Group 5 Final Report

Harjot Gill, Yang Li, Tim Miller

September 26, 2013

1 Introduction

This report describes our approach and observations while working on the *Organisms* simulation problem. The simulation is a class of popular *Discrete Cellular Automaton* model, similar in spirit to *Conway's Game of Life*.

The simulation world is a finite grid in which virtual organisms occupy individual cells in the grid. These virtual organisms can choose to move, eat or reproduce in each discrete cycle of the simulation. They are also able to "see" the neighboring cells and are thus able to sense the presence of food and/or other organisms. The simulation is subject to various rules and constraints. Each organism has finite energy and there is a cap on maximum energy an organism can accumulate. Energy is replenished by consuming food and it is consumed in every cycle depending on organism's action - i.e. moving, reproducing or even staying still consumes some finite amount of energy. An organism ceases to exist when it's energy level drops to zero. Food may appear on any empty cell in the simulation world with some finite probability and it may increase by a unit with some finite probability. These probabilities are not known to the organism. The maximum amount of food that is available in any cell is also bound to a finite number. An organism may also "communicate" with nearby organisms by broadcasting an external state, that can used in a versatile manner. The simulation can be run in single-species mode or in multiple-species mode. The winning conditions are not concretely defined but some of the factors that can be considered while judging a winning strategy can be average energy per organism, average extinction rate, total number of organisms for a species vis-a-vis other species etc. after some finite number of rounds/cycles.

Our group came up with various approaches to the simulation and depending on results, we pivoted a few times. Initially, our approach focused primarily to maximize the average energy. We developed an organism brain which used different *threshold* levels to decide whether to move, reproduce or to stay still. The implementation assumed fairly high probability of food appearing on the board. However, the strategy fell flat in conditions where food was scarce. We then implemented a strategy in which the organisms tried to gauge the probability of food appearing and doubling, either individually or collectively using some means of communication. We also came up with mathematical equations to determine the "utility" of making a move, consuming food, moving or staying still in each cycle of the simulation. The utility functions allowed us to choose the "best" move for an organism.

However, the release of *"Sheep"* organism from the previous undertaking of this class completely changed our thinking of this game. We realized that our strategies were not agile enough to adapt to multi-species simulations and our organism species was being constantly starved into extinction by a fairly aggressive reference organism species. We discarded the old approaches in favor of a *"Flood"* based approach, which aimed to reproduce effectively in the simulation arena so as to avoid possible extinction by other organisms. To achieve balance, an organism initially tended to stay at one place, then it transformed to aggressive behavior and the older it got, it again preferred to stay at one place. We found this implementation to be quite effective against multi-species simulations and our implementation was even able cause extinction of the reference *"Sheep"* organism in all scenarios, even in the scenarios when it was outnumbered by 1:5 initially at the start of simulation. On the side, we also took cues from strategies used by other groups in the class and we particularly noticed that structured/patterned farming colony approaches were quite

effective in increasing average energy of the organisms. We also implemented a *"Pattern-Maker"* organism brain which tried to form structured farming colonies, so as to achieve the sustainable balance between number of organisms in the simulation and food growth rate. We found this implementation quite effective in single-species simulation.

The rest of the report is structured as follows: Section 2 describes our initial observations about the simulation world and initial strategy. Section 3 provides a 10,000 feet view of various strategies that went into our implementation. Section 4 discusses in great detail, our implementation and how we deduced various mathematical equations that were used to model the organism behavior. Section 6 enumerates unique contributions of our group in the form of ideas discussed and presented in class to come up with effective strategies. Section 7 discusses limitations of our solution and possible improvements that can be made in the future work. Section 8 acknowledges ideas which were adopted from other groups or external sources. Finally, Section 9 summarizes and concludes this report.

2 Initial Insights and Observations

3 Strategies and Concepts

3.1 Single Player

3.1.1 Surviving Mode

One thing we noticed is that for harsh environment(p = 0.1%), the organism will easily extinct if moving around too much, especially at the beginning of game. However, if we are able to let the food to grow for a while, the board will become abundant, which then allows the organism to make some move and maximize total power. Basically, we attempt to help our organism *Survive* the beginning couple hundred rounds(Stage One) and switch to energy miximizing mode(Stage Two). Note that the surviving mode is just one layer put at the beginning of the game. At some point(details in Implementation), when organism enters stage two, *any* algorithm for maximizing energy could be invoked. The two stages are totally decoupled.

Move or stay. During this surviving mode, apparently, reproduce is not a good action. The energy spliting makes the origanism weak and cost of stayput will doubled. The question remained to be whether the organism should just stay, waiting for food to appear in the neighbor cells or move around to explore food. We answer this question by introducing a probabilistic analysis over the benefit of moving or stay. More specifically, we will calculate the probability of finding food when you move and compare it with the probablity of finding food if those energy are used to stayput and wait, so as to determine the best action in each round.

Eat or wait. The idea above is focusing on finding one food. But once a food is found in neighbor cell, shall the organism just go ahead and eat it or wait for as long as it can. Our answer is the later, based the following three reasons.

1. This is single player mode, no one will appear and rob the food away.
2. The goal of this stage is to survive the beginning harsh environment, waiting is more suitable to this.
3. The expectation of food on that cell grows exponential to the time you wait(more details in Implementation).

3.2 Multiple Player

3.2.1 Flood

Our multiple-player organism is named *Flood*, which comes from the flooding behavior of the strategy.

Generations. We defined three *generations* for this strategy, determined by the amount of energy left, playing different roles for the whole organism family.

Gen 0: Aggressively move, rob food and reproduce. (E.g., the very initial organism(the mother).)

Gen 1: Mildly expand, search food, reproduce if energy is high enough.

Gen 2: Stay quietly, only move if food around. If find more than one friend around, try moving away and be sparse looking.

Generation 0 is attempting to spread out as fast as possible, find more food, cover more ground. Generation 1 is in between, not so aggressive, but also not really quiet. Generation 2 will stay for most of the time, attempting to defend the zone covered and absorb resource on it.

Stay Sparse. We want to highlight our strategy for generation 2 of being spare here. We believe that we are the very first group to actually point this out. There are lots of advantages of such action.

1. Use less organism to cover more places.
2. The empty space in between can be treated as some sort of farm, making the coverage more sustainable.
3. Each organism has a chance to see more empty block around, provides it more chance to survive and hold the area.

Genetic Mutation. The key trick of this player is that, we enable mutation for the organisms based on the energy it has. For instance, if a generation 2 organism finds a cell with lots of food, then it has enough energy to be aggressive and will thus mutate to generation 0. Such strategy allows the family to form a self-balanced ecosystem. When organisms have low energy, they will stay still and try to stay as long as possible and wait for chance. When some organism has high energy, they will attempt to expand, use the extra energy to cover more ground and to suffocate the enemies.

Block Enemy. For any generation, if observe enemy in neighbor cell, it will stay still, attempt to block the action of that enemy and only move if observing food.

4 Implementation

4.1 Surviving Mode

Recall that the goal of this surviving mode is to live through the beginning stage of the game, when there are very little food on the ground and switch to the mode that maximize energy(pattern maker).

Move or stay. To determine whether it is better to stay where we are or to move, we try to calculate the payoff of each actions. More specifically, we will calculate the probability of *not* finding any food if (a) spend v energy to move or (b) say the energy to stay for $\frac{v}{s}$. Assume the organism has already wait for t time and there is still no food observed in the neighbor cells.

- If moving, in the next round, the organism will be able to see three new cells each having probability of $(1-p)^{t+1}$ of no food grown. Remember that the organism leave the original cell at time(round) t, which allows food to grow on this cell at $t+1$ round. Therefore, overall, the probability of finding no food after move is $(1-p)^{3(t+1)+1}$, i.e., probability of no food observed on the three new cells and no food grows on the original cell.
- If stay for $\frac{v}{s}$ rounds, each round, the probability of no food grows in each of the four neighbor cells is $(1-p)$. Thus the overall probability of no food observed after those rounds is $(1-p)^{\frac{4v}{s}}$.

According to the above analysis, apparently, the organism should stay at the beginning until the time $3(t+1)+1 > \frac{4v}{s}((1-p)$ is smaller than 1, thus larger index yields smaller probability). Note that an interesting property of this formula is that the analysis depends on p while the final formula itself does not, which is important in this case since p is not given. Once the organism starts moving, it won't stop and will keep following one direction, simply because make any turn will lose the chance to see an entirely new cell.

Eat or wait. Once the organism find a food nearby, it will wait for as long as it can, and then eat the food. The three reasons of this has already been discussed in previous section. Now we briefly explain why the expected amount of food is exponential to the time you wait(more specifically, $(1+q)^t$). Note that this is not concrete proof, just reasoning by intuition. First of all, the simplest case, when there is one food, in the next round, the expectation is $q \times 2 + (1-q) \times 1 = 1+q$. Using induction, at time t, assume the expectation of food is $(1+q)^t$. The expectation could be written as $\sum_i p_i \times i$ where p_i is the probability of there being i food on cell. For i food, each of them has expectation to $1+q$, with summing up being $i \times (1+q)$. The expectation on $\sum_i p_i \times i$ now becomes $\sum_i p_i \times i \times (1+q)$. Therefore, after time $t+1$, the expectation of food is $(1+q)^{t+1}$.

Switch mode. In the current implementation, when $energyleft + foodleftxu > 2M$. In other word, if there is enough energy for both this organism and the child it (could) reproduces.

4.2 Flood

Recall that flood is our multi-player strategy that having three generations, which can be mutated from one to another, allowing the whole family to switch between being aggressively or being quietly depending on different scenarios they encounter.

Reproduction. To cover the board as fast as possible, when an organism reproduce, we use the strategy of one north(east) and one south(west). Remember that the newly reproduced organism will following the same rule, which is essentially how we achieve the goal of exploring all directions simultaneously. The condition for an organism to reproduce is simply a check on its energyleft. If having more than 40% of M(which turns out to be not robust when M is set low, more details in Analysis section), the organism will reproduce to the direction following the rule we just described.

Genetic Mutation. The mutation is achieved by checking the energyleft of the organism. In the current implementation, it is Gen 0: $> 80\%M$, Gen 1: $60\%M - 80\%M$, Gen 2: $< 20\%M$. Such implementation turns out to be both clean and efficient.

Detect Enemy. We use same external state for all of our organisms which allows us to identify the organism appears in the neighbor cell is a friend or not. The value of such state is picked arbitrary and currently hard coded.

Keep Spare. Based on the enemy detection method above, the organism is now able to check number of friends around. If there are more than one friends, then it will try to move and stay away from each other. In addition, the organism always remember the last direction it moves so as to avoid stepping back and forth and waste energy.

4.3 Other Implementation Highlights

Directions. Instead of directly store and use the direction constant given by the simulator, we have a private lib which is essentially built on an array representation of the four directions. The most important advantage is that we can use for loop when we need to check all directions instead of having a switch and handle each direction individually. This makes our code much more compact and make it easy to have util-functions like check if there is food(enemy) around, left, right, reverse of a given direction, etc.