum [5].

In order to increase the speed of the simulation, the GA chromosomes are not converted to binary. Conversion to binary would be optimal to ensure maximum schema mixing, however this conversion process has been found to reduce the computation speed and algorithm convergence rates in preliminary simulation runs, and so real-values are used in the GA. The primary crossover mechanism acts by swapping the weights between chromosomes according to a user defined crossover rate. Additional methods for crossover include a linear combination of weights from the chromosomes. This crossover strategy works best for genes that can relate input and desired output as a linear combination, and may converge more quickly in task environments with these characteristics. The uniform mutation operator alters weights by adding a uniformly distributed noise parameter 90% of the time, and by replacing the weight altogether the other 10% of the time. Another type of mutation, boundary mutation is generally set to occur less often. Boundary mutation sets the parameter to be mutated to either the maximum or minimum weight or bias of the agents thus exploring the boundaries of the weight and bias range [29].

In line with MacLennan's original denotation of the passage of time [18], the environment of the agents is changed at cycles called days. The periodic resetting of the environment encourages the networks to generalize their response rather than training to a specific environmental layout.

Part of the generality of the tool lies in the fact any selection mechanism such as sexual selection, cultural transmission or natural selection can be disabled to examine the effects of isolated methods on solution convergence. Due to the model complexity it is not expected that an analytical model of the solution mapping can be predicted a priori.

In default, death occurs when an agent's fitness drops below a pre-defined threshold. Thus the population size is not fixed and can adjust itself to the needs of its environment during periods of feast and famine. However, an alternative method of mating and death is included so that the population size remains constant and embodied agents such as teams of mobile robots can be more easily simulated.

5. Simulation Results

Perhaps the most important part of this research is the ability to effectively measure the results in order to make changes to the simulation or to develop engineering principles. For very limited cases equations can de derived, for example the rate of energy loss and rate of change in population size as in [9]. However, in many cases the environment complexity requires analysis of empirical results.

CommEvolve contains metrics that are capable of tracking specific agents over time by tracking training sets, the fitness of their training sets, and the number of offspring among other parameters. As a group, average and best fitness as well as average and best rate of fitness change are tracked [18]. Average training set fitness is also tracked and observed as it changes over time. To provide further robustness, it is possible to periodically test the best individual.

There are a number of forces at operation in the *CommEvolve* simulation tool, each of which can be individually activated to gain a sense of its contribution to the complex web of interactions the various selection forces. These forces include communication, natural selection, sexual selection, and cultural transmission.

A small sample of the initial results of *CommEvolve* follow. These results demonstrate the ease with which various parameters can be changed in order to observe the effects they have on the population over time. The simulations have been conducted over a period of 20 simulation days with 10 hours per day. The number of trees, predators and prey is set such the world space is initially half occupied.

In Figure 3a, the initial energy of the prey has been varied from an amount equal to the predator's energy to an amount 50% greater. When predators defeat prey, they acquire a proportion of the prey's energy. This change in initial prey energy therefore should allow predator energy to rise more quickly than without the change. The population metric reflects a negative or positive growth in population size from day to day. It is calculated according to:

$$\lambda = 1000\log\left(\frac{N}{N_{-1}}\right) \tag{3}$$

where λ is the population metric; N is the current predator population on a specific day; and N_{-1} is the predator population size of the previous day.

By the end of this sample simulation, the population with greater initial prey energy is in a state of expansion as demonstrated by the positive λ value while the other population is contracting, possibly heading toward extinction. A growing population should be limited by the carrying capacity of the environment, i.e., the resources in a fixed environment that will be able to support a relatively constant number of predators. However, it is very easy to indirectly set the carrying capacity to be larger than the number of available grid spaces, thus causing the simulation to prematurely end due to overpopulation. In this simulation the energy at which predators are able to breed is set at the initial predator energy. Breeding, although it increases the population size, incurs a penalty on average fitness because energy for the new predators is derived from the parent predators. Therefore in a growing population, a day of population growth often correlates with lower average predator fitness (not necessarily lower best predator fitness) followed by recovery as the new predator and its parent's energy increases back toward the population average. In general, this simulation demonstrates the ability to quickly produce representative results and test hypotheses through altering environmental parameters.

In Figure 3b, one population is simulated with parameters identical to that in Figure 3a, while the other population has identical environmental parameters, but has genetic and cultural selection disabled. Because breeding is interpreted as part of genetic selection and is therefore disabled, there is no mechanism for the population to increase in size, however, there is also no breeding penalty as happens when a new predator agent takes its initial energy from its parents. The population without selection enabled shows a much greater amount of stability as expected. It appears then that this population has an environment sufficiently rich enough in energy that it can survive, but not sufficient enough to allow reproduction. In a rich enough environment, it would be expected that a longer simulation would show the population with selection enabled eventually exceeding the population without selection as it becomes better adapted to its environment. Longer simulations are also necessary to demonstrate the advantage of communication.

6. Conclusion

This paper has presented a tool for studying the evolution

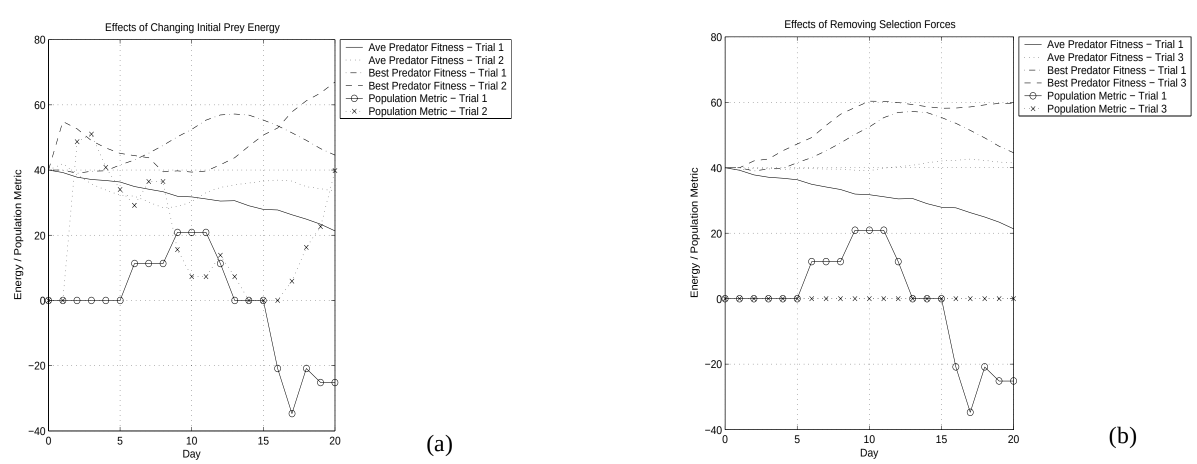


Fig. 3. Simulation Results Demonstrating the Effects of Changing the Initial Prey Energy (a) and Disabling the Selection Forces (b). In (a) Trial 2 has 50% greater initial prey energy than Trial 1. After 20 days the Trial 2 population exhibits a 45% greater average predator energy and a 59% greater best predator energy. In (b) after 20 days the population with selection disabled has a 50% greater initial energy, exhibits a 33% greater average predator energy and a 91% greater best predator energy.

of communication in populations of competing agents. The tool includes a large number of user-defined parameters in order to allow a broad amount of research to be performed. Initial results successfully demonstrate the effects of changing environmental and selection parameters.

Future work includes porting *CommEvolve* to C++ in order to increase simulation speed. Also now that the tool has been developed it will be used to study how a network of agents best communicates in more detail. Other selection mechanisms such as dominance and kin selection will also be added. A dominance hierarchy would help in coordinating behavior and in resolving conflict between agents. Leaders may emerge. A dominance hierarchy also helps to provide the structure through which a new leader may emerge if the current leader is eliminated. A kin-selection system that creates a system of alliances based on the degree of relatedness of agents helps in the evolution of communication and behavior that is more altruistic than would be expected if every agent acted in a way that increased only its chance of survival.

Acknowledgements

The authors would like to thank Professor Emeritus Juris Vagners at the University of Washington for providing important background on this topic. This material is based upon work supported by the National Science Foundation under Grant No. ECS-0137422. J. M. McNew is also supported by a NDSEG Fellowship, sponsered by the Department of Defense, and an ARCS Fellowship.

References

[1] Allexandre, C. and Popescu-Belis, A., “Emergence of grammatical conventions in an agent population using a simplified tree adjoining grammar,” in *Proceedings of the International Conference on Multi Agent Systems*, 1998, pp. 383 –384.

[2] Ashiru, I.; Czarnecki, C.A., “Experiments in evolving communicating controllers for teams of robots,” in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, vol. 4, 1998, pp. 3498 –3503.

[3] Baray, C., “Evolving cooperation via communication in homogeneous multi-agent systems,” in *Proceedings of Intelligent Information Systems*, *IIS ’97*, 1997, pp. 204-208.

[4] Brighton, H., “Compositional syntax from cultural transmission,” *Artificial Life*, vol. 8, no. 1, 2002, pp. 25-54.

[5] Brown, A. D. and Card, H. C., “Evolutionary artificial neural networks,” in *Engineering Innovation: Voyage of Discovery. IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, vol. 1, 1997, pp. 313-317.

[6] Cangelosi, A. and Parisi, D., “The emergence of a language in an evolving population of neural networks,” *Connection Science*, vol. 10, no. 2, pp. 83-97, 1998.

[7] Cangelosi, A., “Evolution of communication and language using signals, symbols, and words,” *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 2, April 2001, pp. 93 –101.

[8] Crumpton, J. J., “Evolution of two symbol signals by simulated organisms,” M.S. thesis, University of Tennessee, Knoxville, December 1994.

[9] Di Paolo, E. A., “An investigation into the evolution of communication,” *Adaptive Behavior*, vol. 6, no. 2, 1997, pp. 285 – 324.

[10] Di Paolo, E. A., “Social coordination and spatial organization: Steps towards the evolution of communication,” in *Fourth European Conference on Artificial Life*, Phil Husbands and Inman Harvey, Eds., MIT Press, 1997.

[11] Enee, G. and Escazut, C., “A minimal model of communication for multi-agent systems,” in *Proceedings of the 2001 8th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2001, pp. 11 –16.

[12] Freeman, S. and Herron, J. C., *Evolutionary Analysis 2nd Edition*, Prentice Hall, 2001.

[13] Ficici, S. G. and Pollack, J. B. “Coevolving communicative behavior in a linear pursuer-evader game,” in *Proceedings of the Fifth International Conference of the Society for Adaptive Behavior,* Pfeifer, Blumberg, Kobayashi, Eds. Cambridge: MIT Press, 1998.

[14] Haynes, T., “Evolving multiagent coordination strategies with genetic programming,” University of Tulsa, *Tech. Rep. UTULSA-MCS-95-04*, 1995.

[15] Hemelrijk, C. K., “Effects of cohesiveness on inter-sexual dominance relationships and spatial structure among group living virtual entities,” *Fifth European Conference on Artificial Life*, 1999.

[16] Kirby, S., “Natural language from artificial life,” *Artificial Life*, vol. 8, no. 2, 2002, pp. 185-215.

[17] MacLennan, B. J., “The emergence of communication through synthetic evolution,” Dept. of Computer Science, University of Tennessee, Knoxville, Tech. Rep. UT-CS-99-431, October 20, 1999. To appear in *Advances in Evolutionary Synthesis of Neural Systems*, edited by Vasant Honavar, Mukesh Patel, and Karthik Balakrishnan (MIT Press).

[18] MacLennan, B., “Evolution of communication in a population of simple machines,” Department of Computer Science, University of Tennessee, Knoxville, Tech. Rep. CS-90-99, January 1990.

[19] Mataric, M. J., “Kin recognition, similarity, and group behavior,” in *Proceedings of the Fifteenth Annual Cognitive Science Society Conference*, Boulder, Colorado, June 1993, pp. 705-710.

[20] Noble, J., “Talk is cheap: evolved strategies for communication and action in asymmetrical animal contests,” in *Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, 2000.

[21] Picault, S. and Collinot, A., “Designing social cognition models for multi-agent systems through simulating primate societies,” in *Proceedings of the International Conference on Multi Agent Systems*, 1998, pp. 238 –245.

[22] Prince, C. G., Mislivec, E. J., Kosolapov, O.V., and Lykken, T.R., “Towards a theory grounded theory of language,” in *Proceedings of the 2nd International Conference on Development and Learning*, 2002, pp. 119 –124.

[23] Riolo, R. L. “The effects of tag-mediated selection of partners in evolving populations playing the iterated prisoner's dilemma,” Santa Fe Institute, Working Paper 97-02-016, November, 1996.

[24] Saunders, G. M. and Pollack, J. B., “The evolution of communication schemes of continuous channels,” Ohio State University LAIR, Tech. Rep. 94-GS-EVCOMM, 1994.

[25] Shibata, T., Fukuda, T., and Tanie, K., “Fuzzy critic for intelligent planning by genetic algorithm,” *Emerging Technologies and Factory Automation, 1993. Design and Operations of Intelligent Factories. Workshop Proceedings. IEEE 2nd International Workshop on*, 27-29 Sep 1993, pp. 78 -85.

[26] Smith, K., “Natural selection and cultural selection in the evolution of communication,” *Adaptive Behavior*, 2002.

[27] Steels, L., “A self organizing spatial vocabulary,” *Artificial Life*, vol. 3, no. 2, 1999.

[28] Stone, P. and Veloso, M., “Multiagent systems: a survey from a machine learning perspective.”, *Autonomous Robots*, vol. 8, no. 3, July 2000.

[29] Zhao, Q., Krogh, B. H., and Hubbard, P., “Automatic generation of test inputs for embedded control systems,” *IEEE Control System Magazine*, 2003.