Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

INVESTIGATING THE DYNAMICS OF COMPETITION: COEVOLVING RED AND BLUE SIMULATION PARAMETERS

Mary L. McDonald

George Mason University
4400 University Dr
Fairfax, VA 22030, U.S.A.

Stephen C. Upton

Referentia Systems Inc.
11435 Newington Ave
Spring Hill, FL 34609, U.S.A.

ABSTRACT

In this paper we explore the concept of two-sided competitive coevolution as a mechanism to explore the dynamics of competition in a simulation context. One potential value of doing so is the ability to rapidly explore simultaneous adaptations to two sides, presumably Blue and Red, in order to find solutions that perform well and are relatively robust even in the face of an adaptive adversary.

1 WHY COEVOLUTION?

Analyses using combat simulations typically involve perturbing a base case in order to perform sensitivity analyses on a set of measures of effectiveness. Frequently, the opposing side's input parameters, which may represent physical or behavioral characteristics, are held constant. Although this process can yield interesting insight into the scenario at hand, it lacks the ability to characterize what can happen if Red and Blue parameters are changed in response to each other, i.e., as a result of direct competition with each other. The topic of coevolution is presented here as it pertains to evolving red and blue force parameters in direct response to each other, allowing the simulation of a competitive "arms race" for continued improvement, and reaching a more robust solution. In some part, this may be construed as an abstraction of simulating an adaptive enemy.

Coevolution is a term originally from the field of evolutionary biology that describes the phenomenon of two species evolving over time, in direct response to each other. Coevolution can be either cooperative or competitive. This paper addresses competitive coevolution. In competitive bi-partite coevolution, the fitness of an individual in a predator population is based on direct competition with individual(s) from a prey population. (Rosin and Belew 1995). Notable is the fact that both populations assume the role of predator and prey simultaneously. As stated in (Ficici and Pollack 1998), "the key to coevolutionary learning is a competitive arms race between opposed participants". Also, "coevolution has been proposed as a way to evolve a learner and learning environment simultaneously such that progress arises naturally with minimal inductive bias. Each participant (candidate solution) in a coevolutionary system is both a learner as well as an environment against which other participants learn".

Many believe that a competitive learning process encourages an evolutionary development such that as one learner develops new strategies, its opponent adjusts its abilities and discovers new strategies of its own (Angeline and Pollack 1993). There have been several studies that have suggested that coevolutionary learning can increase the efficacy of the search for the global optimal solution (if one exists), and can also result in finding a less globally optimal but more robust solution, as compared to more conventional search techniques or one-sided evolutionary learning (Blair et al. 1999, Juille and Pollack 1998, Paige and Mitchell 2001).

2 SOFTWARE AND ALGORITHM DETAILS

2.1 Coevolutionary Software and Algorithm

For a general overview of the field of evolutionary computation including the basics of the standard genetic algorithm and coevolutionary algorithms, see (Mitchell 2000) and (Hillis 1991). For example, the issues of parent selection, recombination, and mutation were treated in a "standard" way, the word standard being in quotes because there actually is much exploration ongoing with regard to evolutionary algorithms. Basically though, an evolutionary algorithm is a biologically-inspired approach to finding a solution to a problem whose search space is large or whose characteristics are unknown. Inspired by the Darwinian principle of natural selection, possible solutions compete for survival. Generally speaking, the stronger solutions remain to reproduce and give birth to new ones, while the weaker solutions are discarded and replaced by new (random) members. This iterative process of replacement of solutions with new ones typically leads to better solutions being discovered.

A software package called NALEX (for Natural Algorithm Experimenter), developed in Java was used to conduct the coevolution. NALEX is a software framework designed to accommodate many different natural algorithms. Natural algorithms are a set of heuristic search techniques that have been motivated by analogies from the natural world. In the physical world, the annealing process of solids has motivated the development of a class of algorithms called simulated annealing. Biological analogies have inspired the class of evolutionary and co-evolutionary algorithms, immune algorithms, and bacterial algorithms. And the social world has inspired the development of particle swarm, ant, and cultural algorithms. Also, realizing that there is no a priori reason for limiting the algorithms to constraints imposed by the natural world, algorithm developers have begun extending and combining these algorithms to form new algorithms with the goal of enhancing performance. NALEX is designed to accommodate these extensions as well.

The components of NALEX easily allow researchers to experiment with a number of algorithms, but additionally, by providing the necessary structure indicative of a framework, to develop their own versions of various structures by extending the components included in NALEX. These user-developed components can then be shared with other researchers, facilitating the development of reusable component repositories that could then be used as a common computational foundation for behavioral experiments

There are many variants of coevolutionary algorithms. Some interesting ones can be found in (Ficcici and Pollack 1998) and (Rosen and Belew 1995). The following are some key characteristics of the coevolutionary algorithm used in this effort.

- Four parameters were chosen for Blue to evolve over (n = 4) and four parameters for chosen for Red to vary over (m = 4). In general, m need not equal n. One "typically" will evolve over a larger number of parameters, to take advantage of the power of the evolutionary algorithm. However, a total of four parameters per side were chosen to ensure that the emergent behavior resulting from these parameter settings is analytically tractable.
- A population (for either Red or Blue) will consist of 30 individuals, also referred to as chromosomes in the evolutionary computation literature.
- The concept of iterations, or turns, is utilized in the algorithm. One side (either Red or Blue) will go first and find a best response to the current set of adversaries. Then the other side will evolve against the new set of its adversaries, and so on. An iteration consists of 30 generations of evolution. The end result of an iteration is that a "best-so far" will be stored. In addition a "worst so far" will also be stored, for comparative analysis purposes.
- "Best-so-far" is determined to be the individual with the best average fitness against the best performing individual of the opposing population each across 30 random restarts.
- The fitness measure used was the Force Exchange Ratio.

It is beyond the scope of this paper to discuss the potential variations to the above, and the pros and cons of doing so. However, there is much to be gained by altering the "dynamics of competition". As merely one example of varying the dynamics discussed in this section, we could choose to evaluate an individual based on a subset of the current adversary population which includes both good and poor performers, or that includes some randomly selected individuals, instead of choosing to evaluate against the best adversary as we did in this effort. There is some discussion in the literature that suggests that such a strategy may make the best-so-far more robust. Some studies have suggested that there may be a tradeoff between search efficacy and maximum evolved fitness of individuals (Paige and Mitchell 2001).

2.2 Simulation Software and Scenario

The simulation used for this experiment was Map Aware Non Uniform Automata (MANA) an agent-based model of combat developed by the New Zealand Defense Technology Agency. This work was part of a larger study performed for the U.S. Marine Corps to examine the effectiveness of aggregation/dispersion behaviors, distance between units, and unit size (squad vs platoon) in an experimental distributed operations scenario. Many of the details of the scenario will not be fully described here in the interest of keeping this paper focused on the details of the coevolutionary work.

3 EXPERIMENTAL DESIGN OF THE COEVOLUTIONARY RUNS

A brief description of the Blue simulation parameters that were evolved over is as follows:

- Cluster: the size of the group that Blue should try to stay in regardless of situational circumstances. One agent represented a fire team, so a size of 3 can be interpreted as squad-sized.
- Inorganic Situational Awareness (SA): Toward Friends: the weighting that governs how much Blue wants to move towards or away from other friends (can be interpreted as a cohesion or dispersion parameter), that it knows about because

the information was passed to the squad from an external source. Negative weights imply moving away from and positive weights imply moving towards.

- Inorganic SA: Towards Enemy Threat 3: the weighting that governs how much Blue wants to move towards or away from enemies with the highest perceived threat level that it knows about because the information was passed to the squad from an external source.
- Max Distance to Inorganic Friends: the distance Blue should try to keep from other (non squad) friends. One pixel represented 250 meters.

A brief description of the Red simulation parameters that were evolved over is as follows:

- Cluster: the size of the group that Red should try to stay in regardless of situational circumstances.
- Squad SA: Toward Friends: the weighting that governs how much Red wants to move towards or away from other non-squad friends (can be interpreted as a cohesion or dispersion parameter), that it knows about from internal sensing capabilities.
- Squad SA: Towards Enemy Threat 3: the weighting that governs how much Red wants to move towards or away from enemies with the highest perceived threat level that it knows about from internal sensing capabilities.
- Max Distance to Organic Friends: the distance Red should try to keep from friends.

The description of the Blue and Red simulation parameters are quite similar. In fact, in this case, the only major difference between the group of Blue parameters evolved over and the group of Red parameters evolved over, is that Blue was acting on information it received "inorganically", in other words, information that was passed to it from other squads or entities over the network.

4 RESULTS

The numeric results of the coevolution are presented in Tables 1 through 4. The parameters corresponding to the highest fitnesses are highlighted displayed in bold.

Table 1: Blue Variable Settings Corresponding to the Highest Fitness, Per Turn

BLUE	cluster	friends	enemy	max dist
turn 1	14	-42	22	134
turn 2	13	-38	19	137
turn 3	15	-42	20	136
turn 4	19	-41	25	130
turn 5	**12**	**-59**	**37**	**116**
turn 6	18	-42	21	135

Table 2: Blue Variable Settings Corresponding to the Lowest fitness, Per Turn

BLUE	cluster	friends	enemy	max dist
turn 1	9	-8	67	264
turn 2	6	47	-16	251
turn 3	7	-49	-55	5
turn 4	9	-35	-27	379
turn 5	5	83	-93	244
turn 6	8	-29	-27	379

Table 3: Red Variable Settings Corresponding to the Highest fitness, Per Turn

RED	cluster	friends	enemy	max dist
turn 1	5	11	-87	112
turn 2	4	8	-81	22
turn 3	**9**	**8**	**-81**	**16**
turn 4	11	3	-82	24
turn 5	7	7	-81	24
turn 6	12	18	-83	284

Table 4: Red Variable Settings Corresponding to the Lowest Fitness, Per Turn

RED	cluster	friends	enemy	max dist
turn 1	10	2	6	299
turn 2	19	45	-89	334
turn 3	19	20	69	297
turn 4	9	48	43	347
turn 5	7	1	5	298
turn 6	19	45	-17	252

It was further observed that, in this round of coevolution, the highest and lowest fitnesses for each turn of evolution did not significantly vary from one turn to the next. One interpretation of this that the best solutions found for each side were relatively robust to random variation.

Another reason for the lack of intense competition is that these eight parameters, when explored over these ranges, only account for limited variability in the success of the Blue or Red Forces. If we were to repeat the coevolutionary experiment, we may choose to evolve over a greater number of parameters per side, to allow for the possibility of even greater variation in the fitness measure. But, for this first test of the coevolutionary runs, we chose to evolve over a smaller number of parameters to start, and the variation seems suitable for our purposes.

Next, we examine how the evolution progressed within each turn. A common way to view this progress is to plot the fitness of the best-so-far by generation. That plot, for the first turn of Blue evolution is contained in Figure 1. The figures for the other turns were very similar so they are not displayed separately here.

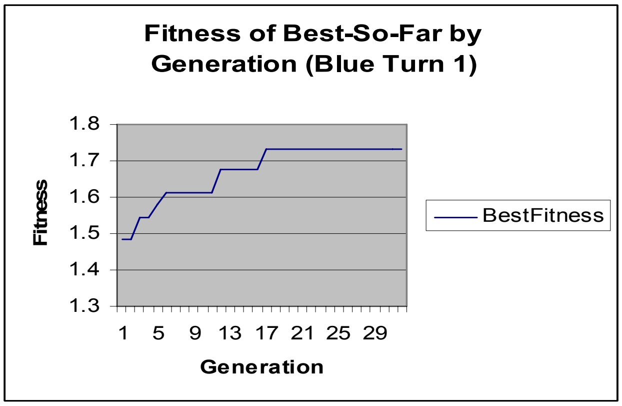

Figure 1: A Look at Evolutionary Progress for Blue Turn 1

In order to further examine how well the coevolutionary algorithm performed, we should assess how much of the "possibility space" was explored. As a first step towards that end, Figure 2 displays the two histograms for Blue killed and Red killed, respectively. We can conclude that there was sufficient variation in the fitness measure.

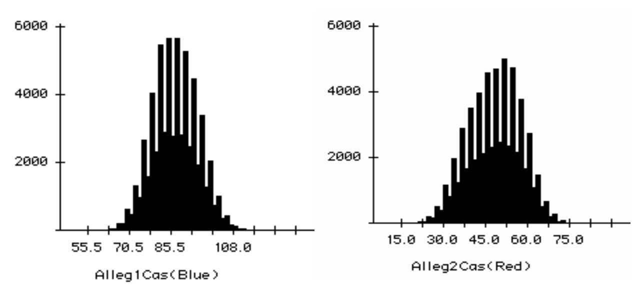

Figure 2: Histograms for Blue and Red Casualties

We now examine to what extent each of the Blue parameters was explored by the algorithm. Figure 3 depicts the histograms for each of the Blue parameters evolved over.

We would not expect to see uniform sampling in Figure 3, since the evolutionary algorithm attempts to balance exploration and exploitation. However, since we note that there is less variation in the inorg en3 and inorg friends parameters, we now take a closer look at the projection onto the *inorg en3 – inorg friends* space. In Figure 4 we depict a colored surface plot where the z axis (and the use of color) represents the fitness metric for Blue.

From Figure 4 we see that the space looks pretty well sampled, with a high peak corresponding to a propensity to stay close to other friendly squads while also pursuing enemy squads.

Turning now to the analysis of results within the context of the overall study that was being performed, there were some interesting findings in this data. It is noted though that all findings discussed below resulted from the exploratory analysis of an agent-based model, and additional means for seeking insight into this scenario is recommended.

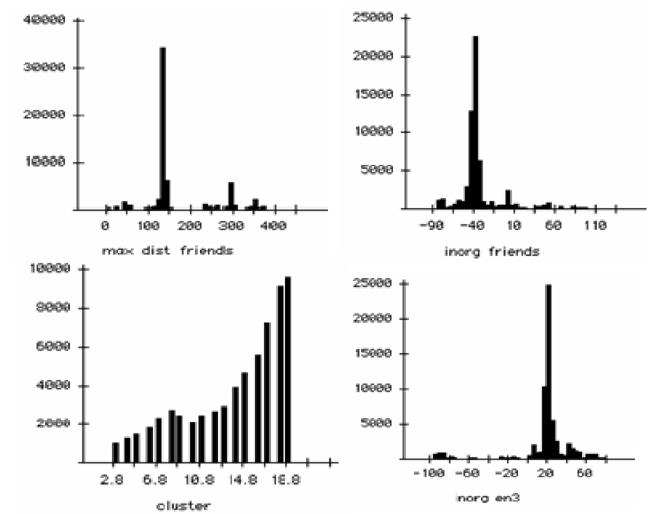

Figure 3: Histograms for Blue Parameters Evolved Over

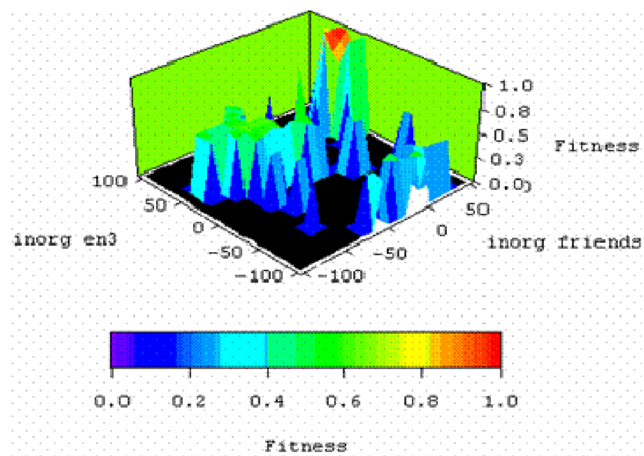

Figure 4: Examination of Two Dimensions of the Blue Space Explored with Corresponding Fitness Measure

In this set of experiments though, the Blue cluster parameter that yielded the best results was 12. In this effort one agent represented a fire team, so 12 fire teams represents a group that is just over one conventional platoon in size. This suggested that it may be true that a Distributed Operations (DO) Unit the size of a platoon is more survivable, in general, then a DO Unit the size of a squad.

In addition, examination of the Inorganic: Friends weight suggested that, under a scenario assumption that DO Units were not specifically trying to aggregate, a DO Unit of the proper size, should generally maintain distance from other DO Units. The maximum distance was suggested by the evolution of the Max Distance parameter, which was approximately 30 km for the best performing Blue. In other words, Blue performed best when they were separated by no more than 30 km on (in this scenario) 150 km by 150 km piece of medium complexity terrain. In this case, they would have been able to aggregate had they wanted to in about one hour, given trafficable terrain that was trafficable by the vehicles represented in this scenario.

Further, examination of the Inorganic: Enemy Threat weight suggested that Blue did best when pursuing (not avoiding) the conventional Red threat.

An analysis of the evolution of Red parameters suggested that they do best when they disperse, to a point. Specifically, Red did best when clustered in groups of about a platoon size, separated by at most 4 km away from each other, a much smaller distance than the distance evolved for Blue. Also, Red should avoid being in close proximity to other Red, and with strong propensity, should avoid contact with Blue if at all possible. This may all suggest that dispersion and avoidance is the best Red could do against a Blue DO-type attack. This is plausible in that the more dispersed and on the move Red is, the harder it will be for Blue to hit all of it with inorganic assets.

5 CONCLUSION AND FUTURE WORK

This effort represented a first attempt to use a coevolutionary algorithm as a wrapper around simulation runs in order to better explore the space of possibilities. As mentioned previously, there are many variants to the coevolutionary algorithm, some of which may perform better in different circumstances. We intend to continue this exploration in pursuit of more effective algorithms, and in pursuit of algorithms which perform well at discovering robust, multipeaked Blue solution sets.

ACKNOWLEDGMENTS

The specific examples presented here constituted part of an effort performed for the Marine Corps Combat Development Command Studies & Analysis Division and the Marine Corps Warfighting Lab (MCWL), and we collectively acknowledge their support they gave to us during this effort. We also thank Mr. John Wasser of MCWL who gave us public release for this work so we could discuss it in this forum. Finally, we would like to acknowledge David Ang for his assistance in conducting the coevolutionary runs on the high performance cluster .

REFERENCES

Angeline, P., and J. Pollack, 1993. *Learning with a competitive population.* The Ohio State University, Columbus, Ohio.

A. Blair, A., E. Sklar, and P. Funes, 1999. *Co-evolution, determinism, and robustness*. Springer-Verlag, Berlin, Germany.

Cioppa, T. and L. Brown, 2004. Agent based simulation efforts. Institute for Operations Research and the Management Sciences (INFORMS) Conference.

Ficici, S. G. and J. B. Pollack, 1998. *Challenges in coevolutionary learning: arms-race dynamics, open-endedness, and mediocre stable states.* The MIT Press, Cambridge, Massachusetts.

Hillis, D., 2001. Co-evolving parasites improves simulated evolution as an optimization procedure. *Artificial Life II*, Addison-Wesley, Menlo Park, California.

Juille, H. and J. Pollack, 1998. *Coevolutionary learning: a case study.* Morgan Kaufmann Publishers, San Francisco, California.

Lucas, T., S. Sanchez, L. Brown, and W. Vineyard, 2002. Better designs for high dimensional exploration of distillations. *Maneuver Warfare Science 2002*, edited by G. Horne and S. Johnson.

McDonald, M. and S. Upton, 2003. Coevolving Red and Blue: analyzing the dynamics of competition. 71st Military Operations Research Symposium, Working Group 31.

Mitchell, M., 1997. *An introduction to genetic algorithms*. MIT Press, Cambridge, Massachusetts.

Paredis, J. 2002. *Coevolutionary algorithms.* Institute of Physics Publishing, Temple Back, Bristol, United Kingdom.

Pagie, L. and M. Mitchell, 2001. *A comparison of evolutionary and coevolutionary search*. San Francisco, California.

Rosin, C. D. and R. K. Belew, 1995. *Methods for competitive co-evolution: Finding opponents worth beating.* Morgan Kaufmann, San Francisco, California.

AUTHOR BIOGRAPHIES

MARY L. MCDONALD is a doctoral student in the School of Information Technology and Engineering at George Mason University and is a member of the Senior Staff at Systems Planning & Analysis, Inc. She has a B.A in Mathematics from Northwestern University and an M.S. in Applied Mathematics from the Naval Postgraduate School. Her research interests include evolutionary computation and artificial intelligence. She is a member of INFORMS Military Applications Society, the Military Operations Research Society, and the American Association for Artificial Intelligence. She previously spent 11 years on active duty with the U.S. Marine Corps. Her e-mail address is mmcdonald@spa.com.

STEPHEN C. UPTON is a Senior Systems Engineer with Referentia Systems, Inc. He holds advanced degrees in Physics, Operations Research, and Systems Engineering. His research interests include artificial immune systems, evolutionary computation and agent-based simulation and their application in studying complex adaptive defense systems. He is a member of INFORMS Military Applications Society, the Military Operations Research Society, and the Association for Computing Machinery. He previously spent 24 years on active duty with the U.S. Marine Corps. His e-mail address is supton@referentia.com.