Towards Emergent Design: Analysis, Fitness and Heterogeneity of Agent Based Models Using Geometry of Behavioral Spaces Framework.

Martin Cenek¹ and Spencer Dahl²

¹Department Computer Science and Engineering, University of Alaska Anchorage, Anchorage AK, USA
²Columbia College, Columbia University, New York NY, USA
mcenek@uaa.alaska.edu

Abstract
Detection and analysis of collective behavior in natural and artificial systems is a difficult task which is commonly delegated to a human observer. We present a statistical framework to automatically detect emergent, collective behavior of agents in agent based simulations which exhibit swarming and flocking behavior. Our tunable, translational-, rotational-, and scale-invariant framework – geometry of behavioral spaces – identifies common behaviors among agents and translates these behaviors into a system’s behavioral primitives, along with the agent transitions from one behavioral primitive to another. Finally, we use complex network analysis to detect collectives of agents that gravitate into a common cluster of behavioral primitives as the system’s emergent behavior condenses or decays. We apply complex network theory to the analysis of collective behavior dynamics in the simulations of flocking and swarming to validate our analysis. Our framework does not use the knowledge of the parameter space that drive the models, and only relies on the temporal agent trajectories of exhibited behavior. The utility of detecting emergence from exhibited behavior makes this technique suitable as a fitness function for stochastic search algorithms, analyzing evolutionary dynamics of systems with collective behaviors, detecting structures in artificial chemistry experiments, or analyzing physical system such as bacterial formations.

Introduction
Agent Based Models (ABMs) are widely used to study and model collective phenomena in natural and artificial systems. However, there is no universal methodology for how to construct or analyze these systems. The solutions that are found by a group of cooperating agents that only use local agent-to-agent interactions to problems defined on a system-wide level such as foraging for food, designing grassroots movements in societies, or escaping a predator, can frequently only be understood by a human observer. Designing ABMs with cooperative behavior to solve problems is difficult since it required bridging the scope between the design of individual interaction rules and the system-wide problem definition. We present the results of applying previously developed statistically based computational framework to detect the onset and dissipation of collective behavior in a system using only displayed system dynamics – without any knowledge of the parameter space that controls the system or the laws that drive the system’s components. Our work is focused on analyzing agent based models, although framework can be used in both the analysis of emergent system behavior and system engineering.

In nature, flocking and swarming are examples of the system-wide behaviors used by organisms to interact with their environment as a collective rather than as individuals. Many natural and artificial systems with collective dynamics share common characteristics of how they are constructed: they are composed of simple agents that interact with each other using simple interaction rules, yet the sophisticated collective dynamics of the ensemble results in the system’s evolutionary advantage, survival, or a task completion. We use the flocking and swarming ABM models behavior to test our framework’s ability to (1) generalize agent’s “noisy” behaviors into common behavior groups and (2) differentiate between the model behaviors that only differ by the velocity and shape of the agent ensemble.

Stochastic search algorithms are commonly used to automatically design the interaction rules of the ABM’s agents. These search tools rely on a fitness function to measure the quality of system behavior to solve a problem. Building a fitness function that includes the level of cooperation among agents that leads to desired goal requires significant human expertise and (if successful) results in a custom-tailored, single purpose function that is hard to generalize for ABM design for different environment. Simple fitness functions, on the other hand, often only measure if the simulation reached or made progress towards a goal – regardless of how the goal was reached. It is not guaranteed that the resulting system solves a problem using emergent system behavior. Even if the collective behavior is present in the system dynamics it is usually by coincidence and not by design, which is a significant source of error when modelling real-world systems. We show the utility of our framework in measuring the amount and direction of collective behavior in a system, which can be a useful component of a fitness function design to reward the ABMs with cooperative behavior.

Our analytical framework measures both the quality and
quantity of cooperation among agents. Our analytical tool functions independently of the scale and movement of agent behaviors, is applicable to multiple models and is applicable to different optimization techniques. The bee-hive optimization, particle optimization, swarm robotics are examples of additional techniques developed to solve problems using collective, hierarchical system organization. Although these tools share common principles, there is a lack of tools to analyze system behavior for different optimization tools on varying granularity of analysis. Since our analysis only uses the agent behaviors without any knowledge of the model’s parameters, the framework is independent of the optimization techniques. This allows for a computational comparison of not only different model executions, but also comparison of results found by different optimization techniques.

The presented results illustrate both the ability to generalize stochastic agent behaviors in the system dynamics into common patterns of behavior and differentiate between two very close collective behaviors of flocking and swarming. Plotting the counts of agents in each prototypical behavior at each time-step of the model’s execution allows us to predict the phase transition in the system dynamics from collective to dis-associative and vice-versa. Inspecting the temporal dimension of the behavioral spaces allows us to explicitly measure the level of self-organization, emergence and the directionality of the system dynamics: towards the condensation or decay.

The fitness function design for the stochastic search algorithms to evolve agent systems that solve problems using emergence and analysis of ABM dynamics to analyze the collective behaviors in an ABM model are only two applications of the the geometry of behavioral spaces framework. The framework can also be used to explore the multi-parameter spaces that drive various models, fine tune models to increase performance, or identify a set of high diversity parameter spaces that drive various models, fine tune models to increase performance, or identify a set of high diversity solutions by identifying how they solved a problem. With alternations, the framework can be used as a general pattern finding engine regardless of application.

**Background and Motivation**

The hierarchical system decomposition (HSD) (20), pattern oriented modeling (POM) (4) and morphogenetic engineering (MGE) (3) are common methods used to build systems of agents that solve problems. The HSD is a reductionist approach where the system is built by addition of the constituent components and fails to construct systems with nonlinear dynamics. The POM requires a spatially explicit landscape for its agents to move across and interact with. The spatial nature of these models allows for the use of spatial statistics to detect local, emergent behavior, but fails to identify the emergent behavior on the global scale. The MGE is the most sophisticated and relies on “expressing” the system’s construction to measure its quality. For example, the rule-based swarms and graph-based grammars describe the system construction and self-organization in computational development swarms (10; 13). The solution’s quality is measured as system’s ability to perform its task or the difference between the expressed versus desired patterns (14). The POM and MGE made significant progress towards developing quantifiable ABM models. The general tools to analyze the dynamics of these models lag behind.

Analyzing the non-linear system dynamics has been at the forefront of science for years with a few efforts to analyze the dynamics of the individual and agent based models. Miller used a system of non-linear equations to describe the ABM’s dynamics to measure the effect of changing the model’s parameter configurations to the resulting system dynamics (11). A summary of statistical and mathematical tools to describe the collective dynamics of swarm and multi-agent systems was presented by Lerman et al. (7; 8). The pragmatic efforts of Calvez and Stonedahl resulted in extension and implementation of automatic stochastic search tools to explore the parameter search-spaces of ABM configurations (11; 15). Controlling the vehicle swarms using physics-based expert systems was proposed by Spears et al. (13) while Miner et al. proposed Markov processes to characterize multi-agent behaviors (12).

The individual and system wide dynamics were analyzed using the information theoretic tools by: Van et al. who studied the expected agent behaviors in the KuglerTurvey’s ant colony model (6; 17), Lizier et al. used Shannon based entropy to analyze the micro- and macro-level agent dynamics (9), and the behavior of the swarm robots was analyzed by Wang et al. and Lizier et al. also using the information theoretic tools (8; 19).

The geometry of behavioral spaces framework complements previous research by providing a problem domain independent analysis with the following features: the ability to detect emergent processes from the agent behaviors, producing heuristics to tune the system behavior, the ability to filter agent behaviors of varying frequency and visualize the network of behavioral primitives for both high level and detail inspection of the model’s constituent behaviors. One feature of our framework that is not addressed by the previous work is the ability to provide a real-time system monitoring to measure the velocity and direction of the system’s condensation or decay towards organized behavior.

**Methodology**

Geometry of behavioral spaces framework is a multi-step process that analyzes behaviors for common patterns of behavior. First, each agent in the model records the direction of its movement then the statistics and similarities of agent behaviors at each time step are computed. The second scan of the recorded behaviors is used to construct a behavioral transition state-space and the distribution of how many agents at each time steps were in which behavioral primitive.
The following sections will describe the details of each analytical step: simulation space discretization and recoding of agent trajectories (Figure 1a), at each time step agent’s past and future windows of behavior are first compressed and then logged into the co-occurrence matrix (Figure 1b, c), then the groups of common past behaviors that have the same or similar enough future behaviors are created and define the behavioral primitives of stable behaviors (Figure 1d), and finally, the behavioral space is constructed to reveal the system-wide transitions among behavioral primitives along with the likelihood of agents to transition from one behavioral primitive to another stable pattern of behavior (Figure 1e,f). The final analysis of the behavioral space is used to differentiate if the system dynamics are condensing into the flocking versus swarming behavior or if the system behavior is decaying into a random agent behavior. The complex network analysis is showed in the results section (Figure 3 and 6).

Data Collection and Space Discretization

In order to report agent trajectories, every agent records its initial position and compares that position to its current position. When the distance travelled exceeds a threshold value, the agent resets its initial position to its current position, repeats this process. In this manner high-frequency (small-scale, repetitive) behaviors are not recorded, as they fall below the threshold for distance travelled from the previous report, whereas large scale movements are more frequently reported because these agents move more quickly and therefore report more often. Using this approach, we can ensure that every agent experiencing the same behavior reports in the same manner, independent of the agent’s spatial disposition within the model.

At the end of the simulation, each agents trajectory is a vector of directions that the agent moved with values ranging from one to four, the agent records the behavioral transitions in two subsequent steps creating the state-space transition matrix $T$.
agent’s past history and the $\delta^+$ subsequent moves is agent’s future history.

**Co-occurrence matrix $M$**

Each agent’s past and future histories at each time step $t$ are compressed into a vector of length four, where each vector position (attribute) has the number of times the history contained the move in a given direction (Equation 1). The number of cardinal directions to record agent’s movement can vary to yield higher resolution analysis or to reflect the different space tessellations. In this paper, we use four cardinal direction.

$$v(\delta^-) = \langle \Sigma(1 \in \delta^-), \Sigma(2 \in \delta^-), \Sigma(3 \in \delta^-), \Sigma(4 \in \delta^-) \rangle$$

$$v(\delta^+) = \langle \Sigma(1 \in \delta^+), \Sigma(2 \in \delta^+), \Sigma(3 \in \delta^+), \Sigma(4 \in \delta^+) \rangle$$

(1)

The co-occurrences of encoded past and future behaviors $v(\delta^-)$ and $v(\delta^+)$ are logged into $C[v(\delta^-_x), v(\delta^+_x)]$ for all time steps $t$ (Equation 2).

$$C[v(\delta^-_x), v(\delta^+_x)] = \Sigma_{ij} (v(\delta^-_{x+t}), v(\delta^+_x))$$

(2)

**Behavioral Primitives**

Each row of the co-occurrence matrix $C$ is a likelihood distribution of a given past behavior resulting in any of the observed future behaviors. The next steps, we group all such rows that are similar to each other, creating clusters of past behaviors that resulted in sufficiently similar future behaviors to be considered a behavioral cluster - a behavioral primitive $\epsilon$ (Equation 3). We used the $\chi^2$ test of statistical independence between any two rows that were not previously assigned to a behavioral primitive to measure the similarity between two row distributions of past behaviors. Other tests can be used to calculate this similarity.

$$\epsilon_i = \text{if} \left( \chi^2(C[i,:], C[j,:]) < \alpha \right) \text{then} \epsilon_i \cup C[j,:]$$

(3)

**Behavioral Space $T$**

The behavioral matrix $T_{x \times p}$ describes the high-level agent behaviors and the transition of agent’s behavior from one stable behavior to another. To construct the matrix $T$, we scanned the agent trajectories for a second time. For each agent, we look up which behavioral primitive agent’s past behavior belongs to at two consecutive time steps $t$ and $t+1$ $p = v(\delta^-_x)$ and $r = v(\delta^+_x)$ respectively. We record the agent transitions between two behavioral primitives into a matrix $T[p, r]$ - a state-space transitional matrix of exhibited behaviors in the simulation (Equation 4).

$$T[p, r] = \Sigma_{\epsilon_k} (\epsilon_p : (v(\delta^-_x) \in \epsilon_p), \epsilon_r : (v(\delta^+_x) \in \epsilon_r))$$

(4)

![Figure 2: A complex network visualization of the final behavioral matrix $T$ for swarming (top) and flocking (bottom) models with the system behavior varied in 400 time-steps from cooperative to random. The communities of highly connected nodes were further clustered into communities and colored with the same color.](image-url)

**Geometry of Behavioral Spaces**

The final step is a complex network analysis of the matrix $T$, with the behavioral primitives being the network’s nodes and the behavioral transitions are the network’s edges. We further clustered the tightly coupled behavioral primitives into the communities of closely related behaviors, filtered transition edges with low edge weight, and used a network layout to visualize the behavioral space [5]. For additional details of the behavioral spaces analysis, please see [Cenek and Dahl] in press.

To construct a dynamic view of the agent behaviors during the simulation, tracking the progress towards the condensation or dissipation of collective behavior, we counted...
Figure 3: Agent counts in each behavioral primitive at every time-step of the swarming (left) and flocking (right) model executions with the oscillation period between condensed cooperative and decayed random behaviors every 400 time-steps. At each time-step a bar chart shows the top 25 behavioral primitives with highest counts of agents that were selected (globally) at the end of the simulation. Each color of in the bar represents one active behavioral primitive and its size is proportional to the number of agents with that behavior at that time-step. The top of each plot shows a miniature snapshot of the active transition edges of the behavioral matrix $T$ as a complex network 200 time-step intervals.

how many agents, at any given time step, are at which behavioral primitive. Figure 4 shows the histogram of the top 25 most frequent behavioral primitives and the same aspect is showed in Figure 2, where the network nodes representing each behavioral primitive have their size proportional to the count of agents in that behavior (the node’s weighted degree).

Results

We applied the geometry of behavioral spaces framework to the flocking and swarming models with global co-operative behavior among agents [16]. To illustrate the framework’s ability to detect the regime changes between the cooperative and random system behaviors, we varied the model parameters that control agent’s ability to coordinate with other agents (alignment, vision, radius etc.) every 200 or 400 time-steps to force the cooperative system behavior or its dissipation into a random, dis-associative behavior.

The models ran for between 1500 and 1900 time-steps which allowed for 7 and 2 regime changes at 200 and 400 time-step periods respectively. The system dynamics were analyzed using the past and future history vectors of 15 components long, four cardinal direction of reporting agent movement, and the threshold $\alpha = 0.05$ for $\chi^2$ measure to group the rows of co-occurrence matrix $C$ into the behavioral primitives.

The complex network view of the behavioral space only shows the behavioral primitives of the giant component with the degree $> 0$ and the minimum edge weight of 10. Figure 2 shows the global behavioral landscape of for two models. The network’s node and edge size is proportional to their weighted degree and weight.

The node clusters with the same color identify the highly connected behavioral primitives [5]. These communities of behavioral primitives show common transitions among agent behaviors. For example, the behavioral landscape of the swarming model (Figure 2 top) has the behavioral primitives of condensed cooperative behavior organized as the center cluster of nodes. The peripheral behavioral clusters represent the agent behaviors after system decayed into random behavior. The communities of behavioral primitives in the flocking model (bottom) are interconnected, since the model’s condensed emergent system behavior has several small groups of agents that move in one of eight generalized directions. Each direction is seen as one community of nodes with the same color. After the decay of global cooperative behavior, the agents trajectories slowly diverge and increasing number of behavioral primitives are activated. This change of behavior can be seen in the time-slice plots of the active behavioral spaces as the miniature sub-networks (top) in Figures 3 (please note the coloring and layout for these miniature graphs is different than in Figure 2). A window of 20 time-steps was used to show the active edges of the behavioral space at each time-slice (Figure 3 top).

Analysis and Discussion

All system measures presented in our result plots are quantitative measures of system dynamics and can be used to analyze a system’s emergent behavior, to be included as a parameter in the fitness function of the stochastic search algo-
We executed the model with different random seed, but the analytical framework’s results were stable with little variation in the reported dynamics. We do not report the results of the statistical validation and variance of the analytical framework. The stability of the analysis to generalize the system dynamics can be seen as highly correlated agent counts in each behavioral primitive in the series Swarming200 vs. Swarming400 and Flocking200 vs. Flocking400 showed in Figure 5.

Note that the "random" behavior is different in each of the models. In the flocking model, the disordered regime constitutes the agents slowly drifting apart at random which means they dissociate slowly from their ordered flocking and never reach a fully random state. In the swarming model, the random behavior is agents moving in completely random directions almost instantaneously after the model parameters were changed. The regime shifts are showed in the Figure 3 as the sudden increase in the counts of behavioral primitives between the steps 400 and 450. This trend parallels the increase in the system behavior diversity. The difference between how the two systems decay to random behavior is the difference between the sudden increase of in the behavioral primitives counts in the swarming model (left) versus the gradual decay (the gradual increase) in the flocking behavior (right). Behavioral primitives histogram in Figures 3 and 4 show the same dynamics features as the time series distributions in Figure 6. They both can be used to detect when system dynamics are condensing towards stable, cooperative behavior and dissipating into random behavior.

The ability to generalize the stochastic agent behaviors into stable behavioral primitives can be seen in all histogram figures (Figures 3, 4) and the system dynamics measured on the state-space transition networks (Figure 6). The oscillation of model’s behavior between the cooperative and random regimes resulted in the histogram’s oscillations, but more importantly, the framework repeatedly generalized the agent behaviors to the same behavioral primitives. This is seen as the same color pattern and structure in the histogram’s periods. The framework also characterized the agent behaviors to the same behavioral primitives in the independent model executions with the regime changes every 200 or 400. The histogram pattern in the Figure 4 is close to identical to the histogram in Figure 3 if the time axis was compressed.

Figure 6 shows the same observation about the framework’s stability to generalize agents’ random behaviors. Regardless of the oscillation period between the cooperative and random regimes in different model executions, or the repeated condensation and decay of behaviors within the same model execution, the shape difference between a pair of model’s executions is minuscule. The difference in offset between the 200 and 400 plot-lines is because the latter model execution lasted 500 time-steps longer and resulted in higher agent counts in each behavioral primitive.

Figure 4: Counts of how many agent behaviors were recorded in each behavioral primitive during the models execution. The x-axis shows all possible behavioral primitives sorted in the alphabetical order from `<0,0,0,15>→<15,0,0,0>`.

The qualitative analysis of the system’s behavior is done by construction of the behavioral landscape networks showed in Figure 2. Note that although the framework’s parameters are tunable, we ran the analysis "straight out of the box" without fine tuning any of the analytic framework’s parameters to identify different behavioral features from the system’s dynamics.
All analytical measures applied to the resulting geometry of behavioral spaces on the system dynamics of flocking and swarming models clearly show the system’s condensation towards cooperative behavior and subsequent dissipated towards random behavior. In this context we used the term “geometry” to refer to the structural features calculated using the complex network analysis. Figure 6 shows the overview of the network’s measures that can be used to identify system’s dynamics towards a regime shift. A sudden increase or decrease in the measure’s slope indicate the direction and the velocity of system’s impending behavior regime shift towards condensation or dissipation. A stable system behavior (cooperative or random) will have the measured slope near zero.

Stochastic search algorithms are popular tools that do not need to know how to find a solution, only what needs to be solved and if the candidate solution made a progress towards a desired goal. As explained earlier, designing a fitness function that drives the automatic searches should be inclusive of different aspects of evolved system dynamics. In this case, reward the solutions that (1) use cooperative behavior among agents to solve a problem and (2) are different than the rest of the solutions. If a stochastic search function found multiple methods of solving a given problem, the final state-space transition matrices and the complex measures of the geometry of the behavioral spaces are one way of differentiating how different candidate solutions solved the problem. A correlation measure between two system measures in Figure 6 will reveal how different the system dynamics are from each other. This allows for rewarding the solutions with different system dynamics than the rest of the solutions that also solved the problem, creating a metric for originality in computer models and a mechanism for creative problem solving.

In the future we plan to test our methodology on a broader range of models with emergent system behavior and focus on evolving agent rules using a fitness function that incorporates the measure of cooperation among system agents to achieve the simulation goal. Outside of the ABM application, we plan on using the methodology to analyze the recorded trajectories of bacteria from a high-speed, high-power video feed to automatically detect the formation of bacterial colonies. Using the framework’s strength as a general pattern finder, we hope to apply our methodology to detect the emerging signatures in cyber-security intrusion attacks.

Acknowledgements

This work was partially funded by Alaska NSF EPSCoR grant 1208927. The authors would like to thank all project collaborators for their comments and suggestions.

References


Doursat, R., Sayama, H., and Michel, O. (2013). Morpho-


