

Emergent Behavior as a Function of Information

Aaron Gorenstein

May 5, 2009

Abstract

In this paper, I briefly introduce and summarize the state of the art of emergent behavior. I also present my own experiment and findings. I used a competitive domain to investigate how information contributes to emergent behavior. Specifically, I attempt to see if there is a certain required level of information for emergent behavior to arise. By using the competitive domain to decide “which team is stronger”, I simply examine the results of a round-robin tournament between teams of agents. The only difference between each team is simply how far the member agents can see. My hypothesis was that there would be a certain “cutoff” point, where after a certain sight range teams, having access to enough or more than the sufficient level of information, would act dramatically better. Sadly, this hypothesis was most likely falsified by the evidence. What I did find was that both too much and too little sight range creates weaker teams, while a middling sight range creates the strongest. This paper ends with a discussion of what next steps might be taken in investigating emergent behavior in this fashion.

Contents

1	Introduction	3
1.1	Emergent Behavior	3
1.2	Current State of the Art	3
1.2.1	Ability to Predict The Behavior of an Emergent System	4
1.2.2	Ability to Predict the Emergence of an Emergent System	4
1.3	The Purpose of This Research	4
1.3.1	The Hypothesis	4
2	Experiment Setup	6
2.1	The Domain Chosen for this Paper	6
2.1.1	Importance of Competitive Domain	6
2.1.2	Will Emergent Behavior Even Arise?	7
2.2	Issues with Experimentation as a Research Tool	7
2.3	Implementing the Domain, the Simulator Software	8
3	The Experiments	10
3.1	Policy Vectors	11
3.1.1	Manually-Set Policy Vectors	12
3.1.2	Mapping/Normalizing Perception Values	12
4	Results	13
4.1	Method of Gathering the Data	13
4.2	Method of Organizing Data	13
4.3	Method of Analyzing the Data	13
5	Discussion	14
5.1	Data Set 1	14
5.2	Data Set 2	14
5.3	General Trends	15
5.4	Hypothesis — Verified or Falsified?	15
6	Conclusions	16
6.1	In What Way Does Information Promote Emergent Behavior?	16
6.2	Measuring Coherence	16
6.3	The Idea of a Competitive Domain	16
6.4	Looking Forward	17
A	Figures for Tournament 1	18
B	Figures for Tournament 2	23

1 Introduction

This paper presents a brief introduction to the topic of emergent behavior, some past and present research in the field, and details my own research in subject. My work was an attempt to help define a formal requirement of emergent behavior, through a series of experiments and simulations. As a result, I found that I needed to also investigate *how* one detects, measures, and evaluates emergent behavior in a system. Before continuing, I will attempt to explain emergent behavior, and summarize the state of the art.

1.1 Emergent Behavior

Emergent behavior can be called a kind of global behavior that arises from many agents interacting in a system, but is not attributable to any particular agent[PMKD02, VVPV97, Nit05]. Some common examples of emergent behavior that arise in nature are hive insects such as bees, ants, and termites, as well as herding behaviors in groups of land mammals, and flocking behaviors in birds[Par97, Mat92, HK04]. However, emergent behavior can often arise in the technological world. Computers can communicate with one another through networks, robots communicate with one another via radio signals, and our factories work off of complex, automated schedules[Nit05, VVPV97, Mog06]. Unfortunately, as for now, most examples of emergent behavior in our technology are for the *worse*[VVPV97, Mog06]. Everywhere we are creating complex, densely connected systems, where countless agents can interact. Clearly, then, the ability to analyze, predict, measure, and in any way *understand* emergent behavior is critical for knowing simply how all of our systems truly work.

1.2 Current State of the Art

There are many papers discussing the powers, applications, and abilities of emergent systems[LSL06, Nit05]. Many revolve around implementing emergent behavior already found in nature[Par97, Mat92, HK04, Nit05], but others focus on behaviors somewhat divorced from the natural world[Mog06, Nit05]. Those papers demonstrate inspiring results on how very simple rules can emulate ants foraging for food, or termites building their nest, or animals herding. There are also papers that present ideas on how to analyze and predict and formulize emergent behaviors in systems, some from a biological perspective and others from a purely theoretical or technical[SBB01, RVH⁺04b, HRRT05, WZ07, VVPV97]. A notable disparity is that, while there seems to be some settled-upon — and quite elegantly simple — implementations of simple emergent behavior such as those listed above, there seems to be very little agreed upon in methods of *analyzing* emergent behavior. This is certainly understandable, as by its very nature emergent behavior is extremely difficult to predict¹. This paper is an attempt to contribute to the communities' ability to analyze emergent behavior.

¹Indeed, a few papers have said that emergent behavior is precisely that which is *not easy* to predict!

Within the general label of “analyzing”, however, there are several logical ways of dividing that topic up into sub-topics. Here I will discuss the difference between attempts at analyzing the behavior of emergent systems, versus analyzing what makes them “emergent”.

1.2.1 Ability to Predict The Behavior of an Emergent System

Most of the papers analyzing emergent behavior attempt to focus on, well, the behavior part of it[RVH⁺04b, HRRT05, WZ07]. The benefit of successfully analyzing the behavior is obvious — one gets to understand what actually would happen with the emergent system! They attempt to analyze emergent behavior through various models and formalisms[RVH⁺04b, HRRT05, SBB01, WZ07]. Using models to predict the behavior allows researchers to attempt to design a system that is purposefully emergent[RVH⁺04b, HRRT05]. On the other hand, some papers hold that *simulation* is a strong way to analyze the behavior of emergent systems[VVPV97, LSL06]. While much of the literature I found focused on analyzing and determining the behavior of an emergent system, it seemed that there was a severe lack of research on *when* or *why* emergent behavior actually arose in a system.

1.2.2 Ability to Predict the Emergence of an Emergent System

The ability to predict the “emergence” part of emergent behavior is obviously critical to understanding the subject. However, while there are many formalisms attempting to describe the behavior of an emergent system, it seemed that there was little research trying to understand how those behaviors precisely emerged. Early research did present “rules of thumb” for identifying emergent behavior[VVPV97]. Early research also showed that the individual agents need not be very smart at all to demonstrate the typical behavior examples, such as flocking or herding[Mat92]. However, there seems to be, simply, a lack of any ability to formally say “*this* level of interaction is required for emergence”, or anything of the like.

1.3 The Purpose of This Research

The purpose of this research is **not** to create a way to formally say such a statement, because that is a huge challenge. Rather, the experiments presented were an attempt to provide initial measurements on how levels of information available to individual agents determined the exhibition of emergent behavior.

1.3.1 The Hypothesis

Consider: intuitively, a large set of agents that simply cannot communicate with one another, and cannot observe the world around them, should really not be able to exhibit any sort of emergent behavior. On the other hand, a set of agents in which each contain total (or at least, most) information about their

surroundings and are smart enough to *compute* how to work in concert seems also not emergent.

Thus, my hypothesis is simply this: for emergent behavior to arise, there needs to be a certain density of information available to each agent that makes up the system.

I test this using a software simulator of my own making, and use it to “host” a round-robin tournament between agents in a competitive domain. The only difference between the agents is how “far” they can see in the simulation world — the ones that see farther clearly have more information than those that do not. A key assumption that was deemed necessary is that the team of agents that wins ought to have acted in a more emergent fashion[Gor09].

The hypothesis would be verified if the experiments showed that a certain “relationship” between information levels resulted in a likely outcome for one side. The hypothesis would be falsified if there was no clear rhyme or reason to which team was the winner in any given matchup.

2 Experiment Setup

Here I will discuss the experiment itself (independent of the software) and explain my reasoning for it. It will conclude with a description of some implementation details for the simulator software.

2.1 The Domain Chosen for this Paper

Most of the experiments in emergent behavior have the same general setup: a set of simple agents that will ultimately work together, and perhaps a few immobile obstacles [Mat92, WZ07, Par97]. However, I felt that while these domains were certainly worth exploring, for my experiments I chose a more unusual one — a competitive one. Essentially, there are two teams of agents, blandly called “red” and “blue” due to their colors displayed on the GUI. These agents all exist on a 2D plane of discrete square cells. Agents have four behaviors to choose from: attack nearest enemy, avoid enemies, move towards nearest friend, or avoid friends. They act by either moving or attacking in a cardinal direction, as appropriate, each timestep. If an agent is attacked often enough by agents from the opposing team, it sadly perishes and is removed from the simulation.

The agents can also investigate their immediate surroundings, and receive a summary in the form of a *perception vector*. The agents can also query nearby friendly agents, and adopt what *they* see as their own perception vector. In this sense, agents can communicate, and the most important messages spread around the system.²

As this research is an attempt to study what effect “more or less information” has on an emergent system, the *sight range* value, which dictates just how far away each agent can see from its position, is the adjusted variable. The round-robin tournament was essentially between 10 teams, identical to one another except that each had a different sight range. Team 1 has sight range of 1, team 2 has sight range of 2, all the way up to team 10 with sight range of 10. With the *sight range* value being the only changing variable in the simulation, I hoped that this round-robin tournament would help reveal what role in emergent behavior information played.

But why a competitive domain?

2.1.1 Importance of Competitive Domain

As stated before, emergent behavior can easily arise from computer networks, and other systems of interacting technology. The current emergent behavior research is helpful towards understanding those systems. However, it seems to me that there are *so many* examples of interconnected systems, that there may be multiple emergent systems in the same space. Studying this competitive

²The idea of a red team versus a blue team in a competitive setting is incredibly common for computer and board games. However, in the context of emergent behavior, this is *not* the first paper to use such a system — the ISAAC system [Ila94] was the first, as far as I can tell. That system was in some ways significantly influential in the design of this software, the exact extent of which will be discussed later.

domain may hopefully lead to some insight on how *entire emergent systems* interact with one another. Furthermore, many of the aspired, future uses of emergent systems involve pitting them against rival systems. A proposed NASA mission involves sending a fleet of simple, emergent-behavior-exhibiting drones into the asteroid belt[RVH⁺04b, RVH⁺04a, HRR05]. Perhaps that mission can be best seen as one emergent system interacting with another – the asteroids. An exciting medicinal application would be a host of microscopic robots entering a human body and killing viruses[Hol07]. Clearly, that can also be construed as two competing emergent systems.

Clearly, studying emergent behaviors in competitive domains is a necessity, and so this research uses such a domain.

2.1.2 Will Emergent Behavior Even Arise?

Given this competitive domain, the next questions to ask are “is it sufficient for emergent behavior to arise,” and “why is sight range the way to adjustable variable for how much information density the agents receive?” Fortunately, it seems likely that emergent behavior is quite possible in this domain. As stated before, *very* simple agents can result in emergent behavior[Mat92, Par97]. Indeed, early research concluded that many emergent behaviors can “build up” from simpler ones, in a fittingly bottom-up approach[Mat92, BMM94].

Sight range was chosen as the adjusted variable in the experiments because there is prior research that indicates it can be the deciding factor for emergent behavior. The ISAAC system was developed to find new ways of simulating land warfare for military-research purposes, using emergent systems[Ila94]. It uses a simulator not entirely different from the one I developed, though theirs adds significant complexity to better reflect military structure. Ilachinski mentions at one point that, when adjusting one team’s equivalent of *sight range*, a team became much more effective and demonstrated much greater cooperation. But as the value was increased further, the team broke down and was very ineffective. These findings, actually tangential to the purpose of the ISAAC paper, seem to indicate that chosen independent variable is the correct one to investigate.

2.2 Issues with Experimentation as a Research Tool

As a tool to understanding emergent behavior, experimentation through simulations is an accepted method[WZ07, Nit05]. However, experimentation has its own flaws. I will list and attempt to address the most glaring of these in this section.

- Is there even a way to extend results I find to the entire field of emergent behavior? Or will the conclusions be limited to just this domain, with just this simulator?

I feel that there is a possibility of broadly applicable conclusions from these experiments. If they show, for example, that the larger the sight

range on a team, the more effective it is, I believe that it is possible to extrapolate some more general conclusions. Perhaps there will not be a neat mathematical formula that computes exactly how much information is required for emergent behavior (indeed, simply measure what numerical value to assign to information is a feat in and of itself!), but I think it reasonable to be able to extend the results to create “rules of thumb” about emergent behavior.

- Given a particular experiment run of this simulation, or any simulation, what tools exist to definitely show that emergent behavior either arose or did not? Is it possible to look at a recording of a simulation, and quantitatively say that “feature X” is indicative of emergent behavior?

How I went about addressing this issue is a significant portion of both the results and discussion sections. Just as it seems difficult to predict what is necessary to bring about emergent behavior, the exact ways to *decide* whether or not emergent behavior is occurring in any given simulation is difficult. However, this issue is in large part dealt with by the initial assumption that, for any given competition, the “winning” team demonstrates “more” emergent behavior.

Conveniently, due to the competitive nature of the domain, the winning team is easily calculated: whichever team has more pieces remaining at the end of the simulation is the winner!

Hopefully this brief discussion has addressed a few of the central concerns about the validity of the experiments.

2.3 Implementing the Domain, the Simulator Software

The last concern, or perhaps question, to address is simply why I chose to implement my own simulator, rather than use an already-existing one. While there are many other simulation programs out there, such as NetLogo[TW04], I thought it best to make my own for a number of reasons.

As this is my first real foray into studying emergent behavior, I thought it would be best if I understood as deeply as possible what concerns and issues went into the software. The best way to gain such an understanding is, of course, to make one yourself. In a similar vein, there is always the need to understand *exactly* what is going on in a simulation, and writing your own addresses that. I also entertained the hope that by implementing the simulator, I would gain further insight into the very nature of emergent behavior. If that occurred, I did not notice. Lastly, I wanted the ability to provide full recording of both the GUI in the form of an animated GIF, as well as “behind-the-scenes” activity in the form of a plaintext log file. Simply put, I was unable to find if there was any software that allowed one to do that to the extent that I wished.

With that reasoning, I created the simulator software. I never named the simulator, so I simply refer to it as “the simulator” and the like. For brevity

(which this paper sorely lacks) I will not include a full description of the simulator software. Here are some choice facts about it.

- It is written entirely in the Python, using a 2.5.2 interpreter for Ubuntu 8.04 LTS.
- For graphics display, it uses the Pygame 1.7 module, which is an SDL wrapper, so it should work on video cards with OpenGL support.
- The simulation proceeds in discrete steps.
- Agents receive sensor input through a *perception vector*, a 5-element list. In order, the elements of that list are, the number of friends seen, the number of enemies seen, the difference between the centroid of friends versus the centroid of enemies, the time in which this perception was made, and the location at which this perception was made.
- A brief sketch of how agents behave is as follows:

As stated before, the agents can exhibit one of four behaviors: attack nearest enemy, avoid enemies, go towards nearest friend, or avoid friends. Given a perception vector to act off of, agents act based off of their *policy vector*³. The policy vector, inspired by the *personality* vector used in the ISAAC system[Ila94], is simply a collection of four vectors. Every timestep, the agent takes four dot products — one for each pairing of the perception vector with one of the sub-vectors in the policy vector. Each of those four dot products refers to one of the four behaviors available. The highest dot product is the behavior that the agent will exhibit.

However, agents are not only able to perceive their immediate surroundings, but prior to making acting on their perception vector, they also “listen” to nearby friendly agents. With a big list of all of *their* perception vectors as well, the agent simply sees which perception, out of all of them, produces the highest-value dot product. That perception, regardless of where or when it originated, will be what the agent chooses as “its” perception vector⁴. A nice side-effect of this implementation is that an agent’s behavior can be entirely described through its policy vector, a mere 24-number encoding.

This was the simulator used for all of the experiments that will be discussed shortly.

³Credit for the name goes to Professor Brown of the University of Rochester Computer Science Department.

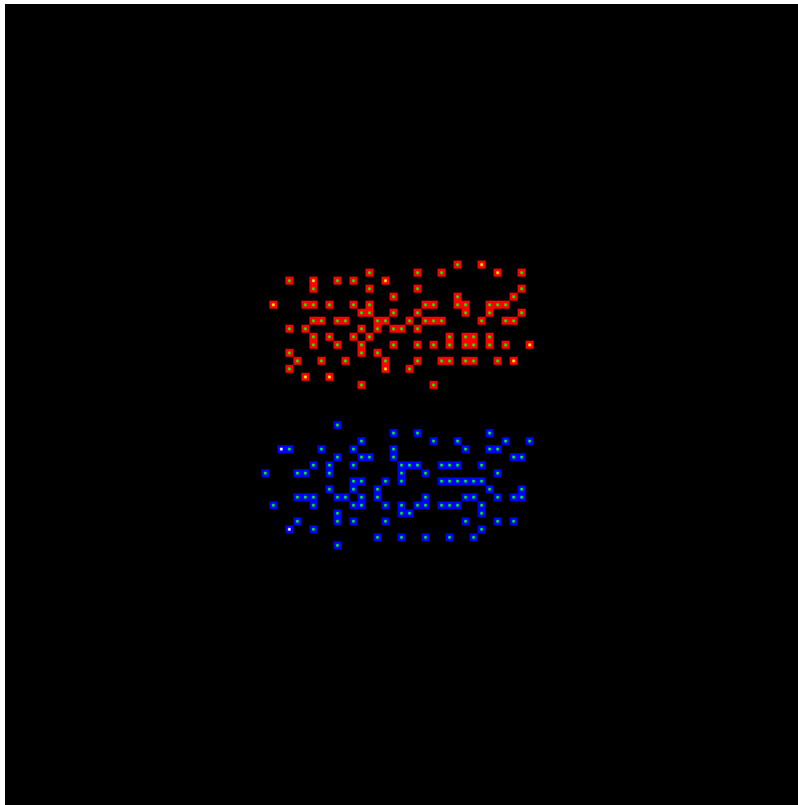
⁴Note that, intuitively, a sane policy vector will have negative values paired up with the time and location value — as a message should be less important as it gets older

3 The Experiments

I ran two sets of experiments for this research. Both were round-robin tournaments between the teams with *sight range* ranging from 1 to 10. They both took place in a simulation with the x and y directions set to 100 — the simulator world had 10,000 discrete cells. The only difference between the tournaments were the initial starting conditions. In the first tournament, each team was placed in its own 30×10 rectangle, a little distance apart. Both teams filled up every third location, so there were a total of 100 agents per side, or 200 agents altogether. The second tournament, each team had 150 agents, but they were scattered randomly throughout the map initially.

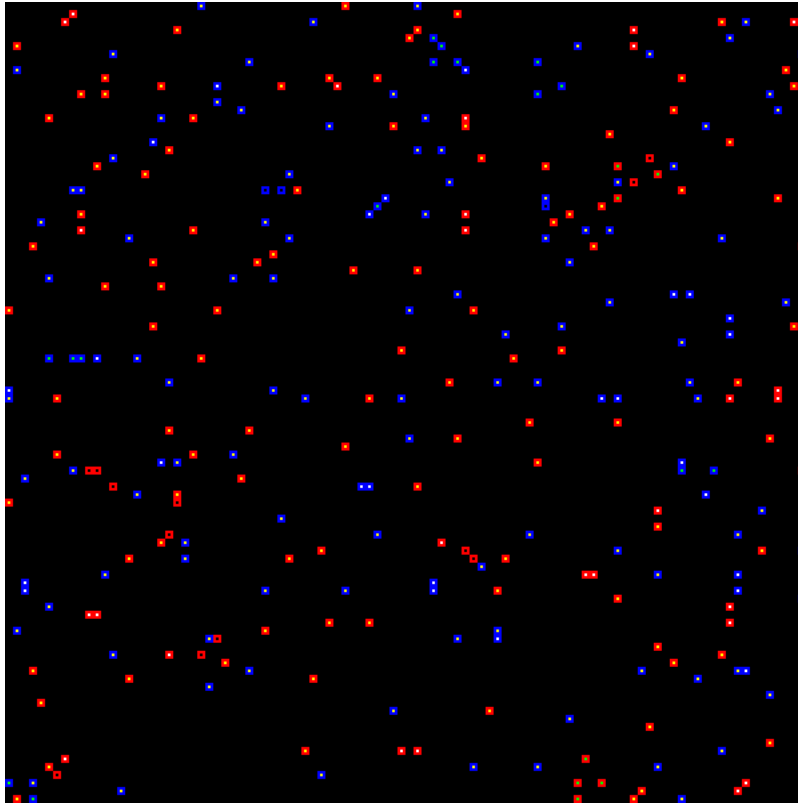
In an attempt to get more robust data, I ran the first tournament three times. Due to time constraints, I was only able to run the second tournament once.

Here is an example of an early timestep in a first-tournament game:



Readers with good eyesight may notice that it appears each square has an “inner square” of a different shade. This is displayed simply for debugging purposes — those inner squares are visual representations of what behavior the agent is acting upon. Black refers to “attack enemies”, white is “approach friends”, yellow is “flee enemies”, and green is “avoid friends”.

Without further ado, here is an example of an early timestep in a second-tournament game:



Included in the submission of this paper are a few choice simulations archived and compressed. I could not include all of my data, especially not in any sort of appendix, because the amount of data produced by these tournaments was substantial. The first tournament, using 200 agents a simulation and 55 simulations⁵, produced around 170 megabytes. The second tournament, using 300 agents a simulation and another 55 simulations, produced around 350 megabytes.

3.1 Policy Vectors

The tricky part about beginning these experiments was determining the policy vectors for the agents. Without question, all agents would use the same policy vector, but how could I settle on the values? Initially, I attempted to “train” ideal policy vectors — simply by randomly generating one, and pitting it against a previous winner. However, such attempts did not work out. I have entertained

⁵That number simply comes from number of games in a tournament: $(\sum_{i=1}^{10} i)$

the idea of a genetic algorithm process, but that was simply beyond the scope of this research.

3.1.1 Manually-Set Policy Vectors

Ultimately, the solution I settled upon was a simple one: I will simply manually set the policy vectors⁶! Thankfully, the underlying implementation of policy-vector interpretation made the manual setting straightforward. For example, it is intuitive that an agent would want to attack enemies when it is far from the centroid of enemies, when it is surrounded by friends, and when it can see many enemies. Thus, those respective weights in the policy vectors would be positive. After determining which values ought to be positive and which negative, it was simply a matter of fiddling until I found ones that had the agents exhibit sane-enough behaviors. In an attempt to “normalize” everything, I tested these values with *sight range* set to 5 — the middle of the range of values that I would be testing.

Here are the values used in these experiments:

Action	# Friends	# Enemies	C Diff	T Diff	L Diff
Attack	2	10	-1	-0.1	-0.1
Retreat	-6	20	-1	-1	-0.5
Approach Friend	-2	4	2	-1	-0.5
Avoid Friends	2	-50	1	-1	-0.5

C Diff refers to the centroid distance, which is to say the comparative distance between the centroid of friends, and the centroid of enemies. T Diff is the time difference — how recent is this message? L Diff is the location distance — how far was the origin of this message to this current recipient?

One can see how certain values make a certain action more or less preferred, given a certain perception.

3.1.2 Mapping/Normalizing Perception Values

An especially astute reader may notice a potential flaw: the perceptions of an agent grow disproportionately to one another, given an increase in sight range! That is to say, increasing the sight range will potentially increase the number of seen friends or enemies by roughly a quadratic function of the sight range. However, the *distance* between the centroids will only increase by about a linear function! This has the effect of making policy vectors have wildly different effects, even for roughly equivalent perceptions, for agents with different sight ranges. Thus, behind the scenes, the simulator actually performs some functions to normalize the values, and divide them by either the area observable by the agent, or the sight range of the agent, as appropriate. So, in actuality, the “number of friends” value is more like “density of friends,” and so on.

⁶Again, credit goes to Professor Brown of the University of Rochester for this suggestion

4 Results

In this section I will explain the various methods used data to get results from the experiments. The discussion section, appropriately, will contain the real discussion about the data.

4.1 Method of Gathering the Data

The results were culled from the plaintext log file produced by every simulation. The numbers gathered are simply various ratios and relationships based off of how many agents there were per team at the *start*, compared to how many remained at the *end*. After all, as stated before, a core assumption is that the more successful team demonstrates “more” emergent behavior.

4.2 Method of Organizing Data

The data, as one can find in Appendix A and B, is organized into a series of charts. Appendix A has the charts for tournament one, where the teams were arrayed in initial formations. Appendix B has the charts for tournament two, where the teams were scattered all across the map at the start of the simulation. In each appendix there are ten charts, one for each team, so the first chart in appendix A is for team 1, the second for team 2, and so on. Each chart is organized like so:

The *Y* axis is between 0 and 1, and refers to a percentage — 1 refers to %100, and so on. The *X* axis iterates through all 10 possible opposing teams. The blue bars refers to what percentage of the *main* team remains — so for the chart with Team 3, the blue bar above point 5 refers to the percentage of Team 3 members that remained at the end of a matchup between Team 3 and 5. Likewise, the red bar refers to the percentage of the *other* team remaining. So, again, on the fifth point in the Team 3 chart, the red bar refers to the percentage of *Team 5's* agents remaining. Lastly, if one were to call pieces removed during a simulation “casualties”, the yellow bar refers to what percentage *of the casualties* were for the *main* team. These three values are computed for every possible matchup.

4.3 Method of Analyzing the Data

The simulations were in fact very time-consuming, and so sadly I was only able to run the few I have here. I believe that there are enough data points to begin identifying underlying trends, and draw conclusions from those. However, there are not enough points to start running statistical or regressions-based analysis — there simply are not enough numbers to make those meaningful. Thus, I simply looked at the charts, and many of the simulation recordings, to try to identify what was going on, and how emergent behavior did or did not arise.

The interpretation of the results are in the discussion section.

5 Discussion

The data, between data set 1 (data from the first tournament) and data set 2 (from the second tournament) has both significant differences and similarities. First I will discuss those trends that seemed particular to each tournament style, and then the general trends. Lastly, I will discuss how these findings either verify or falsify my hypothesis.

5.1 Data Set 1

Originally I had hoped that the nature of the first tournament, with its “starting formations,” would allow for really clever emergent behavior to arise, especially from those with longer sight range. Unfortunately, it seems that almost the opposite happened — larger sight-range teams, for tournament one, would often charge straight into the middle of the enemy team, allowing themselves to get surrounded unless if their opponents were still more zealous. For this reason, high-sight-range teams (around 8, 9, and 10) performed surprisingly poorly.

However, teams with very little sight range also performed poorly, but very reliably poorly, against all opponents. These were the teams with sight range 1 or 2. This trend seems to lend itself to verifying the hypothesis — is sight range 3 this “cutoff point” for information density, the line between non-emergent and emergent behavior? Sadly, this single data set is far too small to verify the hypothesis.

The best performing teams were in the middle sight range — around 5 to 7. This may have been in part because I originally hand-tuned the policy vectors for sight range 5, but at least in tournament 1 it was team 6 that seemed to perform even better. That little detail could maybe imply that emergent behavior does not “commute” very well between different sight ranges — perhaps a larger sight range could reliably beat a shorter, but with different policy vectors.

Overall, it seems that this data set gives some small hope to the hypothesis, in that there is the cutoff-in-efficacy effect at sight-range 3. That a team becomes *less* effective over an increase in sight range was observed before, in the ISAAC report[Il94].

5.2 Data Set 2

Here we see much a more linear growth in team effectiveness and sight range increases. The number of surviving team-members grows pretty steadily from chart 1 to chart 2, and so on. Interestingly, it is again sight range 3 where a team first beats another with higher sight range.

Examining the charts of the higher-sight range teams, it would appear that they reliably beat teams 1, 2, and 3 in slightly decreasing rates of effectiveness. However, at sight range from 4 to 6, it seems much less consistent. This would imply, perhaps, that there *still is* a cutoff point for this tournament as well, but at the sight-range 4 mark, rather than 3.

5.3 General Trends

Between both tournaments, very high and very low sight-ranges were the worst performers. This observation neatly lines up with the intuitions following the statement of the hypothesis, much earlier in the paper. For intuition does lead to the idea that emergent behavior requires some kind of “middle ground” as far as information is concerned — to little and there is no cooperation, too much and they would all act as individuals. However, the higher-sight-range teams for tournament 1 performed remarkably badly in part, I think, simply because of the way the matches were set up.

In tournament 2, with completely random placement, the differences in efficacy between teams were smoother. That single observation leads me to think that random-placement is the way to proceed studying this particular domain. I will expand upon such thoughts in the conclusion section.

5.4 Hypothesis — Verified or Falsified?

Finally, what is the verdict concerning the hypothesis? Sadly, I must say that it is mostly **falsified**. The reason I say falsified is because the trends in tournament 2 are simply too smooth to present a sharp cutoff where one can point and say “here is where emergent behavior began”. Tournament one did have that to some degree with sight range 3, but it does not seem sharp enough to counteract the evidence from tournament two. But I say “mostly” falsified because the data sets gathered here are very small. I did not have enough time to run more, but several dozen more tournaments each would create truly definitive trends.

6 Conclusions

Even with my hypothesis falsified, the idea of exploring the boundary between emergent and non-emergent behavior is key to understanding emergent behavior. These experiments probably have disproved my initial thoughts on how emergent behavior arises, but they also raise many more questions, thoughts, and paths of investigation.

6.1 In What Way Does Information Promote Emergent Behavior?

These experiments seemed to show that there is not a sharp cutoff for efficacy of emergent behaviors as a function of information, but rather a trend. This just raises the question: what kind of role *does* information play with efficacy of emergent systems? Does comparative efficacy increase linearly? Polynomially? Or in some terribly unpredictable pattern?

6.2 Measuring Coherence

This paper has measured emergence by simply whichever team won, or did better, in a given simulation. But there are other ways of measuring emergent behavior, one of which could be how “coherent” the teams were. How often did the agents react in concert? How infrequently did they have counteracting behaviors? This idea presents an entirely different way of analyzing the results of these experiments, and could perhaps shed new light on the matter. Sadly, time constraints prohibited me from being able to fully analyze the effect of coherence in a quantitative fashion. Reviewing the recordings of a few simulations, however, it is pretty clear that larger sight range encourages coherence, but to what degree is the fascinating part, and the part that would require truly in-depth analysis. I am unaware if there is even an established methodology for measuring coherence.

6.3 The Idea of a Competitive Domain

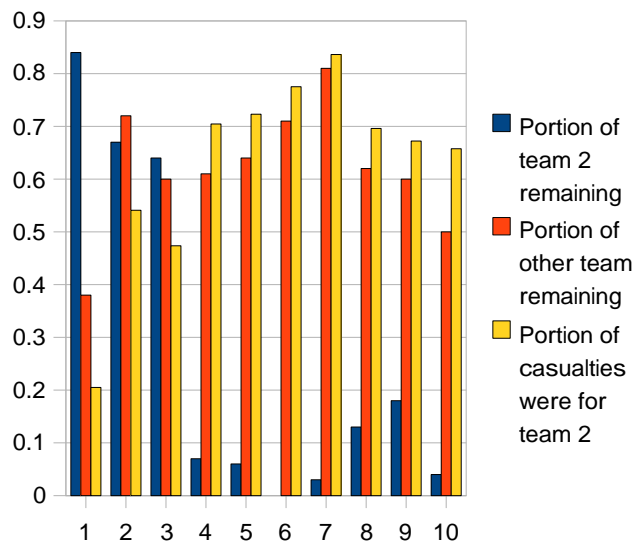
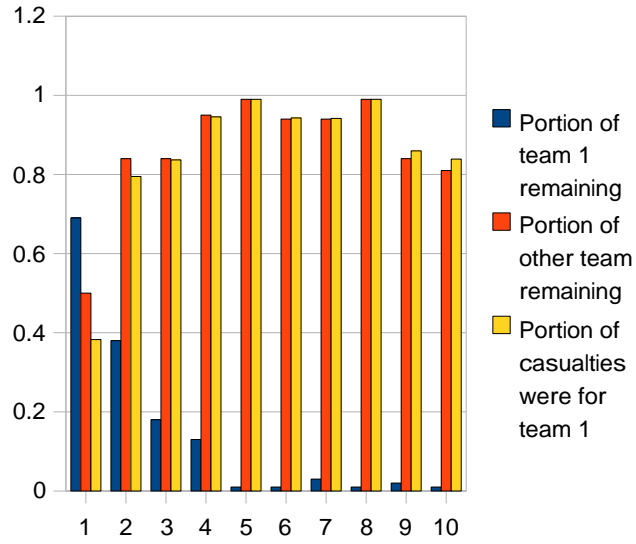
This paper also used a competitive, rather than single-team domain. Rather than trying to get a herding effect or flocking effect, I investigated how different teams of rivals would attempt to defeat one another. As stated in the introduction, I believe further research in this area is imperative for understanding more on how emergent behavior works, and also for *producing* emergent systems that really do what we want in fighting disease, or exploring dangerous terrain. This paper mainly used the competitive domain for ease of analysis — I let the winner tell me who was more “emergent” than the other. But understanding these interactions between two emergent systems, I think, is a path worth following in its own right.

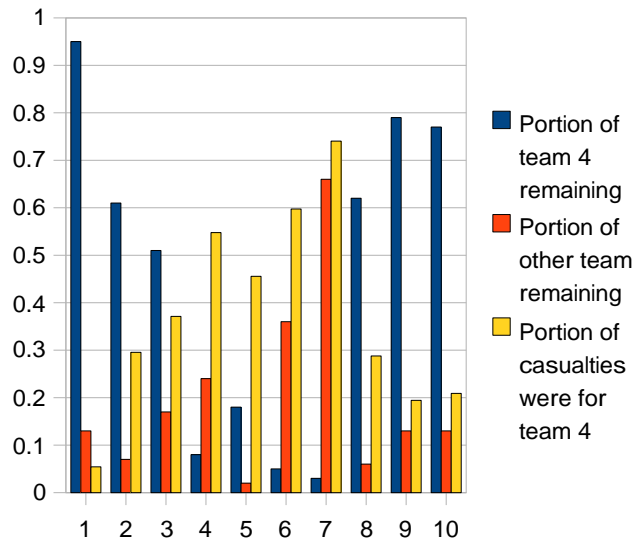
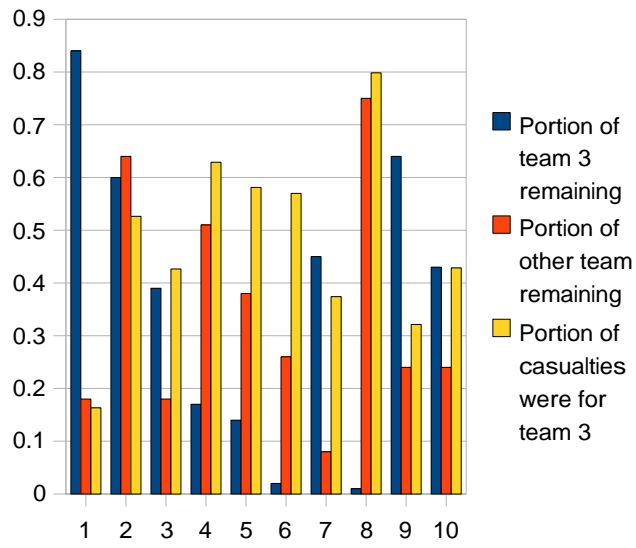
6.4 Looking Forward

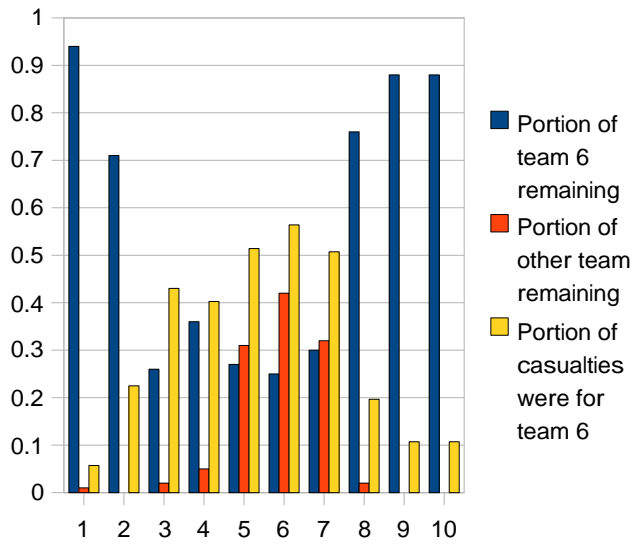
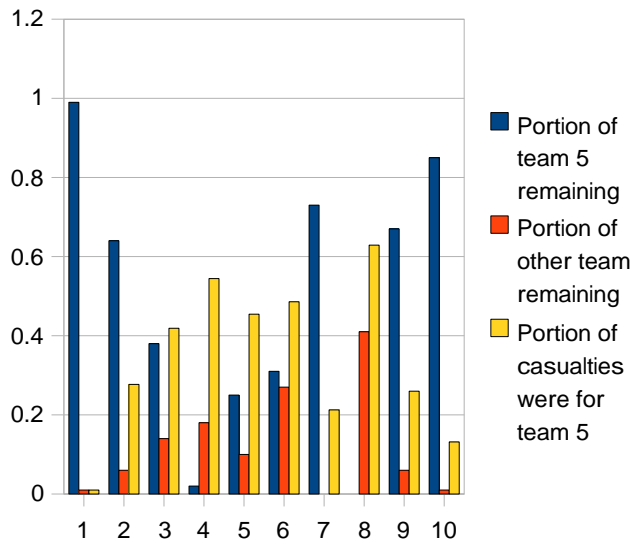
I believe all of those are ideas worth pursuing in the study of emergent behavior. More immediately, I believe the next steps for this research is to first simply run more test, and establish a real solid base of numbers. The next way to progress this particular research is to deal with the manually-set policy vectors. As hinted at in the results and discussion, it may be that different sight ranges require different policy vectors. I do not think that accounts for all the differences in the results, but I believe that automating a way to at least search for better policy vectors for each sight range would be fruitful. It is almost impossible, but to establish a few competitions where one knows that the “best possible policy vector” is in use for each side would be ideal, and stretching towards that goal is worthwhile.

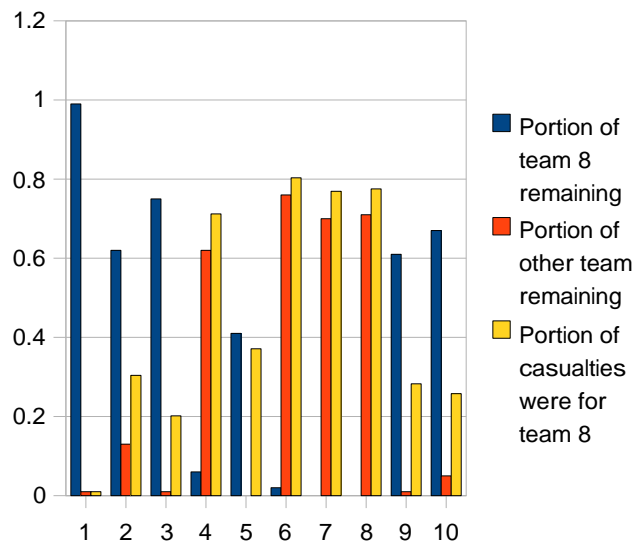
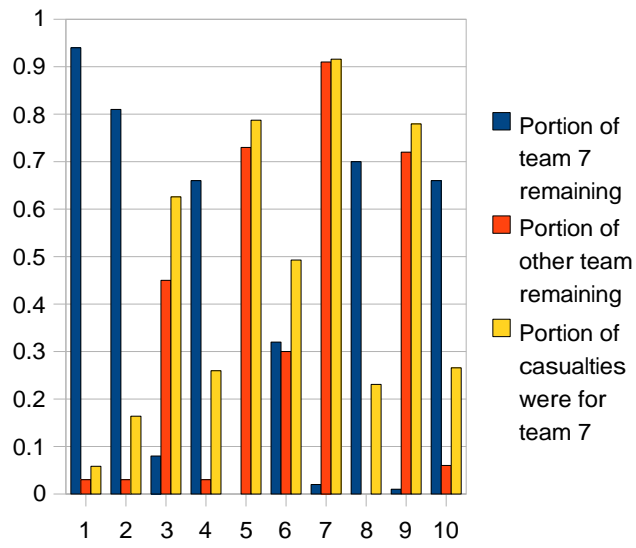
In summary, the results in this paper have not even scratched the surface of what is left to find out. More experiments, better experiments, and larger experiments will all hopefully reveal more information about this fascinating field. Certainly, every aspect of emergent behavior is fascinating. Hopefully new research will tell us more about this fascinating phenomenon.

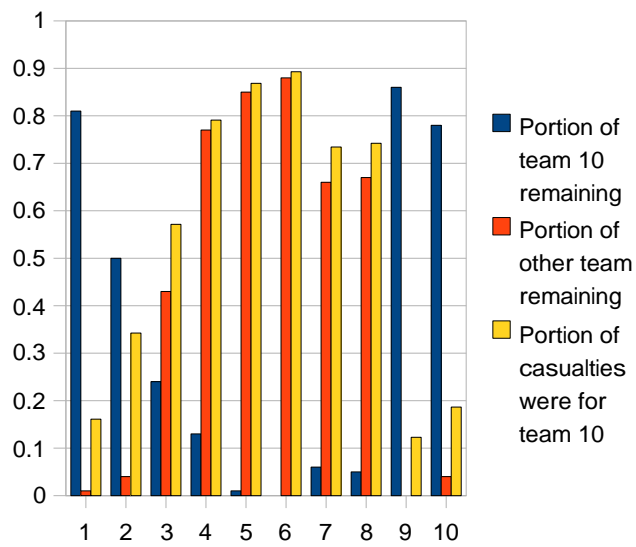
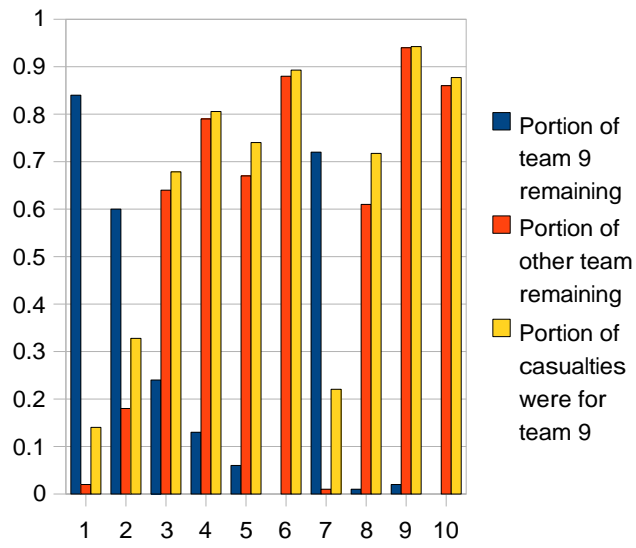
A Figures for Tournament 1



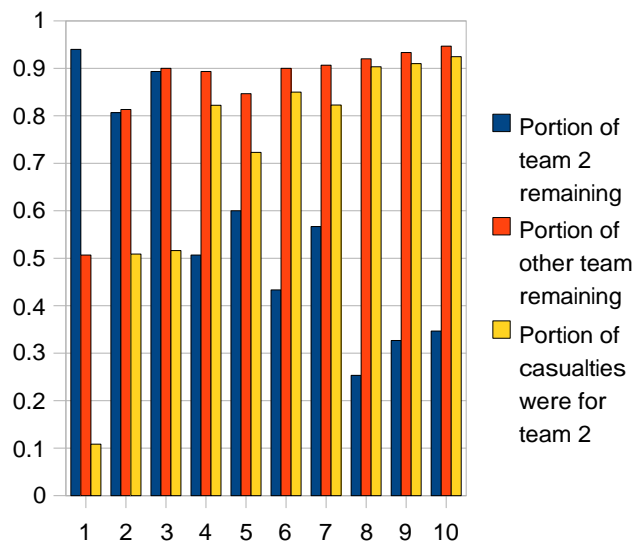
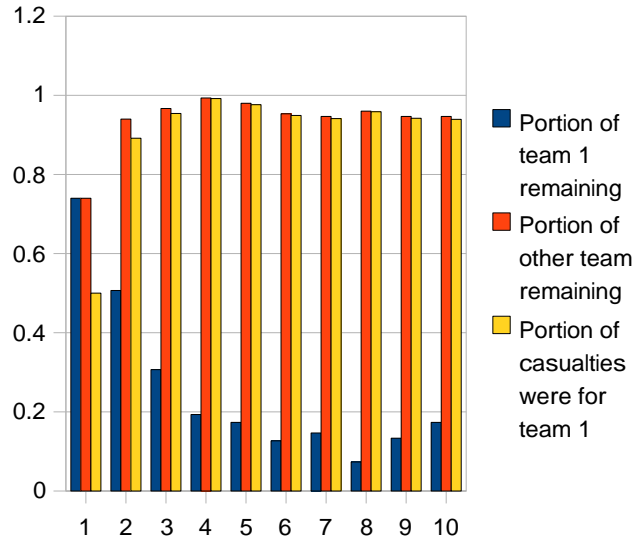


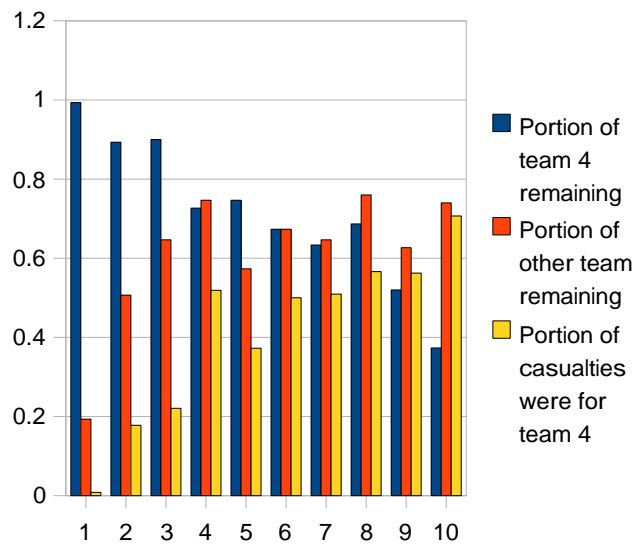
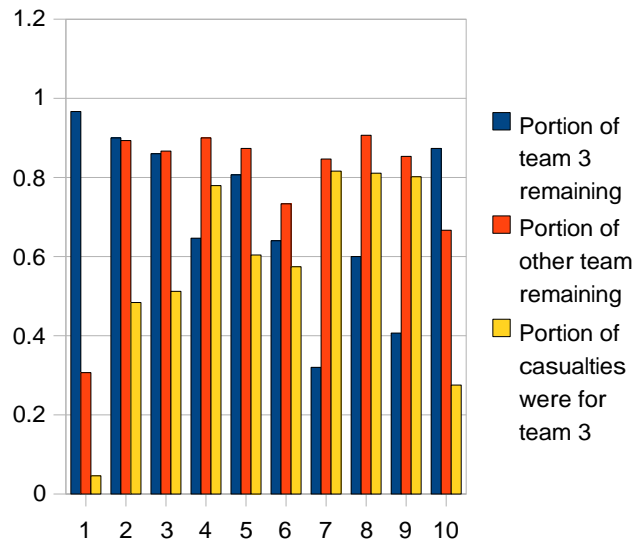


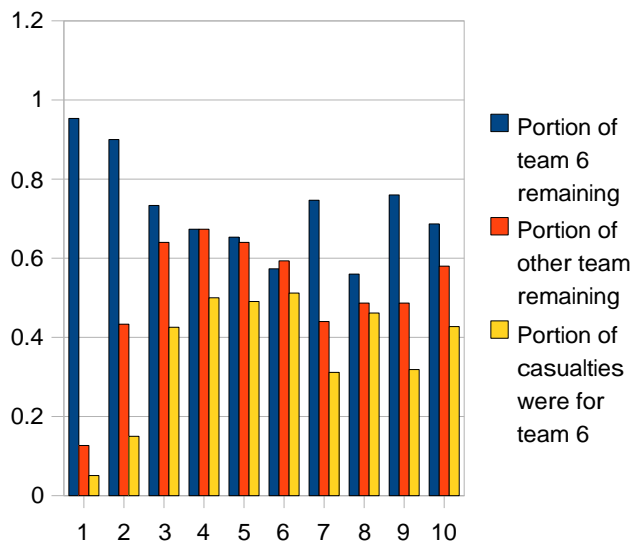
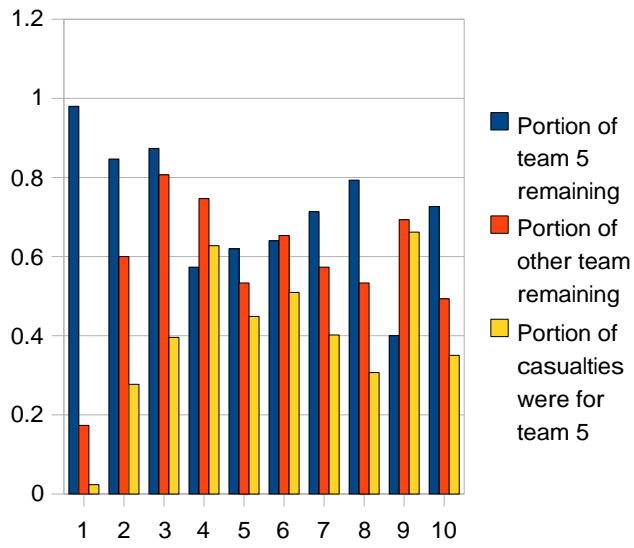


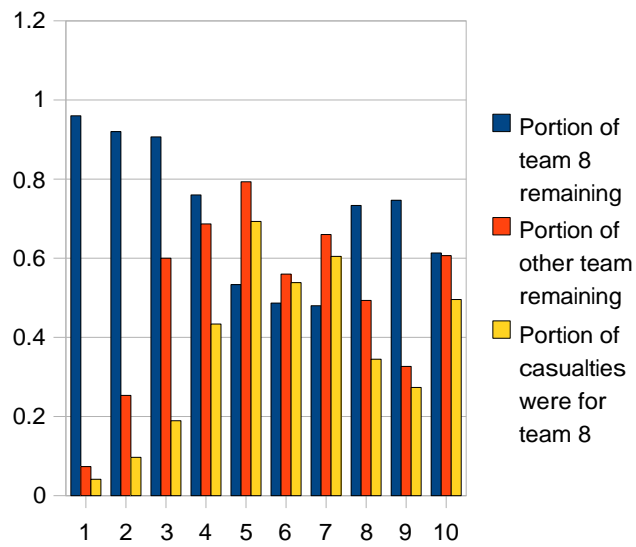
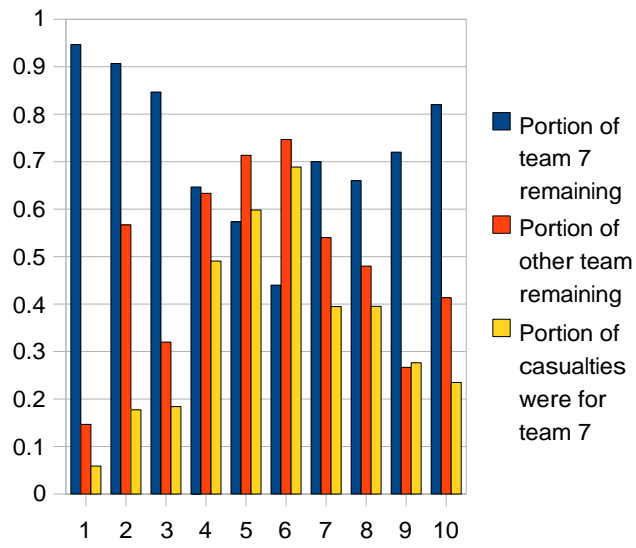


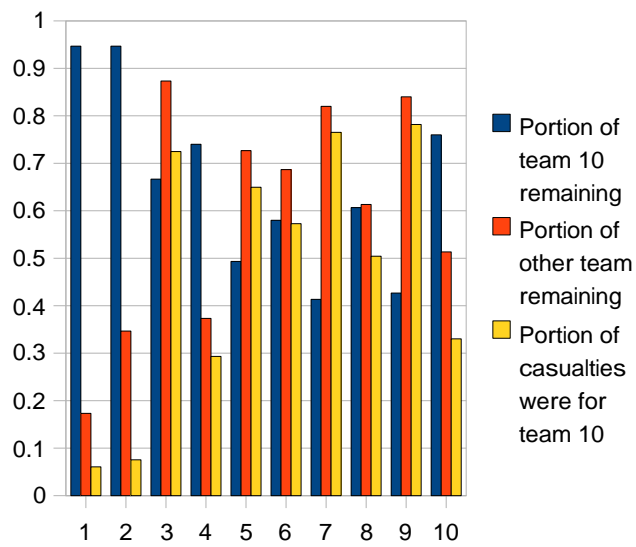
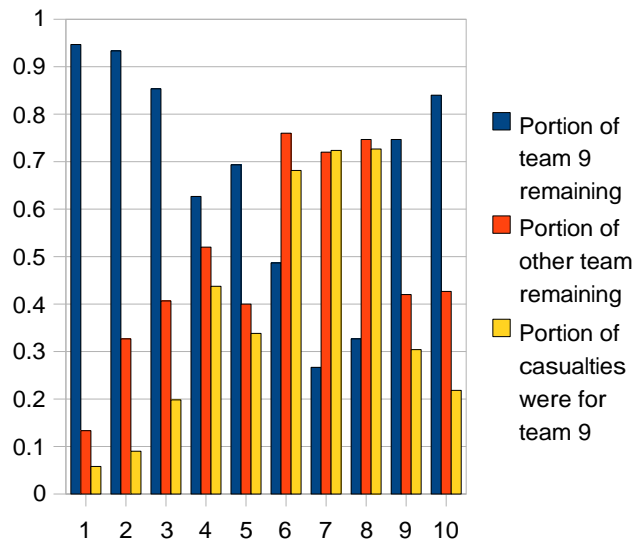
B Figures for Tournament 2











References

- [BMM94] Rodney A. Brooks, Frederic R. Morgenthaler, and Maja J Mataric. Interaction and intelligent behavior. Technical report, MIT EECS PhD Thesis AITR-1495, MIT AI Lab, 1994.
- [Gor09] Aaron Gorenstein. CSC 242 Term Project Proposal. Project Proposal for Research in Emergent Behavior, April 2009.
- [HK04] C.Scogings H.A.James and K.A.Hawick. A Framework and Simulation Engine for Studying Artificial Life. 6(ISSN 1175-2777):143–155, May 2004.
- [Hol07] O. Thomas Holland. Taxonomy for the Modeling and Simulation of Emergent Behavior Systems. In *SpringSim '07: Proceedings of the 2007 spring simulation multiconference*, pages 28–35, San Diego, CA, USA, 2007. Society for Computer Simulation International.
- [HRRT05] M. G. Hinchey, C. A. Rouff, J. L. Rash, and W. F. Truskowski. Requirements of an integrated formal method for intelligent swarms. In *FMICS '05: Proceedings of the 10th international workshop on Formal methods for industrial critical systems*, pages 125–133, New York, NY, USA, 2005. ACM.
- [Ila94] Andy Ilachinski. Towards a Science of Experimental Complexity: An Artificial-Life Approach to Modeling Warfare. Technical report, Presented at 5 th Experimental Chaos Conference, 1994.
- [LSL06] Zhengping Li, Cheng Hwee Sim, and M.Y.H. Low. A Survey of Emergent Behavior and Its Impacts in Agent-based Systems. pages 1295–1300, Aug. 2006.
- [Mat92] M. Mataric. Designing Emergent Behaviors: From Local Interactions to Collective Intelligence. In *In Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, volume 2, pages 432–441, 1992.
- [Mog06] Jeffrey C. Mogul. Emergent (mis)behavior vs. complex software systems. In *EuroSys '06: Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, pages 293–304, New York, NY, USA, 2006. ACM.
- [Nit05] Geoff Nitschke. Emergence of cooperation: State of the art. *Artif. Life*, 11(3):367–396, 2005.
- [Par97] H. Van Dyke Parunak. “Go to the Ant”: Engineering Principles from Natural Agent Systems. *Annals of Operation Research*, 75:69–101, 1997.

- [PMKD02] M. Privosnik, M. Marolt, A. Kavcic, and S. Divjak. Evolutionary Construction of Emergent Properties in Multi-Agent Systems. pages 327–330, 2002.
- [RVH⁺04a] Christopher Rouff, Amy Vanderbilt, Mike Hinchey, Walt Truszkowski, and James Rash. Properties of a formal method for prediction of emergent behaviors in swarm-based systems. In *SEFM '04: Proceedings of the Software Engineering and Formal Methods, Second International Conference*, pages 24–33, Washington, DC, USA, 2004. IEEE Computer Society.
- [RVH⁺04b] Christopher Rouff, Amy Vanderbilt, Mike Hinchey, Walt Truszkowski, and James Rash. Verification of emergent behaviors in swarm-based systems. *Engineering of Computer-Based Systems, IEEE International Conference on the*, 0:443, 2004.
- [SBB01] D. J. T. Sumpter, G. B. Blanchard, and D. S. Broomhead. Ants and Agents: a Process Algebra Approach to Modelling Ant Colony Behaviour. *Bulletin of Mathematical Biology*, 63:951–980, 2001.
- [TW04] Seth Tisue and Uri Wilensky. Netlogo: A simple environment for modeling complexity. 2004.
- [VVPV97] H. Van, Raymond S. VanderBok, H. Van Dyke Parunak, and Ph. D. Raymond S. V. Managing Emergent Behavior in Distributed Control Systems. In *In Proceedings of ISA Tech '97, Instrument Society of America*, page 97, 1997.
- [WZ07] Shufeng Wang and Hong Zhu. An Experimental Study of the Emergent Behaviors of Self-Organized Agent Communities. pages 3239–3246, Sept. 2007.