Destiny or Luck: What Causes Virtual Tribes to Outlast their Competitors?

Lucas Nunno
University of New Mexico
Computer Science Department
Albuquerque, New Mexico
lnunno@cs.unm.edu

Abstract

When civilizations begin to grow, their access to resources can have a great effect on determining their survival. This study uses the Movable Feast Machine to analyze the competition of hostile tribes in an environment with limited resources. A simple model of population growth, tribal identities, and conflict is used to investigate what can make a population survive longer than its genetically identical competitors. We have discovered emergent complex behavior from a set of simple rules. Tribes' initial positions and their access to resources were varied to analyze the effect that controlling resources has on their survival. We have discovered several configurations that appear asymmetrical but yield surprising results; where intuition would expect a tribe to fail, but instead it flourishes.

Introduction

Background and Motivation

There has been much research in the area that attempts to explain why certain civilizations prosper while others fail. Diamond (1997), for example, famously suggested that Eurasian civilizations' success was not due so much to their ingenuity over the New Guinean population as it was a product of opportunity and necessity. This is in a large part due to their access to resources, domestication of animals and early exposure to viruses such as smallpox that held a major advantage when venturing to the Americas for the first time. While an isolated New Guinean people did not have the same opportunities as their Eurasian counterparts, therefore not making as many technological strides as other civilizations on the globe that had many interactions with other early neighbors and access to a variety of rich resources.

It is with *Guns, Germs and Steel* and research of this type that one imagines how this can applied to artificial life. Often in the field of artificial life genetics is emphasized over physical location in the natural world. In some instances we see physical spatial relationships completely disregarded over a complex genome such as in the Avida artificial life work as discussed by Lenski (2003). In these artificial life simulations that emphasize genetics over physical location and spatial concerns, it loses a critical aspect of the real world. This may be fine for studies that wish to emulate millions of years of evolution, but doing a study on a relatively short period of time requires that the physical laws of nature be obeyed and creatures that are spatially located are the only ones who can interact.

This paper forgoes the idea that this has to do with genetic traits and instead investigates what role resources play in the success of tribes. Although genetics do obviously play an important part in the evolution of species, here we focus on a time scale that is small enough not to be melded by the hands of evolution, but instead the environment plays a large part in a group's success.

The model presented does not aim to exactly emulate the way which civilizations work, instead we define tribes as very simple life forms that perform very few functions including: moving, reproducing, and killing.

Our work shares similar ideas presented in the artificial chemistry done by Fontana and Buss (1994) and Hutton (2007); however this research is working on a different timescale and focuses on the macro level. We are not presenting an artificial chemistry, but we do have the perspective of looking at the initial spark of human societies and seeing how changing the initial conditions of the world effect the results and which tribes survive. We will often refer to members in particular tribes as ”atoms” this is meant to be consistent with the vocabulary of the Movable Feast Machine, as described in the next section. For an in-depth look on tribal warfare one can reference Gat (2006) to see the intricacies of early human warfare. However, we will be analyzing a simplified model.

Modeling an entire civilization is extremely complex and requires a large number of independent variables. Civilization simulation can require many complex subsystems including modeling population densities, specialized workers, and other dynamic interactions within populations. In our simulation we collapse the complex behavior into the Base element, which one can imagine internally represents a more complex ecosystem and is represented here as a collection of points on the grid. We can imagine the Base represents a roving tribe that can collectively produce new members of

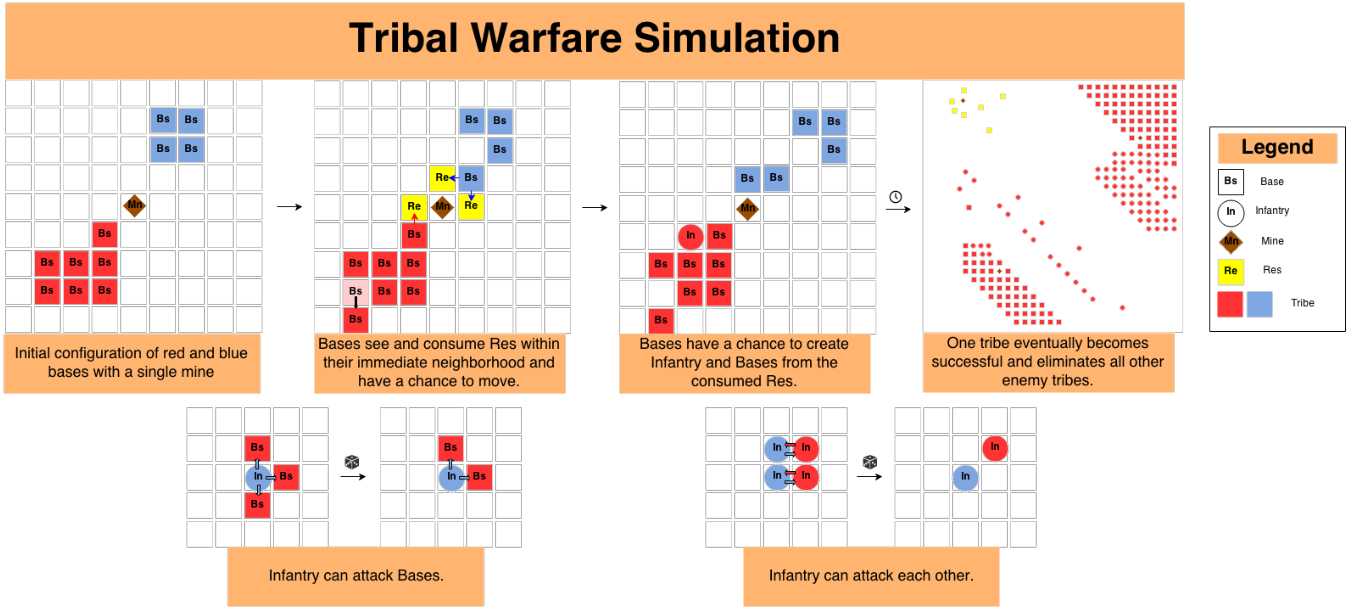

Figure 1: Model summary. **(Top):** An illustration of tribe growth and conflict. Tribe bases collect resources and create infantry units, these infantry units kill members of foreign tribes. The last frame depicts a zoomed-out possible end state where the red tribe grows and takes over a large portion of the world. **(Bottom):** An illustration of the infantry interaction between different tribes. Note that there is a probability to kill another unit as depicted by the dice graphic.

its population given adequate resources.

Using the Movable Feast Machine as an Artificial Life Platform

The research and experiments described here is performed on the Movable Feast Machine (MFM), an indefinitely scalable asynchronous cellular automata platform. See the full paper by Ackley (2013a) for a full description of the MFM. Figure 2 is an illustration from Ackley (2013a) that illustrates how the MFM works, with a description of the event window, elements, and atoms; all of which that will be referenced in the work here.

We used the MFM simulator implemented by Ackley and Small (2014) and added our own elements to explore tribal behavior. One of the key takeaways with using the MFM is that it is nondeterministic, so one of our sites in the grid (atoms) may receive an event in an indeterminate time in the future, however, stochastically it should roughly have the same amount of events per site. This is not strictly or globally enforced however, and is a design trade-off made to make the indefinitely scalable nature of this architecture possible.

See figure 1 for an outline of how the elements we've created interact in the MFM.

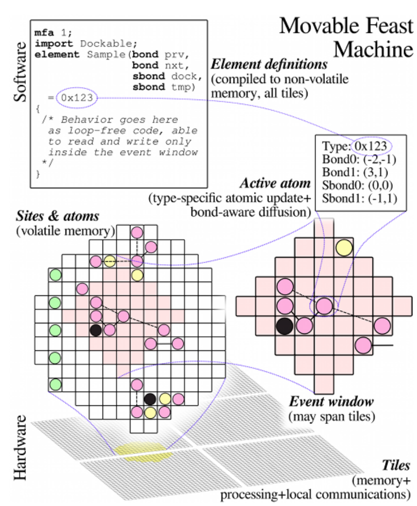

Figure 2: Architectural overview of the movable feast machine used by permission from Ackley (2013a).

Model Description

The model of tribal survival that we present is defined by the following elements:

1. Abstract Tribal Element
2. Mine
3. Base
4. Infantry

These elements simulate tribal identities, their resources, and the conflict between hostile groups.

The parameters described are variable by the virtue of the MFM, but the experiments run keep these parameters constant. We have chosen instead to vary the locations of tribes' Bases and Mine atoms. The default parameter values were chosen after prior experimentation and yielded the most interesting preliminary results.

Resource Elements

The Res Element The Res element represents available resources in the environment. It was defined previously in the initial development in the MFM by Ackley (2013a). It diffuses through the environment and can be consumed by Bases. Bases turn Res into gold at the rate of *Gold Per Res* (See Table 2).

The Mine Element The Mine element generates resources (Res) that can be used by any tribe.

Table 1 is a summary of the element parameters that a Mine has, these are constant through the simulations performed and give an idea of how each individual Mine atom works.

Parameter	**Description**	**Value**
Res Spawn Odds	The probability that a mine will spawn a Res.	$\frac{1}{5}$
Exhaustion Rate	The rate at which a mine gets exhausted. This decreases the probability that a Res will be spawned.	$\frac{1}{100}$

Table 1: Parameters for the Mine Element.

If the space chosen to produce a Res is occupied by a Base atom, the Mine increments the amount of resources in the Base by G_{pr}, which is identical to the behavior a Base has when it collects Res.

Tribal Elements

The following elements mentioned in this paper all extend Abstract Tribal Element and are therefore called *tribal* elements in this paper:

1. The Base Element
2. The Infantry Element

The Abstract Tribal Element The Abstract Tribal Element describes properties that all tribal elements share: the ability to identify oneself as belonging to a particular tribe and to detect if another atom is a member of a tribe. It defines functionality for tribal atoms to detect if other neighboring atoms are tribal and if they share the same tribe as itself.

There are four bits allocated in each tribal atom (as shown in figure 3) that contain the tribe number that the atom belongs to. So there are $2^4 = 16$ possible tribes, however we will only be conducting experiments with two tribes at a time.

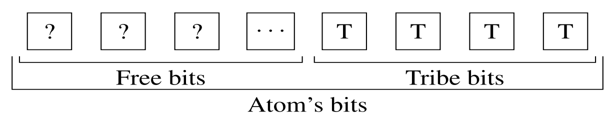

Figure 3: A Tribal Atom's Internal State. Allocates the four right-most bits to identify tribe (T). The rest are free bits (?) to be allocated by inheriting elements.

When Tribal atoms are created their tribe bits are set based on the tribe parameter that is set in the simulation. This parameter is dynamic, so units of different tribes can be created by changing the tribe parameter of that element. Note that the Base and Infantry elements have an implicit tribe parameter by the virtue of being an Abstract Tribal Element.

Tribal elements also have a notion of *element gradient*. This is to differentiate tribal elements with varying shades of their base tribe's color. It is defined as follows. Tribes' base colors are usually a pure shade of that color, for example the starting shade of the Red team is `0xFF880000` in the ARGB color space. Let the base color of a tribe be denoted as C_T and the gradient for the element be defined as G. Therefore the color of a tribal element is:

$$\text{Color} = C_T + G$$

The Base Element Base atoms are responsible for creating all tribal atoms. Base atoms can create additional Infantry and Base atoms. There are two phases that each Base enters during an event that it receives:

1. Collection phase
2. Creation phase

Elements of a Base's behavior is also determined by the element parameters as described in Table 2. When a Base atom receives an event, it first enters the collection phase. The first step of the creation phase is performed by scanning its event window and counting the number of friendly bases f_b. Base atoms are spatially aware of other Bases of the same tribe and have a preference to stay where they are if

Parameter	Description	Value
Gold Per Res (G_{pr})	The amount of gold produced for each Res collected.	5
Base Gold Cost (C_B)	The cost (in gold) of producing a base.	5
Base Create Odds (P_B)	The odds that a base will be attempted to be created.	$\frac{1}{5}$
Infantry Gold Cost (C_I)	The cost (in gold) of producing an infantry unit.	1
Infantry Create Odds (P_I)	The odds that an infantry unit will be attempted to be created.	$\frac{1}{3}$
Base Stability (S_b)	The factor that nearby Bases have on influencing a Base to stay where it is.	10

Table 2: Parameters for the Base Element.

there are friendly Bases surrounding it. The odds of the atom moving to a new location (C_m) are

$$C_m = \frac{1}{S_b \cdot f_b}$$

Then, it looks for Res in its immediate neighborhood. If it sees Res, it consumes the Res and increments its internal gold counter by G_{pr} the internal allocation of bits can be seen in figure 4. This ends the Base atom's collection phase.

After the collection phase has completed, the Base atom enters the creation phase. First, the Base looks for an empty space in its immediate neighborhood to create a unit. If there are no available empty spaces the creation phase ends. If we have found an available empty, we then do a create check for a Base with probability of P_B. If this check passes we then check our internal gold counter G_C. If $G_C \leq C_B$ we create a Base and place it in the empty space that we have previously found. We then repeat the creation process for Infantry.

$\boxed{?}\ \boxed{?}\ \boxed{?}\ \boxed{\cdots}\ \boxed{\text{G}}\ \boxed{\text{G}}\ \boxed{\cdots}\ \boxed{\text{T}}\ \boxed{\text{T}}\ \boxed{\text{T}}\ \boxed{\text{T}}$

Figure 4: A Base Atom's Internal State. There are eight gold (G) bits, four tribe bits (T), and an unspecified number of remaining free bits (?).

The Infantry Element The Infantry Element is responsible for attacking other tribes. This unit actively attempts to prevent other tribes from expanding and taking available resources by attempting to "kill" enemy units. Each Infantry unit has a P_k chance to delete an enemy unit from its MFM cell for every enemy unit detected within its immediate one cell neighborhood. This behavior and its movement is determined by the parameters described in Table 3.

Parameter	Description	Value
Direction Change Odds (P_{dc})	The probability that this infantry unit will change its movement direction.	$\frac{1}{10}$
Kill Odds (P_k)	The probability that this infantry unit will kill an enemy unit.	$\frac{1}{4}$

Table 3: Parameters for the Infantry Element.

Infantry units move every event they receive in the direction that they store internally. Infantry units initialize by setting their movement to an arbitrary direction (set as East) and have P_{dc} chance of changing their movement to another random direction on every event they receive. The internal allocation of bits for the tribe atom can be seen in figure 5.

The number of direction bits needed is calculated by the following formula

$$|\text{Dirs}| \gg 1 - 1 = 8 \gg 1 - 1 = 3$$

Which in this implementation we have eight possible directions: the four cardinal directions and the four diagonal directions. Therefore, the result is as shown.

$\boxed{?}\ \boxed{?}\ \boxed{?}\ \boxed{\cdots}\ \boxed{\text{D}}\ \boxed{\text{D}}\ \boxed{\text{D}}\ \boxed{\text{T}}\ \boxed{\text{T}}\ \boxed{\text{T}}\ \boxed{\text{T}}$

Figure 5: An Infantry Atom's Internal State. There are three direction (D) bits, four tribe bits (T), and an unspecified number of remaining free bits (?).

Results

Self-healing Tribes Experiment

The MFM is a framework that is centered around robust computation. Therefore, it is vital to create elements within this framework that have self-healing properties. One way that we have done this in this research is by allowing the Base element to reproduce.

In this experiment we have an initial configuration of a radius five cluster of red bases on the left of a medium-sized MFM grid and an identical cluster of DReg mirrored on the right.

We have modified the parameters of the Base element slightly to make creating Infantry atoms a little more unlikely since they do not promote Base self-healing. This deviates from the parameters as described in Table 2 for this

Parameter	Description	Value
Infantry Gold Cost	The cost (in gold) of producing an infantry unit.	10
Infantry Create Odds	The odds that an infantry unit will be attempted to be created.	$\frac{1}{10}$

Table 4: Modified parameters for the self-healing experiment.

experiment only, because that is typically closer to the desired behavior and creates an appropriate ratio of Infantry units to Base units. However, there is still a chance for them to be created. They often get deleted immediately by the DReg in the environment. This configuration and the end state can be seen in figure 6.

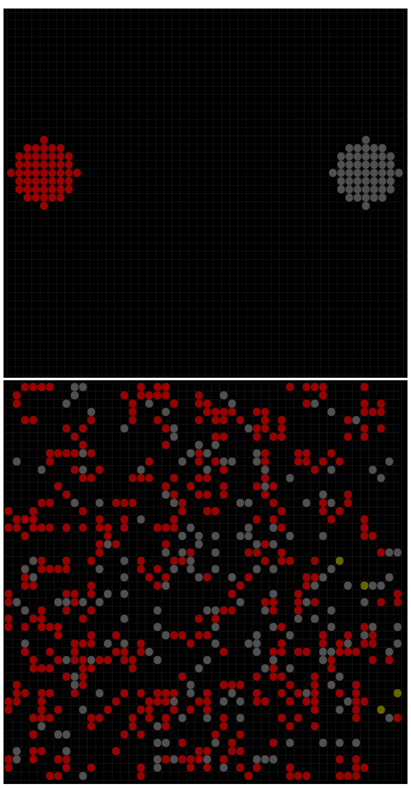

Figure 6: Beginning and end states of the self-healing experiment. **(Top):** Base and DRegs (red, left; grey, right). **(Bottom):** The settled steady state of Base, DReg, and Res.

Figure 7 shows the results of the self-healing experiment. Notice that the Base population remains constant for a number of AEPS until Res has diffused to the Bases' location. Then we can see an exponential increase of Base growth that corresponds with the consumption of Res. We can then see

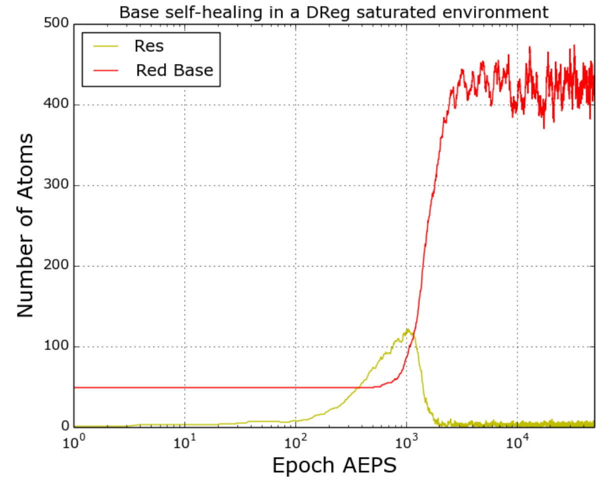

Figure 7: Demonstration of the self-healing property of Base atoms.

that the Base population stays relatively constant with perturbations from the DRegs in the environment that continually destroy the Base atoms in the environment. However, the Base element is able to reproduce from the emitted Res in the environment and is therefore able to heal itself.

Self-healing and assembly is an important property to maintain when designing robust cellular automata and is a corner stone for the work that is done on the MFM. Therefore, we should be satisfied to see that out element has a self-healing property before continuing further experiments.

Symmetric and Asymmetric Experiments

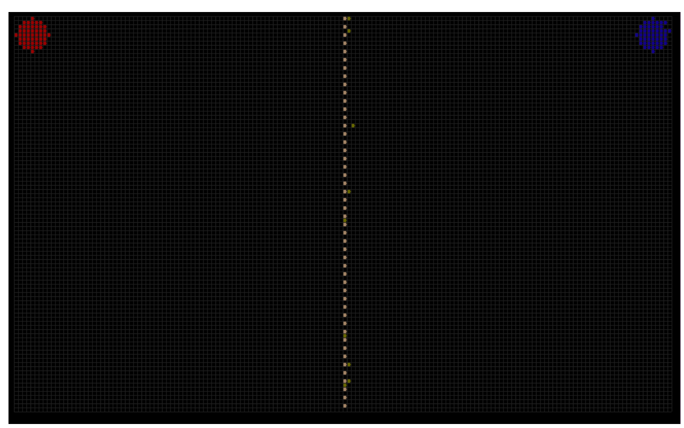

Figure 8: Start state for the symmetric experiment after a few initial time steps. Red (top-left) and blue (top-right) Bases begin to diffuse and collect resources.

There are a number of symmetric configurations of tribes and resources that we could hypothesize each tribe should have an equal chance of succeeding. In the real world, it is difficult to imagine a scenario where the distribution of resources is symmetrical between two competing tribes. However, it may be useful to analyze scenarios that are symmetrical in order to determine which setups are beneficial to a particular tribe. Our symmetrical configuration can be seen in figure 8.

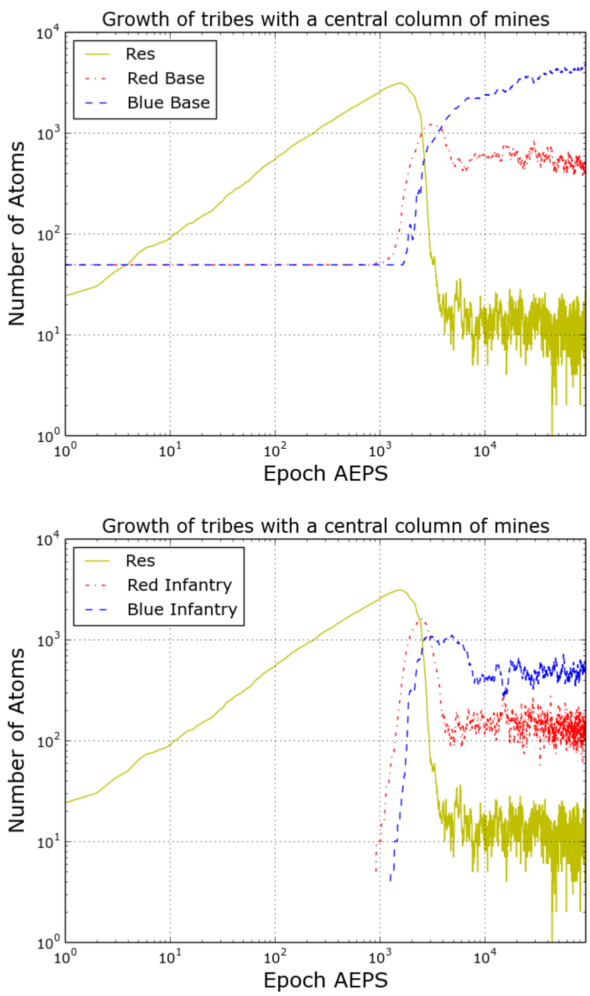

Figure 9: A particularly interesting symmetric experiment run. The Red bases expand early but the Blue bases end up being more successful. Notice how there is sustained fighting for upwards of 90kAEPS. A video of this run is available at Nunno (2014b)

Some of the more interesting behavior occurs when two or more tribes are in a stable state; producing an amount of infantry that does not completely wipe out the other competing tribes (results seen in figure 9).

Typically, we see a tribe's Base atom diffuse to a cache of Res atoms where a large increase in that tribe's population occurs. We can see a run with this behavior in figure 10.

The experiment is run ten times with each Mine column x position specified either for 10kAEPS or when one tribe

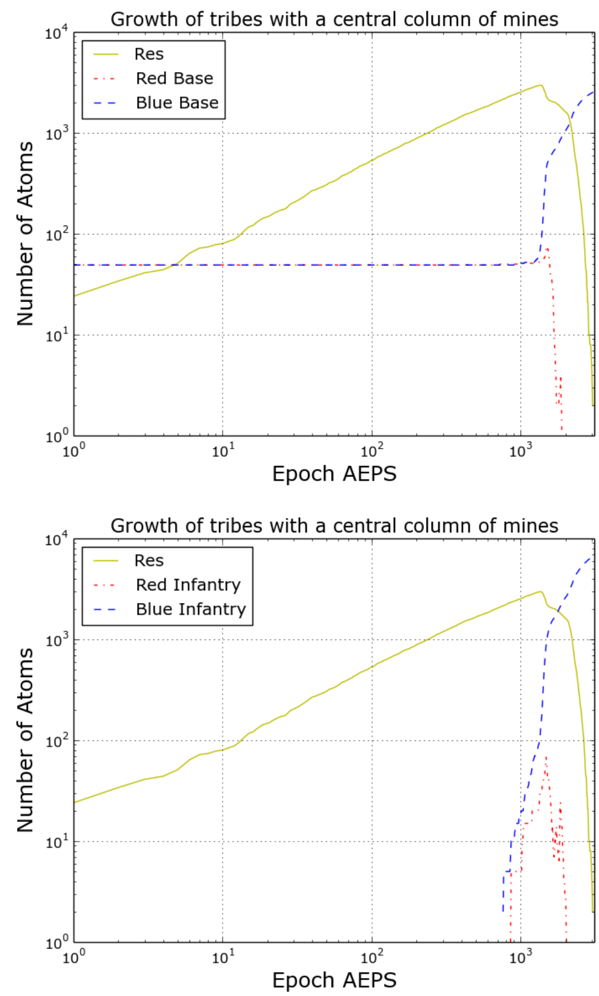

Figure 10: A more typical symmetric run. Blue finds the resources early and quickly expands, winning the simulation in 3kAEPS. A video of this run is available at Nunno (2014a)

has been completely eliminated, whichever occurs first. The score for each tribe is calculated as follows at the end of the run:

- **1 point:** Completely eliminated the other tribe.
- **0.75 points:** Having more Base atoms then the other tribe at 10kAEPS.
- **0.25 points:** Having some Bases at the end of 10kAEPS, but losing to the other tribe.
- **0 points:** A loss.

The results using this scoring system can be seen in figure 11. It may not be surprising that tribes that are closer to the Mine column are more successful, but we did see some interesting emergent complex behavior that may be unexpected. Particularly, note that slight shifts in the Mine column's x position did not have as large of an effect. A tribe could still become successful even if its competitor was at a slight advantage.

Discussion

Many more tribal elements were proposed, but only the Base and Infantry elements were added for the sake of simplicity. The core behavior of the simulation revolves around the interaction between Base and Infantry atoms of different tribes.

The Abstract Tribal element is generic enough so any other element can extend it, the only requirement is that it uses a set amount of bits to denote its tribe.

Bases with a small number of neighboring friendly Bases are more likely to move than Bases that are surrounded by many other friendly Bases. This is an attempt to have Bases self-assemble and form large clusters, but much of this depends on luck. These clusters are often larger than an event window, so there must be some chance for atoms that are in clusters to break free and seek out even larger clusters of friendly bases. See the work conducted by Ruiz (2014) for more work done on the MFM dealing with self-assembly of large structures.

For a sense of community to be gained in the MFM, we believe that additional work will be needed in the areas of communication protocols and self-organizing behavior.

We have also seen potential issues in the inherent spatial nature of the MFM. Often in our simulation there arises the issue where a tribe will expand rapidly and form clusters around the Res producing Mines. This often has the implication where the Bases with the most gold are unable to create units because there is a lack of open space. This may be a larger issue with discrete cellular automata, however; and it's difficult to say how this effects the results.

The research described in this paper highlights the need for a native communication protocol in the MFM. If the tribes as described in this paper were capable of communication, we could potentially see emerging complex behavior that is much more coordinated and orderly. We have shown that finding the initial Res production locations are absolutely critical for a tribe's success. If this could be communicated to members of the other tribe, we could see a more intelligent organization of a tribe's manpower. The work in Stallings (2014) is promising for the type of results that we wish for. The model of pheromone-like trails seems to be the most powerful communication possible in a machine that is inherently local with no global information possible due to the nondeterministic and indefinitely scalable nature.

The experiments run show a base case for the success of tribes, it is easy to imagine that there are a number of grid configurations that can be made to study the effect of the resource distance to the tribe's success. We have considered experimenting with a Mine grid that shifts in both the x and y dimensions to see how the bias in two dimensions effects the results, but this is left to future work. We have also considered adding more tribes to the experiments, but determined in these early tests that simple is better and adding more tribes would make it difficult to analyze the effect of varying the independent variable has over the stochastic nature of the machine.

Another experiment that has the potential for future work is varying Infantry behavior and determining the effect that it has on a tribe's success. The Infantry direction change odds P_{dc} parameter has some interesting dynamics when it is varied. Particularly when it is set very low, Infantry units will explore as far as they can in a single direction. This changes the results fairly dramatically since the Infantry units are quite bold and tend to destroy an enemy Base at all costs, while abandoning their own home.

Conclusion

We have discovered a number of non-trivial resource production initial locations that promote the growth of certain hostile tribes over others. We have shown that shifting a central column of Mines greatly effects a tribe's success. This may be somewhat obvious, but it does indeed seem to show that there is some notion of destiny over luck. The tribes that are closest to resources are more likely to succeed. In a stochastic environment such as the MFM this may be the most we can hope for; that the averages are indicative of greater underlying mechanisms.

Furthermore, the Movable Feast Machine provided a platform to easily explore the effects of initial conditions in tribal growth and fitness. The definition of elements that can identify and attack intruders has numerous applications in security, biology, and other fields. These fields can take advantage of the work done in this paper and extend it to many other areas of research that need a definition of self versus an outside population.

We believe that the work presented in this paper bene-

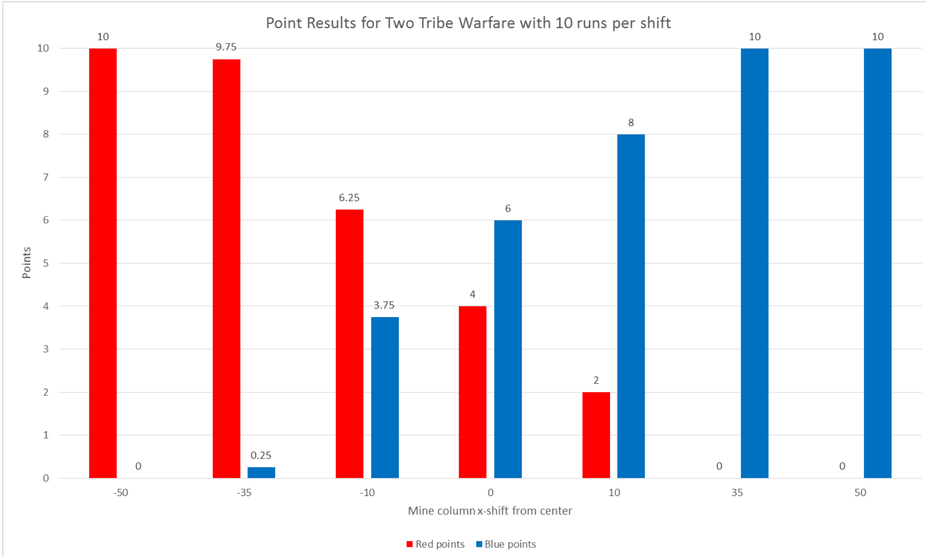

Figure 11: Summary of runs with shifting the x position of the Mine column.

fits the MFM by providing a generic tribal framework for further research in the area of interacting communities of organisms.

Acknowledgment

The author would like to thank Dave Ackley and Trent Small for their development of the Movable Feast Machine and the students in the Robust Artificial Life class at the University of New Mexico for providing a forum for discussion and development of artificial life.

References

Ackley, D. (2013a). Bespoke physics for living technology. *Artificial Life*, 19(3):347–364.

Ackley, D. (2013b). Beyond efficiency. *Communications of the ACM*.

Ackley, D. and Cannon, D. (2011). Pursue robust indefinite scalability. *The 13th Workshop on Hot Topics in Operating Systems*.

Ackley, D. and Small, T. (2014). Mfmv2. `https://github.com/Sixstring982/MFMv2`.

Diamond, J. (1997). *Guns, germs and steel*. Random House.

Fontana, W. and Buss, L. (1994). What would be conserved if "the tape were played twice"? *Proc. National Academy of Science*, 91:757–761.

Gat, A. (2006). *WAR in human civilization*. Oxford University Press.

Hutton, T. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artificial Life*, 13(1).

Lenski, R., O. C. e. a. (2003). The evolutionary origin of complex features. *Nature*, 423.

Nunno, L. (2014a). Mfm: Tribal warfare with a central column of mines 3kaeps. `http://youtu.be/6S6T-Rw5vLU`.

Nunno, L. (2014b). Mfm: Tribal warfare with a central column of mines 90kaeps. `http://youtu.be/g634zkKmTUo`.

Nunno, L. (2014c). Mfmv2. `https://github.com/lnunno/MFMv2/`.

Ruiz, A. (2014). Finding my place in the crowd: group formation and localization in a virtual environment. *Personal Communication*.

Stallings, E. (2014). Resource finding and distribution in an indefinitely scalable machine. *Personal Communication*.