A Coevolutionary Approach to Course of Action Generation and Visualization in Multi-sided Conflicts*

Liana Suantak
University of Arizona
Tucson, Arizona
liana@ece.arizona.edu

David Hillis
U.S. Army Research Laboratory
Adelphi, Maryland
hillis@arl.army.mil

Jerry Schlabach
Battle Command Battle Lab
Ft. Huachuca, Arizona

Jerzy W. Rozenblit
University of Arizona
Tucson, Arizona
jr@ece.arizona.edu

Michael Barnes
U.S. Army Research Laboratory
Ft. Huachuca, Arizona
michael.barnes@hua.army.mil

Abstract – *The current state of military operations includes many stability and support (SASO), multi-sided conflicts. The research presented in this paper attempts to address this complex environment by creating a SASO simulation, coevolutionary generation of courses-of-actions (COAs) for each side, and visualization tools for analysis of the resulting COAs. The SASO simulation is significantly different from previous systems because it incorporates non-conventional warfare units such as terrorists and media. The coevolution algorithm is different because it allows all sides of the conflict to evolve their COAs. The visualization tools are important because SASO doctrine is not as well developed as conventional warfare doctrine. Therefore, visual analysis and understanding of a system that is not well defined provides insight for future modeling and verification.*

Keywords: visualization, genetic algorithms, course-of-action, stability and support operations.

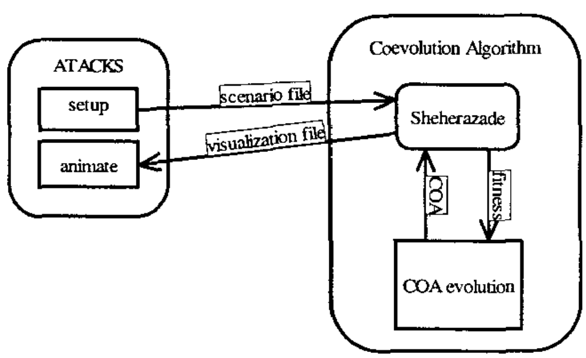
1 Introduction

An increasing proportion of military operations today requires both stability and support functions. The military [2] defines the purpose of stability operations "to promote and sustain regional and global stability" and the primary role of stability operations is "to meet the immediate needs of designated groups, for a limited time, until civil authorities can accomplish these tasks without military assistance." Other major functions may include keeping armed conflicts contained and quieting domestic disturbances.

Many applications currently exist to support commanders' decision-making in conventional warfare scenarios. FOX, created by Schlabach, Hayes, and Goldberg [5] notably used genetic algorithms to create conventional warfare courses-of-action (COA). One of the major drawbacks of FOX was that it only allowed evolution of the blue side against several static COAs of the red (enemy) side. Hillis and Winkler allowed FOX to play each side against itself in [3], creating COAs that *coevolved*.

This paper introduces a system that provides multi-sided evolution (coevolution) for stability and support operations (SASO). Visualization is important to the development of this system because the rules inherent in SASO are currently not as well defined as conventional warfare, allowing insight into our model to aid in development and validation. Section 2 provides an overview of the system. Sections 3, 4, 5, and 6 describe the functional phases of running the system, which consist of setup, coevolution, scenario simulation for fitness evaluation, and analysis of results, respectively. Section 7 concludes and discusses future work.

2 Overview



We have developed a four-part approach: setting up the SASO scenario, the coevolution of courses-of-action

(COAs) by several agents, the SASO simulation named Sheherazade, and the analysis of the results with evolution and scenario visualizations. The first part requires significant user input to define a scenario. The second part (coevolution) uses the third part (simulation) to evaluate solutions. The third part scores solutions by running simulations of entity movements and targets, which are, in turn used by the coevolution to further evolve. Finally, the fourth part allows the scenario designers and analysts to investigate the coevolution and simulation results. These parts are related as shown in Figure 1.

3 Setup phase

In the first phase, a military expert defines simulation parameters to model entities, factions and locales. The entities could be terrorists, refugees, media, non-government organizations, and peacekeeping forces. Each entity has relative combat, intelligence, and other strengths. Each entity belongs to an allegiance, or faction, and each faction has a starting animosity or friendliness to every other faction. The locales contain percentages of the population of each faction, and each locale has a starting attitude, or calmness. The military expert defines all the values for the entities, factions and locales to start, modeling a situation he or she wants to investigate by having the system generate courses-of-actions for each of the entities.

After establishing the environment, the military expert also assigns each entity to an agent. A simple assignment consists of each agent's entities belonging to one faction. The assignment of entities to agents is important in defining the fitness functions, or goals of each agent. For example, if all entities of agent 3 belong to the faction named "Eastern Alliance," then the fitness function of that agent could be to cause local unrest. Another fitness function could be to inflict as much damage as possible on another faction, or it could be a weighted combination of several of these factors.

4 Coevolution

Once the scenario parameters have been defined, the system can begin evolving the agents through a genetic algorithm. The genetic algorithm allows each agent to take its turn evolving. As each agent changes its COA, each of the other agents tries to find a better set of movements and target factions for its own entities to reach a better fitness score.

For many of entities, the chromosomes for the genetic algorithm consist of a list of scheduled movements and targets, if the entity type engages in combat. For the organized military entities that make up the blue or friendly faction, the chromosome is a set of assignments of units to military subordinate commands, which are, in turn assigned to locales.

The genetic algorithm takes these chromosomes and passes them to the SASO simulation, which returns with several scores that make up the fitness function. The genetic algorithm uses the fitness function to select, crossover and mutate new courses of action, which are again played, scored, and selected. The agents take their respective turns evolving against the other agents' COAs, changing their strategies in reaction to the other agents' changes.

5 Sheherazade

As mentioned previously, the coevolution algorithm uses the SASO simulation, named Sheherazade after the famous character of "1001 Arabian Nights," [1] as values for its fitness function. The basic concepts of entities, factions and locales were introduced in Section 3. A more detailed description of the environment, entities, events, and explanation of respective COAs in the simulation algorithm follows. A more thorough description appears in [6].

5.1 Environment

The basic environment consists of factions and locales. Factions are deliberately vague groupings of people assumed to be nominally united by common affiliations and that tend to be thought of as a group. A faction's "animosity" toward another faction represents its "feelings" for that faction. Every provocation (incident, violation, terrorist hit, military attack, etc.) causes the victim's faction to increase its animosity level towards the perpetrator's faction.

Locales are geographical regions, which can be thought of as neighborhoods within a city, or states/ provinces within a nation. Locales have several properties including geographic size, population size, and which other locales are neighbors. The local population of a locale is also divided among the factions. An important property that affects many events in the simulation is the locale's "attitude," which influences the probability that one faction will attack a target of another faction, given the chance. The attitude score cumulatively reflects the effects of the recent incidents in the locale, emotional value of locale, population over/under density, developmental factors, etc. Attitude is also usually an important factor in the fitness functions of the agents.

5.2 Entities and COAs

Entities are indivisible units capable of making and acting upon decisions. Different entity types have

[1] A text version can be freely downloaded from Project Gutenberg at http://promo.net/pg/.

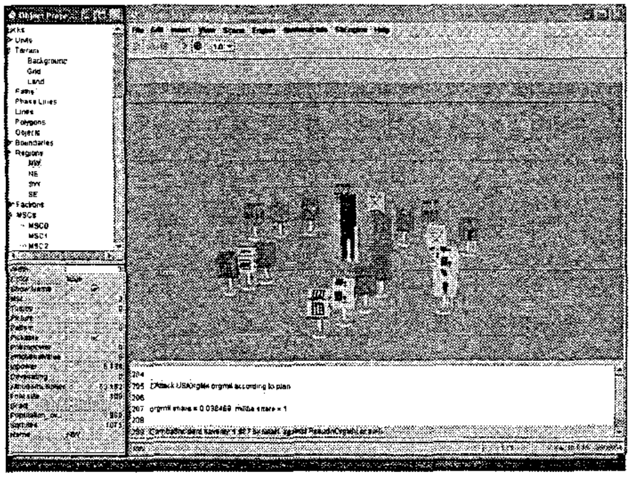


Figure 2. ATACKS Snapshot showing Sheherazade entitites

different COA mechanisms such as a list of movements, times, and targets or a division of power among locales.

For each simulation turn, or clock tick, every entity is located within one of the locales. Entities within the same locale have a probability of interacting, depending upon a number of factors, which include entity characteristics as well as the locales' properties listed earlier.

The entity types Sheherazade offers the scenario designer are organized military, militia, terrorists, information operators and apolitical noncombatants. Each entity type has different subordinate properties and behaviors. An important property all entities have in common is iotype, which can be "calming" or "agitating." Calming entities push the locale attitude down to a degree partly determined by their io power. Entities of type "agitating" push it up in the same way. An entity's iopower is amplified proportionally to the relative size of the local populace in its faction. Every attack and incident drives the attitude up in proportion to its severity.

Each entity type has its own type of COA. In general, COAs are schedules of movements. Entities that engage in combat also have target faction lists. The major exception to this type of COA is the Organized Military entity. Instead, its COA is the distribution of the strengths of its entities among Military Support Commands (MSC), and the distribution of the locales to the MSCs.

5.3 Events

In Sheherazade the entities interact with the environment to model certain real-world events. Clockticks represent time, which can be considered to be a model for either hours, days, weeks, months, or years, depending upon the intent of the designer.

Incidents describe a class of interactions between entities or an entity and the local population. Incidents have an associated "severity" rating to determine the appropriate adjustment to make to attitude levels. Incidents naturally aggravate the animosities of the target faction population, which in turn increases the probability that population or entities of that faction initiate their own incidents. When an incident occurs that includes combat Sheherazade consults the weighted combat values of the contributing entities and assesses combat attrition to each one.

6 Visualization

Our research in this area concentrates on the visualization of the novel symbologies of the entities and their behaviors. We provide visualizations of specific actions, as well as abstracted, conceptual displays of the relationships of entities and regions. We use the Advanced Tactical Architecture for Combat Knowledge System (ATACKS), a three-dimensional visualization tool previously described in [7], to show the movements and graphs of a simulation run.

The visualizations of a simulation run play an important role in our development and analysis of Sheherazade. ATACKS provides a graphical user interface to setup a scenario and then provides several displays to show important events and values for a scenario.

6.1 SASO simulation analysis

Once a military expert has used ATACKS to define a scenario, the resulting file is used to run the coevolution algorithm. As the coevolution is running, it passes COAs to Sheherazade in order to get back the fitness scores. Sheherazade produces an output visualization file of each of the best COAs for each agent per generation. These data files can then be read by ATACKS to create an animation and several displays that show the COAs, movements, incidents, attitudes, damage (attrition), and animosities for that run. Most importantly, these displays show relationships, such as the effect of movement and incidents on locale attitudes, and relative changes of animosities between factions.

We created a set of easily recognizable icons for each type of entity. Combining the entity icon with the background of its faction color forms a colored icon for each entity. These colored icons are then placed on three-dimensional units that move between locales, as shown in the snapshot of Figure 2. The colored bar graph in the center of the locale indicates the percentages of population which belong to each faction in that locale. For each clocktick, entities move in and out of locales in the 3D environment of ATACKS. However, an overall view of the movements and events has proven more helpful in understanding scenarios.

Using consistent colors and icons, a movement graph appears in Figure 3, showing which entities have moved into a locale. The background colors indicate locales; the numbers on the bottom indicate on which clocktick an entity moves. The bar on the right of each icon indicates the iopower of that entity, a value that significantly affects attitude (depending on the type of entity) of that locale. The color of the bar graph indicates whether the entity is calming (white) or agitating (red). Thus, this display conveys entity type, entity allegiance, clocktick moved, locale moved to (and out of), relative iopower, and type of iopower, or six pieces of information for each icon.

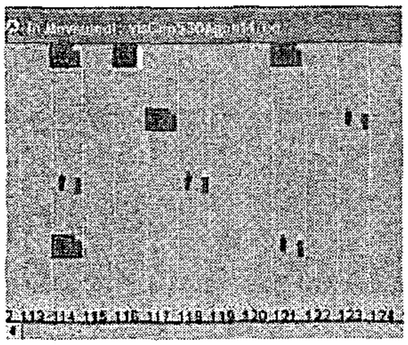


Figure 3. Movement graphs

A similar display can be used to show COAs, except that the color of the indicator bar on the right of the entity icon is set to the target faction. Furthermore, the COA display is interactive. A right click on the icon allows the user to change the target of the entity, and the icons can be dragged and dropped on a different clocktick and locale.

The COA of the organized military unit has a different structure. Its display simply shows which entities have been assigned to which MSC and which MSCs are assigned to each locale. MSCs can be responsible for more than one locale.

Other factors, such as faction animosities and locale attitudes are important influences on the SASO simulation that may be a part of the fitness functions. The animosities are graphed on a starplot. For example, the first graph in Figure 4 shows the animosities for and against the US./OrgMil faction. Each radial corresponds to a faction, consistent with the colors of the entities. The further away from the center, the more animosity the US/OrgMil has for the corresponding faction. The middle of the radial denotes a neutral attitude. By connecting the points on each axis, the blue line denotes the feeling of the US/OrgMil faction for the other factions. The red line denotes how the other factions feel about the US/OrgMil faction. The starplot on the right of Figure 4 shows the animosity for the Northern Alliance faction of our example. From the relative sizes of the triangles formed by the red and blue lines, it is obvious that the Northern Alliance seems to be a generally more disliked faction than the US/OrgMil. Furthermore, its animosity toward other factions seems to be much less than their animosity toward it. This implies that the Northern Alliance may be perpetrating unprovoked attacks on the other factions. The "Play" button at the bottom of the starplot when clicked animates the changes in the animosities over a scenario (clockticks). Currently animosities can only increase.

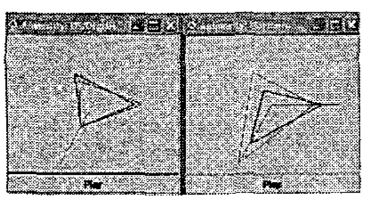


Figure 4. Starplots of animosities

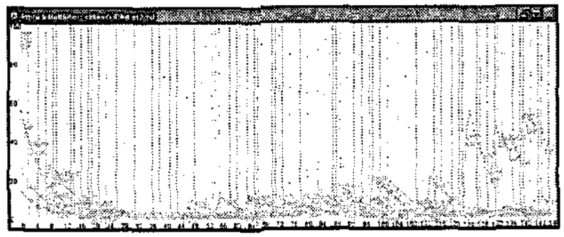


Figure 5. Locale attitudes and incidents

Standard linegraphs, with some annotation, show changes in attitudes, accumulations of damage per faction, and any other factors of interest to the user. The linegraph shown in Figure 5 graphs the attitudes of the four locales over clockticks, color-coded again to the locale colors. Each line for each locale is labeled with the incidents that occurred at that clocktick. The graph in Figure 5 shows calm attitudes in the locales, and more and more incidents as time goes by. A large jump in incidents occurs in the SW locale toward the end of the scenario, agitating that locale. This graph illustrates a strategy by one of the factions to increase the attitude in that locale.

Currently, the most clear and meaningful information comes from looking at the attitude linegraph. Once the analyst identifies a trend, more information about the incidents can be found on the incident graph. Figure 6 shows an example incident, in which a yellow militia attacks a blue organized military unit.

Therefore, to analyze a SASO simulation battle, a user can examine the various displays, from attitudes to animosities, and movements and incidents. This ability has been invaluable in trying to understand the dynamics of the system as a whole.

Figure 6. An incident

6.2 Coevolution analysis

The coevolutionary process requires visualizations of the simulation in order to compare successful COAs. The genetic algorithm explores the search space by finding COAs that produce better COAs. However, it does not explain why one COA may be better than another COA. By comparing the simulation runs, an analyst can use the visualizations to determine what actual strategy resulted in a better score. Therefore, research on the visualization of the military peacekeeping scenario provides insight into the simulation as well as into the coevolutionary process.

While comparing simulation runs may result in an understanding of one strategy over another, it would be helpful to compare generations of COAs to find the interesting runs that can then be compared. Due to the nature of the multi-sided coevolution, this comparison can not be done easily.

Each agent evolves against the current best set of COAs of the other agents, known as the "hill," and only gets a chance to update the hill with its own best strategy every ten cycles. This strategy allows all of the agents time enough to evolve more mature strategies. Therefore, at generation 1, Agent 1 may change the hill. For the next nine generations, each of the other agents evolves against the current hill. Finally, at generation 10, Agent 2 is allowed to place its best set of movements, targets, etc. on the hill. This system has the effect that each agent's strategy can potentially change radically every ten generations.

In this system, as every agent evolves against the hill, its changing strategies will result in different scores for the other agents. For example, in generation 3, all of the agents evolve against the current hill. As Agent 1 is changing its strategies, the battles are scored, resulting in a value for all of the agents. In our prototype, four agents compete. Therefore, in this example, Agent 1's turn would result in a score for each agent. As each agent has a turn in this generation, each of the other agents gets a score in response to that agent's turn. As a result, with four agents, a generation results in 16 scores.

Table 1. Agent scores resulting from one gernation

Agent's Turn	Which Agent's Score			
1	1	2	3	4
2	1	2	3	4
3	1	2	3	4
4	1	2	3	4

A table of which scores result per generation appears in Table 1. The first line in Table 1 represents Agent's 1 turn to evolve in this generation. Agent 1's best strategy resulted in scores for agent 1, 2, 3, and 4, listed on the same line.

As an example, one interesting factor in the system is called iopower. The amount of iopower each entity has depends on the type of entity and the initial setup by the military expert. Iopower greatly influences the locale attitudes, which is a very useful variable as part of the fitness functions. Therefore, an indication of the agent's strategy is how it has distributed its iopower among the locales. Figure 7 shows the IOpower emphasis of all the agents during Agent 2's turn in each of the locales over 100 generations. The most dramatic changes happen at the generations that are multiples of 10, because that is when the hill changes and all agents must change their strategies in response to the new strategies. Figure 7 also shows that Agent 1 found a strategy in generation 40 that places more of its entities with high iopower into Locale 3 instead of 4. By generation 80, it tries again to place more entities into Locale 4. However, from Figure 7, it should also be noticed that we are comparing four different dimensions of data: locales, agents, and generations, and the agent whose turn it is. To see all of this data at one time (three more sets of Figure 7), or to combine the above line graphs would result in a very cluttered display. Instead, we have developed a "heatmap" [4] display to show the same data, as shown in Figure 8.

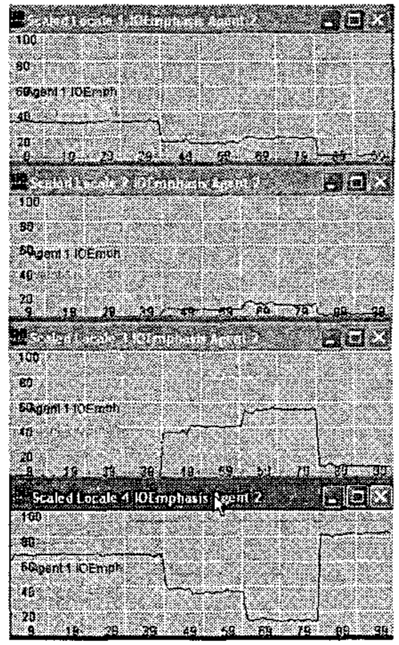


Figure 7. Linegraphs of ioemphasis (Agent 2's turn)

An explanatory key appears in Figure 9. The color of each rectangle indicates the of the iopower emphasis. White corresponds to 0. Progressively more blue indicates higher values. A dark blue indicates the value of 100. This kind of color-coding and multiple position mapping allows the user to see many dimensions of data at one time.

In terms of Bertin [1], this type of graphic makes use of two types of visual variables: the plane and color (or in this case the relative *value* of the color). Furthermore, the plane is used in three dimensions (where the categories repeat both vertically and horizontally by repeatedly listing the agents and the agents' turns across locales and across generations). The use of the plane and color in this

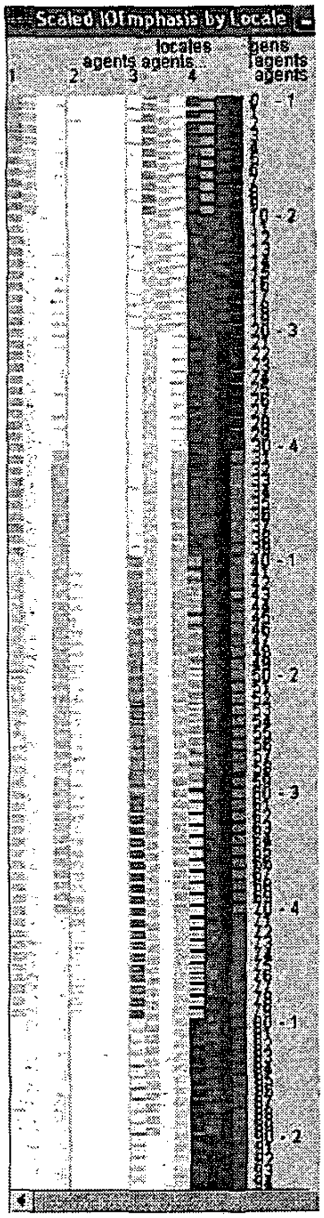


Figure 8. IO Emphasis heatmap

manner allows for a multi-dimensional display. Also, Tufte [7] presents several examples of using color for multi-dimensional displays and also advocates a light background, such as the white used by the heatmap.

The heatmap in Figure 8 clearly shows the trend we saw earlier, in which Agent 1 moves its io power emphasis from Locale 4 into Locale 3 from generation 40 to about 80. At generation 40, Agent 1 had its turn to change its COA on the hill, causing this shift in iopower. At generation 50, Agent 2 gets its turn and places some more of its iopower from Locale 4 into Locale 3 to counteract the changes made 10 generations ago by Agent 1. By the time Agent 1 gets its turn again at generation 80, it has found its strategy was beaten by the other agents, and it once again puts most of its emphasis into Locale 4. Furthermore, the heatmap shows that most of the activity of all of the agents is centralized on Locale 4.

This kind of analysis has led to many insights into our system. Currently, finding which important features to map has been the most difficult problem, but an iterative approach has been helpful. Using visualization to show trends in this complex environment has led us to a much better understand of how the separate rules and the coevolution interact.

7 Conclusion and future work

The current system is a first attempt at creating a meaningful SASO simulation. Extensive research still needs to be done to verify the model and its rules. The visualizations currently are the easiest way to gain insight into the interactions of each part. We will continue to work on refining the algorithms and the visualizations in an iterative process which includes refining the displays each time more insight is gained into the system.

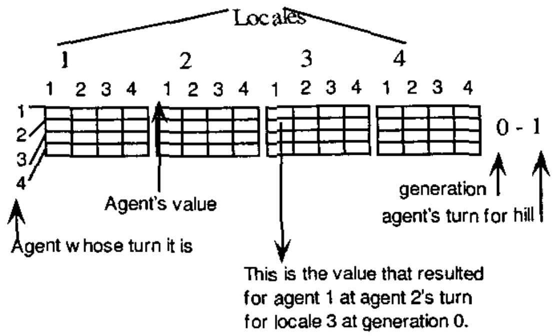


Figure 9. Explanation of heatmap

References

[1] Bertin, Jacques, *Semiology of Graphics*, The University of Wisconsin Press, Madison, Wisconsin, 1983.

[2] Field Manual (FM) 3-07; Stability and Support Operations, U.S. Army Publication, February 2003.

[3] Hillis, D. and R. Winkler. "The fox and the hare: A co-evolutionary approach to course of action generation." *Proceedings of the 5th Annual Fed Lab Symposium*, College Park, MD, 2001

[4] *NeoVision Heatmaps.* Accessed May 2001. <http://www.neovision.com/products/>.

[5] Schlabach, J. L., C.C. Hayes, D.E. Goldberg. "FOX-GA: A Genetic Algorithm for Generating and Analyzing Battlefield Courses of Action." *Evolutionary Computation*, Vol. 7, No. 1, MIT Press Journals, Spring 1999.

[6] Schlabach, J. and D. Hillis. "Sheherazade: A research platform for decision support in military stability and support operations" (BCBL-H Tech Report 2003-01), 2003.

[7] Suantak, L., F. Momen, J. Rozenblit, M. Barnes, and T. Fichtl. "Intelligent decision support of Support and Stability Operations (SASO) through symbolic visualization," *IEEE SMC*, Vol. 5, pp. 2927 -2931, 7-10 Oct. 2001.

[8] Tufte, E.R., *Envisioning Information*, Graphics Press, Cheshire, Connecticut, 1990.