

Report

1. World

My simulation uses 24 grid size which is the default size. I decided to leave it not changed. And there will be 100 turns per simulation as my fitness score for survived creatures is really high. Accordingly, I have to prevent those “lucky” creatures who only know how to pick up food from getting too many fitness points. And the simulation will be run 2000 times as the “mutations” and “crossovers” can be a bit unstable. So, I have to give the creatures some time to recover from a bad mutation or crossover.

2. Chromosome

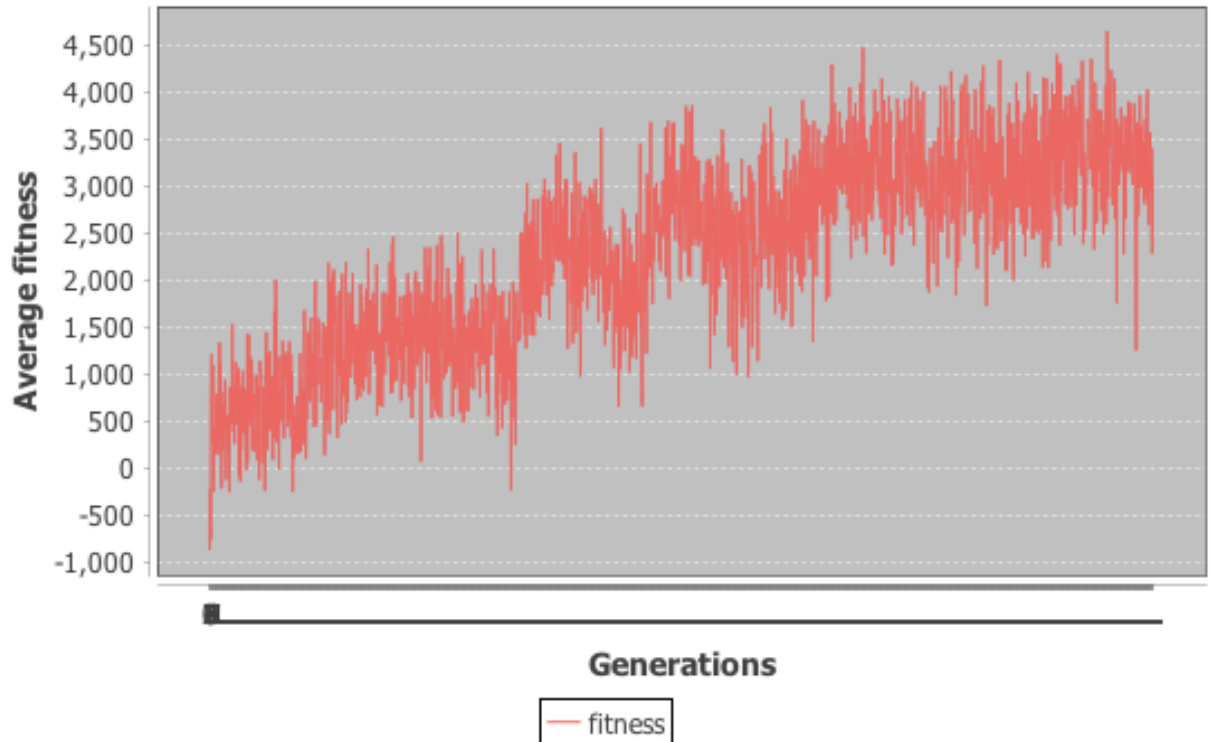
Regarding to the creatures, the chromosomes of the first generation of creatures are random ranged 100 integers from -10 to 50. Negative numbers are for the creatures to get away from something. And positive numbers are there to increase their “desires” for something. As the only thing we wanted to get away from is the monsters, so I decided to give fewer negative numbers than positive ones. And the reason why I used 100 chromosomes is really just I was too lazy to figure out how many chromosomes do I really need. However, I accidentally found out that this kind of “noise” in the chromosome set could help me to find out the diversity of the new generations.

3. Agent Function

I mapped perceptions into actions with functions in a form like “ $y_1 = ax_1 + bx_2 + \dots + gx_9$ ” or “ $y_2 = ax^2$ ” where the action is y , perceptions are x , and chromosomes are constants a, b, \dots . As I only told the program which chromosome to use for which perception, so the program need to find out a reasonable number for that chromosome itself. Again the reason why I selected this way to map the perceptions is that I was too lazy to write dozens of conditions to tell the program which perceptions are important for which action. Instead, I’ll let the program loop through all perceptions of that creature, and give each perception a corresponding chromosome. So the program can learn by itself while I tell it which is good and which is bad. That’s when the crossover and the fitness calculation take part in.

4. Fitness Function

Average Fitness vs Generations



My fitness function takes 4 situations into count:

- If a creature survived over 100 turns, then it must be an elite or very lucky. I'll reward it heavily.
- If a creature died starving, then it doesn't know how to eat or just being a bit unlucky. As you can certainly live longer even if you only know how to eat. So for those creatures who don't eat at all, which means they'll be dead at the 50th turn, I'll punish them badly. In another word, I'll reward those who managed to live longer than 50 turns.
- If a creature died eaten by a monster when it's not sick, then it could be both unlucky or don't know how to get away from a monster. Then I'll punish it, but not in a hard way.
- However, if the creature died eaten by a monster while it's sick, then something really bad is happening there. It must eat a toxic strawberry while there is a monster chasing it. Then I'll reduce its fitness points dramatically.

5. Crossover

I have two key concepts regarding the crossover function that's used in this algorithm.

- a. Only creatures that have a fitness score that's higher than the median score can share its gene with another creature.
- b. There is a 50% 50% chance that the child will get gene from each of its parents.

I've considered about applying elitism by only letting very outstanding creatures to produce the next generation. However, that would destroy the diversity of my new generations. I don't want it to be ended up every creature has a very similar chromosome set, so that creatures can only evolve counting on some lucky mutation.

6. Mutation

The mutation function of my algorithm is simple. Each gene in the chromosome set will have some chance to mutate into something totally random. The "some chance" in this case is 0.0015.

My algorithm can learn how to pick up strawberries very quickly as I predicted. Then it'll gradually learn how to get away from monsters. After that, some smart creatures will have already developed some skills like how to avoid green food while monsters are around or to wait for the food to be ripe. However, it has a hard time deciding which strawberry to eat when it's surrounded by more than one ripe strawberries.

Overall, it's done well. But I can still improve it by giving those creatures genders and set males into a tournament where stronger ones have higher possibilities to participate in the production. I can also give it a heavier reward for those who survived to improve the number of survived creatures after 100 turns.

Compile and run instruction:

I'm using NetBeans 8.1. Just run the MyWorld.java if you are using NetBeans.

For the statistical chart generation purpose, I'm using the SJFree java library.