Project 4: Organisms II: Group 4 Report

Adam Rosenzweig (amr152)
Hanhua Feng (hanhua@cs)
Lyuba Golub (lg532)

December 1, 2003

1 Introduction

Organisms II is a project based on a world populated by amoebic organisms of various species. It is extremely difficult for the organisms to communicate effectively with each other, and their options for actions are also extremely limited. The goal is to develop a species that survives for a long time.

Our group decided to focus on maximizing the quality and length of life for our organisms. By quality of life we mean that the organisms should retain a high energy level through as much of their lifespan as possible. To accomplish this, we needed to keep part of the board open for food to accumulate. We also wanted to prevent the population from fluctuating significantly and regularly.

2 Methods and discussion

2.1 Idle Player - "Tequila Highlander"

Motivation

The idle player was originally conceived as a combination of something extremely easy to do early on, and the realization that the original game length was 500 rounds, with starting energy of 1000, which meant that simply sitting and never moving could ensure survival. Obviously this situation changed, but the player evolved somewhat. It is maximally lazy - if it finds food it will sit on that food until all of it is consumed, and if it is not already on food, it will only move if food is right next to it. This produces an extremely robust player, if your goal is simply to not die. It will retain a high energy state easily, but since it never even tries to reproduce, it will never reach a population above one.

Advantages

- Extremely simple player to code.
- Survives fairly well in multi-player situations as well as in single player situations.
- Really only sensitive to extremely low values of p or extreme overpopulation of its area - as long as food can pop up next to it, it will be able to survive.

Disadvantages

- Never dominates a board.

- Never increases population beyond one, so any problem finding food can kill it completely. Also, having a population of one does not lend itself to a high rating in the tournament.

2.2 Checkerboard player – "Tequila Kasparov"

Motivation

Tequila Kasparov grew originally from the general idea of farming. Having decided that the level of communication required to run a farm properly, even a small one, was too high to be worth bothering with (we changed our minds later), we constructed a player that uses no communication but attempts to keep the board open. We had seen many players overpopulate the board and then quickly go extinct due to there being nowhere for food to grow.

Implementation

What we did was design the player so that it would attempt to construct a semi-stable lattice on the board, where we tried to minimize the adjacency of organisms. To this end, we told the organisms that if there was more than one organism next to them, they had to move away from the organisms they saw. In addition, we told them not to reproduce if there was even one organism next to them. After some experimentation, we kept the chance of moving when alone low (unless there was food needed), because it led to a more stable lattice and reduced the chance of (potentially useless) random movement sapping their energy before they found food. We tolerated there being one organism adjacent without any special treatment (except a slightly higher chance to move regardless of food), because this also led to a more stable lattice.

An additional mechanism for population control is the use of thresholds. There is a threshold for energy which, when an organism is below it, the organism considers itself to be starving, and seeking out food becomes its top priority. There is another threshold which determines if an organism has enough energy to be willing to reproduce.

The end result of this code (around 130 lines, sans comments and whitespace) is a fairly decent approximation of a checkerboard. It is not perfect, but it is fairly close, and remains fairly close even as the organisms move for food and in reaction to each other. In single player, it effectively caps population to approximately 40% of the total board size (130 or so on the default 15x20 board).

In addition, as we were gearing this player towards single player games rather than multi-player ones, we decided that its organisms should not sit on food unless they need to eat. By moving away from food when they are no longer hungry, we not only make the food accessible to other hungry organisms, but we also give the food a chance to multiply.

For lack of anything better to do with our external state, we began to spoof nearby organisms. If we were alone, we picked a random valid state number. If anybody was next to us, we selected one of them at random and copied their state. We have no idea if this actually produced any effect in the games.

Adaptation to Board Parameters

After a couple of iterations, we adjusted the player to determine its thresholds based on the exposed information about the world – M, u and v. Thus, the starvation threshold is set to a value a little larger than one unit of food will grant from being eaten (but how much larger is based on how much a move costs). The threshold for reproduction is somewhat more than double this

threshold. In addition, the (extremely low) probabilities of moving are based on v. Also, the chance to reproduce (assuming that the reproduction energy threshold is exceeded) is based on how many visible tiles have food on them, in a very rough approximation of p.

This massively increased the survivability and population capacity of the player in board configurations other than the one it was originally built on.

Advantages

- Relatively simple player to code (about 130 lines that are relevant to the player's performance).
- Once it reaches a stable configuration, the player almost invariably survives to the end, with minimal population change.
- There is only one fluctuation in population before reaching the stable configuration.

Disadvantages

- The player is extremely sensitive to the ratio of u and v. When u/v is 10, the population reaches the full 40% of the board size. The population drops steadily as u/v drops, but when u/v reaches 2.0, the player loses both stability and the vast majority of its population capacity. This is likely because the player rarely stays on top of food, and thus moves, eats, and moves away from the food. If u/v is 2.0, there is no net gain in energy from this sequence.
- The player is extremely sensitive to the value of p. Because it moves very little to conserve energy and the stable board state, it does not explore the board looking for food. This means that if p is low, the player may not "find" food, since its method of finding is basically waiting for FedEx to deliver it.
- The player does not grow particularly quickly early in the game, which weakens it even more for multi-player games, where it already does not perform especially well.

2.3 Life cycle player – "Tequila Sunset"

The initial idea was to have our organisms imitate nature by building a life cycle into their so-called DNA. We set a ages when a child becomes an adult and an adult becomes a senior citizen. During childhood, the organism eats as much as it can to prepare for reproduction, but does not actually reproduce. Childhood gives the organism time to observe its environment and modify its genes accordingly. Sunset keeps track of the average amount of energy it has gained per round, and from this information sets such "genes" as reproductive rate, adult age, and old age.

Reproduction starts when Sunset reaches adulthood. It clones itself according to a rate determined by its environment. It passes down what it has learned about the environment to its children via the key. The form of the key is: number of digits needed to represent the adult age, digits for old age, adult age, old age.

Once it reaches old age it ceases to reproduce, but continues to search for food until it dies. Halting reproduction controls the population.

In previous versions of Sunset, senior citizens would stay put until food ran out and they expired. While inactive old people may keep real populations thriving, for this experiment they were detrimental. Many times, an organism's descendants died before it. In such a case, organism would sit on its spot stupidly while food grew all around it! The discrepancy between nature and

the simulation is probably due to the fact that when an organism reproduces, the daughter cells get half the energy of the parents cell. Thus, there is a fairly large chance that a child will die before a parent. To account for this, an organism that reaches its old age is sent into a second childhood, where it actively looks for food. A really strong organism that survives to be 1000 years old is reborn, its age set to zero.

Combining life cycle and checkerboard

Originally, the life cycle player only moved when there was food in an adjacent square. This was fine when the probability of food landing on a square was high. However, when p was low, the population would survive only when food happened to land next to its initial position. The Sunset player needed a good strategy for movement. The checkerboard strategy was a good fit, as it performed well on its own and was easy to incorporate into Sunset.

Advantages

Sunset performs well in single player when:

- p and q are both high
- p is high and q is low - as long as it can periodically energize itself, Sunset happily moves through its life cycle.
- p and q are both medium

Disadvantages

Sunset goes extinct when both p and q are low. It is especially sensitive to low values of p because it is highly dependent on finding food quickly. For multi-player games, Sunset does not survive more than a few hundred rounds.

2.4 Tessellations and patterns – "Tequila Octopus" and "Tequila Farmland"

One of our initial thoughts is to place organisms at the pre-arranged locations on the board. However, there are some difficulties in the implementation. For examples, since two dimensions of the board are unknown and the board has a wrap-around effect, the left branch of a pattern would be interacting with the right branch and some unexpected effects, usually combining with over-population, would result.

Using global patterns: growing like a tree

Our first player, "Tequila Octopus", tried to ignore the wraparound effect. This player is a fairly lazy player – all organisms try to stay at the original places where they were. When an organism collected enough energy, it would reproduce toward a designated direction (we call the immediate cell at this direction the front-side cell and the immediate cell at the opposite direction as the back-side cell). If hungry, it would move to an adjacent cell for food, and then return as soon as possible. If a cell is vacant for a long time, the front-side or back-side neighboring organism would try to fill this cell by reproduction. The first organism (or the root organism) tries to reproduce towards all four directions, and the new organism at each direction would use this direction as its designated direction. Each organism has a chance to generate a *branch*, i.e., to reproduce towards

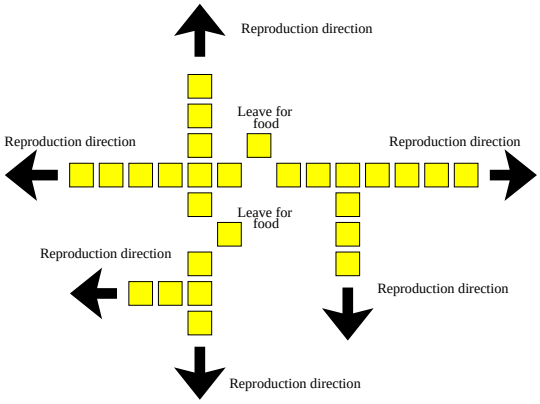

Figure 1: Illustration of "Tequila Octopus"

the a perpendicular direction against its designated one, if there are enough spaces between the new branch and the previous branch (usually at a distance of 3 or 4). Figure 1 illustrates the behavior of this player.

As shown in the figure 1, organisms of this player grow like a tree and the cells between branches are used to supply foods. The organisms on the tree would move locally for food, so it looks like moving arms of an octopus – therefore the name. However, the wraparound effect make this player less look like an octopus. This player works well if resources are not extremely low, but it is vulnerable if all organisms stay locally at a fixed place and the food production is not efficient enough.

Improved patterns with tessellation: "Tequila Farmland"

Later we implemented a new player, "Tequila Farmland", which would not constrain organisms at their original place. The cells in the board are divided into two sets: farming cells and street cells. The organisms are required to stay in the street cells, unless they are hungry, under which circumstance they can go into farm cells to eat food but are required to leave as soon as possible. The organisms can move along the two-way or one-way streets in order to avoid the locality problem. Figure 2 illustrate two patterns of the farms and street: it looks that we are using some kind of tessellation. Other patterns can also be implemented with some minor modifications.

The major problem of this player is the wrap-around effect and we cannot avoid this problem because the board size is unobservable. We then tried to using methods of communications to get the dimensions of board. We have to use about 32 bits to encode both x- and y-coordinates and other information. Once an organism finds out that a neighboring organism has a negative coordinate while it itself has a positive coordinates, it would immediately know a dimension of the board by computing the difference. However, the 8-bit state of each organism is too short to be efficiently used to exchange a 32-bit integer, since both organisms have to stay put for several rounds. Frequent exchange of information make the whole player very dumb. The communication methods would be discussed in a later section.

Finally, we tried to improve the communication methods by only exchanging the lower bits of the x- and y- coordinates (or, say, the relative x- and y-offsets in a single tile), so they can be encoded into a single bytes. This works pretty well - however we did not have enough time to tune it, so this player was not submitted for the tournament.

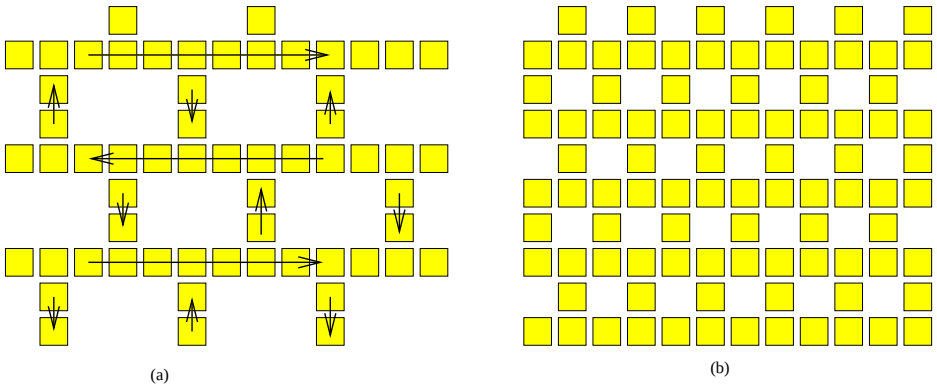

Figure 2: (a) the standard "Tequila Farmland" pattern and (b) a compact pattern. One can use other patterns – even checkerboard is a special case since we allow using a cell as a half-farm and half-street cell. The illustrated two patterns have a special property: street cells are connected and adjacent to at least one farming cell. (a) also illustrates a possible configuration of street directions.

2.5 Penetrating and conquest – "Tequila Idiot"

Meanwhile we implemented another player, "Tequila Idiot", aiming at the multi-player games. This player would make aggressive moves but keep the average expense of the moves low, so they would not run out of energy quickly.

Moving strategy in "Tequila Idiot"

There are two questions in the movement strategy. The first is how frequently an organism moves, and the second is along what direction it move, or in other words, what the movement trajectory is.

The frequency of moves is base on the ratio of the expense of a move to the expense of staying put, namely $v/s = v$. If v is high, organisms will move less frequently. This mechanism makes the average movement expense per round low. The initial number of rounds between two consecutive moves is set to $v-1$, and this number is adaptive via natural selection – each organism can vary this number a little bit from that of its parent. If an organism has a better configuration of parameters, it will have a better chance to be alive and to reproduce and thus will have more descendants.

The trajectory of moves for each organism is very simple. An organism will move in a fixed direction, unless an obstacle (another organism) is in its way. In the latter case, the organism will either make a turn, or keep the original direction after sliding left or right.

Reproduction strategy in "Tequila Idiot"

When an organism has enough energy, it will reproduce. In this player, an organism will reproduce a new organism in the direction which is perpendicular to its designated moving direction, and the designated moving direction of the new organism is the direction of reproduction. This method helps organisms spread all over the board.

The threshold of the required energy for reproduction is relatively important to this player (and presumably also for many of the other players). In order to keep the player less vulnerable, we use a random number generator to give differentiated thresholds for organisms. However, we did not use

any adaptive methods since we found that they do not work well with our player, because organisms that have a lower threshold would always have a better chance to reproduce and the result of using adaptation is that the quality of the population becomes lower and therefore vulnerable.

Final time-based tuning "Tequila Idiot"

Before the tournament, we tuned this player. First, we did a lot of experiments with different configurations of (u, v). We found, if the ratio of u to v is relatively high, it is better to use a lower threshold of average reproduction energy. However, if one player has enough organisms on the board and they have already spreaded to everywhere, using a higher threshold would help to make the player more competitive. After realizing this, we record the round number (a.k.a. the current year) and pass it in the keys to the descendants. If the round number is small, the player uses a low average threshold of reproduction energy for quick expansion; after a while the average threshold will be elevated gradually in order to provide a better quality of life, i.e., a lower birth and death rate and a longer lifetime.

Moreover, we found that the total energy will be higher if the organisms move less frequently. So after 20000 rounds, the player will slow down all organisms by a factor of four, and experiments showed that the total energy can increase by a factor between 10% to 100%. Also, this change will make our player more competitive and increase the possibility of eliminating other players. Unfortunately in the tournament the maximum number of rounds is 20000 instead of the default value of 50000, so this modification became useless in the tournament.

We also planned to switch to the "Tequila Farmland" or "Tequila Sunset" after 25000 rounds, but we did not do that due to time limits and objectives.

Communication between organisms

Since a single byte is too small to exchange information between two organisms, we had problems with initial communication, and it became much harder when other players were present. So we tried to use multiple-byte communication methods by stalling both organisms involved for multiple rounds. In order to dismiss impostors, we first used a two-way hand-shaking mechanism (similar to the TCP protocol) encrypted by a same random key (a constant). In addition, we also added the direction of prospective connections to the hand-shaking message, so an impostor cannot interfere the communication by just copying the states, because the impostor is in a different direction. However, this seemed to be too complicated, so we then moved to a mechanism similar to the UDP protocol. One organism sends out a one-byte message to request information, and the organisms who receive this byte will send out an 8-byte message. The first byte indicates the type of the message and the remaining seven bytes encode a full 32-bit integer. Each byte contains five bits of the 32-bit integer and the other three bits in each byte is the index number of the byte (0-7) to avoid confusion.

However this mechanism is still too complicated, and frequent message requests make the whole system very dumb because it requires both organisms to stay put for at least eight rounds. In the latest "Tequila Farmland" we used a very simple strategy that encode everything a single byte: three bits for x-coordinates and three bits for y-coordinates and the remaining two bits indicating the food quality. Every organism that receives this message will compare it with its own knowledge. If it finds some difference, it can check-up the history information and message sent by other organisms, or more simply, change its own knowledge to agree others in a small probability. [1] To

[1] An interesting problem of "the convergence of the world" can be derived. Suppose there are M choices and N people who make choices. Choices do not have an evaluation, i.e., there are no good choices and no bad choices, and

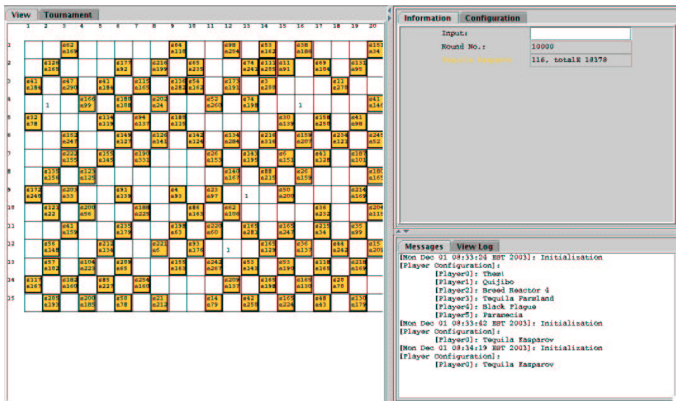

Figure 3: Snapshot of "Tequila Kasparov"

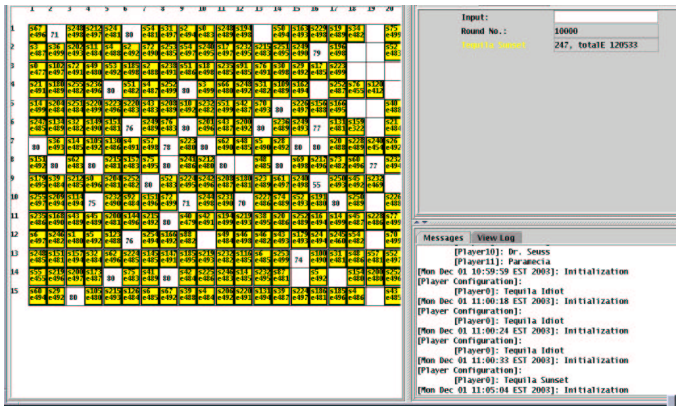

Figure 4: Snapshot of "Tequila Sunset"

avoid impostors, we planned to switch to this strategy only after 25000 rounds; experiments show that only one organism can survive until this time in most configurations.

3 Results

3.1 Implementation of our players

"Tequila Kasparov" exhibits a decent approximation of a checkerboard (Figure 3), and "Tequila Sunset" maintains a huge amount of energy (Figure 4). Figure 5 shows that "Tequila Octopus" is growing, and Figure 6 shows that "Tequila Farmland" have built up the farms and streets.

people are not distinguishable. Each person has a random initial choice and can interrogate a random person to ask for the latter's choice. After interrogation, one may switch to the other person's choice with probability p. If the history of interrogations are not allowed, will the world converge to a single choice? If so, how fast? If p itself is an I.I.D. random variable for each person, (for example, some people may be very stubborn and others are willing to change their choices) what is the relationship between distribution of p and the convergence of the world?

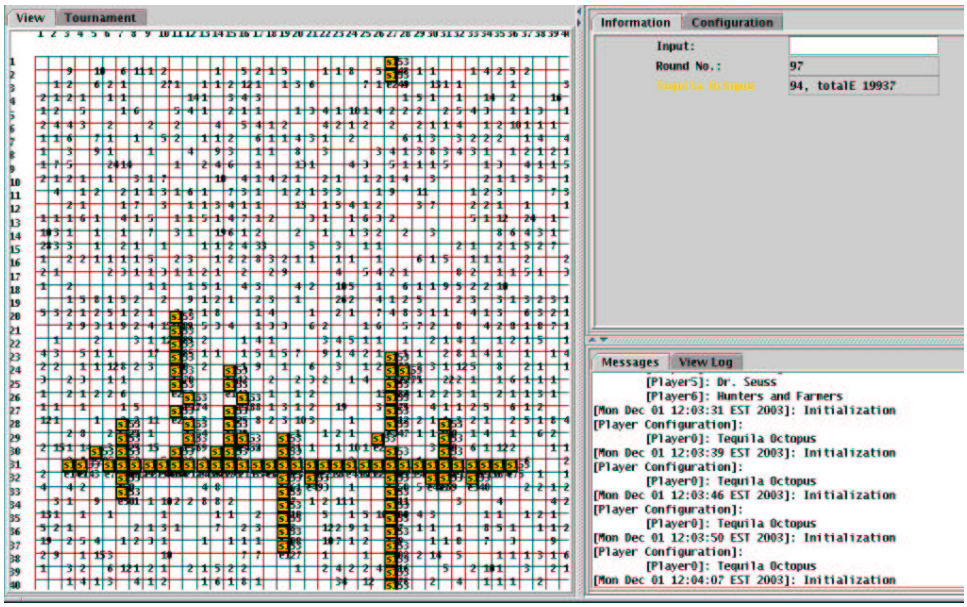

Figure 5: Snapshot of "Tequila Octopus"

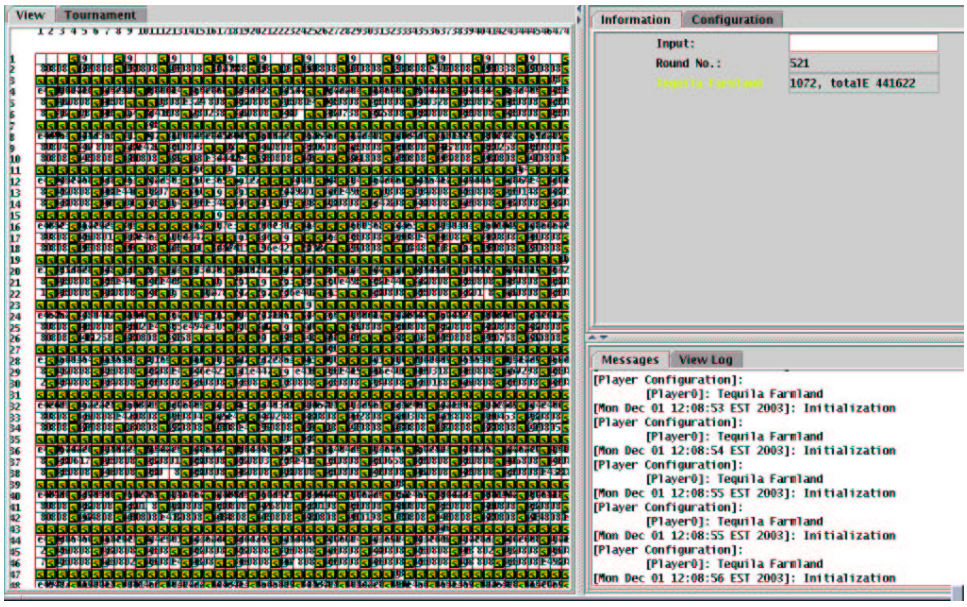

Figure 6: Snapshot of "Tequila Farmland"

Configuration	Group4Player2	Group4Player3	Group4PlayerA4	best players
$u = 20$, $v = 4$, $K = 57$, $q = 0.02$	$p = 0.00044$	$p = 0.00063$	$p = 0.00508$	$p = 0.00022$
$u = 20$, $v = 4$, $K = 57$, $p = 0.001$	$q = 0.00045$	$q = 0.00807$	–	$q = 0.0001$
$u = 20$, $v = 10$, $K = 57$, $q = 0.02$	$p = 0.0004$	$p = 0.00061$	$p = 0.00543$	$p = 0.0004$*
$u = 20$, $v = 10$, $K = 57$, $p = 0.002$	$q = 0.00004$	$q = 0.00142$	–	$q = 0.00001$
$u = 20$, $v = 18$, $K = 57$, $q = 0.02$	$p = 0.00066$	$p = 0.00055$	$p = 0.00452$	$p = 0.00055$*
$u = 20$, $v = 18$, $K = 57$, $p = 0.005$	$q = 0.00033$	$q = 0.00001$	–	$q = 0.00001$*
$u = 100$, $v = 10$, $K = 57$, $q = 0.02$	$p = 0.00033$	$p = 0.00032$	$p = 0.0009$	$p = 0.00005$
$u = 100$, $v = 10$, $K = 57$, $p = 0.001$	$q = 0.00001$	$q = 0.00064$	$p = 0.00725$	$q = 0.00001$*
$u = 20$, $v = 4$, $K = 83$, $q = 0.02$	$p = 0.00047$	$p = 0.00092$	$p = 0.00486$	$p = 0.0002$
$u = 20$, $v = 4$, $K = 83$, $p = 0.001$	$q = 0.00035$	$q = 0.00436$	–	$q = 0.0001$
$u = 20$, $v = 10$, $K = 83$, $q = 0.02$	$p = 0.00055$	$p = 0.00074$	$p = 0.00488$	$p = 0.00032$
$u = 20$, $v = 10$, $K = 83$, $p = 0.002$	$q = 0.00001$	$q = 0.0007$	–	$q = 0.00001$*
$u = 20$, $v = 18$, $K = 83$, $q = 0.02$	$p = 0.00062$	$p = 0.00062$	$p = 0.00312$	$p = 0.00061$
$u = 20$, $v = 18$, $K = 83$, $p = 0.005$	$q = 0.00064$	$q = 0.00001$	–	$q = 0.00001$*
$u = 100$, $v = 10$, $K = 83$, $q = 0.02$	$p = 0.00039$	$p = 0.00067$	$p = 0.00088$	$p = 0.00005$
$u = 100$, $v = 10$, $K = 83$, $p = 0.001$	$q = 0.00001$	$q = 0.00001$	$p = 0.0015$	$q = 0.00001$*

Figure 7: The minimum ps and qs that our players can survive. Asterisks mean that at least one of our player gets the best. Other parameters: $X = 23$, $Y = 36$, $M = 500$

3.2 Performance in the tournament

We submitted three players: tuned "Tequila Idiot" as the first tournament player, Group4Player2, "Tequila Sunset" as the second tournament player, Group4Player3, and "Tequila Kasparov" as the third player, Group4PlayerA4.

3.2.1 Surviving ability

Note that none of our players is optimized for survival under very harsh environment, which was not one of our objectives. However, the results show that two of our players, Group4Player2 ("Tequila Idiot") and Group4Player3 ("Tequila Sunset") survived fairly well when the food is extremely scarce. Figure 7 shows the single-player tournament results of harshest configurations that our player can survive, in comparison to the best players (which sometimes are our players).

3.2.2 Population and total energy

The most interesting and astonishing result in the single-player games is that our player Group4Player3 ("Tequila Subset") maintains tremendous population and energy, induced by the life cycle mechanism. In most of the cases, this player supports a population at least double of other players, and holds almost `10` times of the total energy. The tournament results agree with our experiments – since maintaining a low birth and death rate and then give a better quality of life to our organisms is one of our goals.

Figure 8 compares the population and total energy of Group4Player3 with other players, with the configurations that most of the players have a set of data.

Configuration	Group4Player3		The player at the second place	
	popu.	energy	population	total energy
$u = 20$,$v = 4$,$K = 57$,$p = 0.005$,$q = 0.02$	459.64	203524.2	158.77(Group6Player0)	21796.2(Group6Player0)
$u = 20$,$v = 10$,$K = 57$,$p = 0.0025$,$q = 0.02$	554.75	258117.3	213.02(Group5Player3)	28728.0(Group5Player3)
$u = 20$,$v = 18$,$K = 83$,$p = 0.005$,$q = 0.02$	683.95	334661.2	196.08(Group5Player3)	25846.4(Group5Player3)
$u = 100$,$v = 10$,$K = 83$,$p = 0.001$,$q = 0.005$	604.58	264678	203.07(Group6Player0)	26598.9(Group6Player0)

Figure 8: Samples of comparisons between Group4Player3 with other players in terms of the population and total energy. Other parameters: $X = 23$, $Y = 36$, $M = 500$

3.2.3 Multi-player games

Many of the seven-player games are not very interesting for us because p is too small in p-decreasing games (for example, no p's were in the range of (0.01,0.001), as we had expected). If p is too small, the individual surviving ability under harsh circumstance dominates everything, instead of competitional ability. Furthermore, since the lower-bound of p is 0.001 in the problem specification and we thought contention between players is an objective in multi-player games, we did not optimize our player for extremely low p's.

Nevertheless, the performance of our players were fairly good in multi-player games. In the q-decreasing cases of seven-player games, Group4Player2 survived in nearly all the configurations, since "Tequila Idiot" has a strong competitional ability. In the p-decreasing cases, Group4Player2 did well in most of the cases that p is not too small. For extremely small p's, none of the groups can survive. Comparing the extinction time is not so interesting since it mostly depends on luck: whether or not the random number generator favors you. Our another player, Group4Player3, also survived in many seven-player games, although it was originally designed to maintain a high total energy in single-player games.

4 Conclusion

We implemented many different strategies for this project and each of them gives some interesting insights to this problem. We wanted to exhibit more players in the tournament, but unfortunately, because we submitted three players, each player has only one third chance to attend the seven-player games. If we could have, we probably would have submitted even more players despite how it would reduce their exposure to the large games.