Agent based Evolution Model in JAVA (ABEMJ)

Shama Yazdani
Institute of Technical Education and Research
Bhubaneswar, Odisha, India

Smit Anand
Institute of Technical Education and Research
Bhubaneswar, Odisha, India

Nishat Afreen
Birsa Institute of Technology
Sindri, Jharkhand, India

ABSTRACT

The human dominance on earth exhibits a general rule-'with intelligence comes power'. As of now, human brain is the most complicated object in the entire known universe. Agent Based Evolution Model in JAVA (ABEMJ) is a simulation designed on Java platform that portrays the learning and evolution of artificially intelligent agents. It is a two dimensional environment where agents mimicking human behavior, born with some minimum amount of energy survive, search for food and upon attaining a certain level, produce offspring. The agents showcase intelligence by using their memory for storing their experiences and learnings from their past mistakes. Their will to survive makes them protective about their food and they compete with each other using energy to win their only source of energy i.e. food. This paper describes a platform for the evolution of autonomous square shaped agents thus, providing an efficient tool for the study of artificial evolution and coevolution.

General Terms

Artificial Intelligence, Algorithms, Chromosomes, Crossover, Mutation, Fitness function.

Keywords

Agent Based Evolution Model in Java (ABEMJ), ALife, agents, evolution, genetic algorithm, artificial intelligence, positive food, negative food.

1. INTRODUCTION

The automaton has been an intimation of living creatures since anciently. In late 1940, John Von Neumann propounded the idea of self-reproducing automata. Von Neumann [1966] designed the first artificial-life model (without referring to it as such) when he created his famous self-reproducing, computation-universal cellular automata [3].At about the same time, Wiener started applying information theory and the analysis of self-regulatory processes (homeostasis) to the study of living system [4].It was later in 1989 that the term Artificial Life (ALife) was first coined by Chris Langton. Artificial Life is a field of study that attempts to understand natural life and its underlying phenomenon and aims to create man made systems by subjecting similar dynamics to another media. In fact, artificial life is not yet ready to be curbed by a concise definition. It is a field of study that is growing day by day. ALife doesn't aim merely to recreate the biological phenomenon but has a much wider perspective. But we must be able to differentiate the point where it extends the realm of biology. The coalescence of non-biological mechanisms and synthesis of various biological phenomenon can help us find the solutions of numerous unsolvable conundrums that have vexed us since time unknown.

Studying artificial life has the potential to put theory of evolution in a broader context and to help provide it with a firmer mathematical basis [2].The three broad approaches of artificial life are-soft, hard and wet artificial life. 'Soft' artificial life creates simulations or other purely digital constructions that exhibit life-like behavior, 'hard' artificial life produces hardware implementations of life-like systems, and 'wet' artificial life synthesizes living systems out of biochemical substances[4].

This paper has derived its motivation largely from the distinguished scientist Larry Yaeger's Polyworld simulator. PolyWorld is a computer model encompassing artificial organisms and its ecology. PolyWorld brings together biologically motivated genetics, simple simulated physiologies and metabolisms, Hebbian learning in arbitrary neural network architectures, a visual perceptive mechanism, and a suite of primitive behaviors in artificial organisms grounded in an ecology just complex enough to foster speciation and inter-species competition [1].This paper discusses an agent based environment designed on Java platform. The behavioral suite of the simulated agents is inclusive of moving, eating and reproducing. The artificial agents compete with each other for survival. Those who are vanquished by their contenders eventually die. The behaviors of these agents evolve through multiple generations. This work provides significant insights into natural intelligence.

2. AGENT BASED EVOLUTION MODEL IN JAVA

The model is designed for artificially intelligent agents to mimic basic human behavior. It is developed using Java, a robust, simple, object oriented and platform independent high level programming language. The agents in this model perform some fundamental functionalities like eating food, mating, reproduction, gathering and spending energy, protecting itself as well as its food. The model typically starts with a set of agents with initial energy and some food with either positive or negative energy associated with it. These agents then compete with each other to gain energy via food, spend energy with time, produce better and evolved offspring and then die.

The basic components of the model are described below:

2.1 Programming Language and Model

Java was chosen as the programming language of the model as it is robust, secure and platform independent. Java is a high level, object-oriented programming language developed by Sun Microsystems and released in 1995. Java being a platform independent language can be run in any platform as fast as native non-portable C, C++. Thus, Java is a perfect combination of performance and portability.

The model is designed using JPanel class of javax.Swing package in Java. The detailed screenshots depicting different components of the model is also illustrated below.

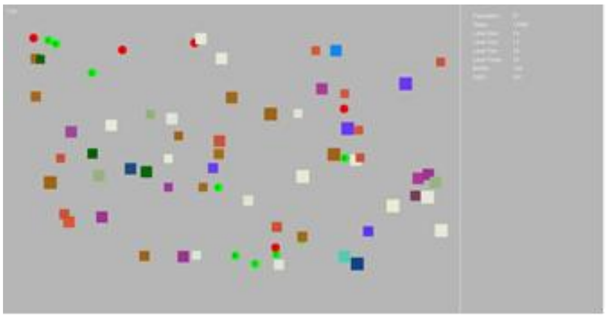
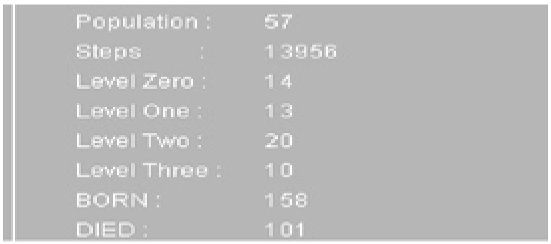


Fig.1 Screenshot of the Model

The model is divided into two parts, the left side of the model occupies a larger area and shows the live simulation of the environment containing agents and food, while the right side shows the live statistics of the environment, clearly shown in Fig 2.



```
Population :      57
Steps       :     13956
Level Zero :      14
Level One :       13
Level Two :       20
Level Three :     10
BORN :            158
DIED :            101
```

Fig.2 Live Statistics

The live statistics shown are:-

2.1.1 Population: The current population of the model.

2.1.2 Steps: Each step is the smallest unit of time of the environment, which is 10 milliseconds.

2.1.3 Level 0: Number of agents currently at level 0.

2.1.4 Level 1: Number of agents currently at level 1.

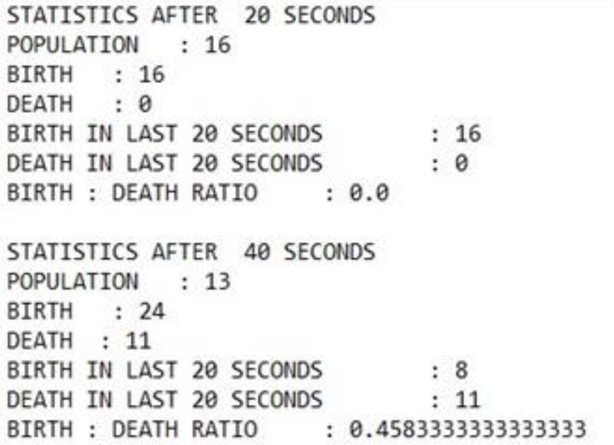
2.1.5 Level 2: Number of agents currently at level 2.

2.1.6 Level 3: Number of agents currently at level 3.

2.1.7 Born: Total number of agents that are born since beginning of the simulation.

2.1.8 Died: Total number of agents that died since beginning of the simulation.

This simulation model does not end at a particular time, it will continue to run till at least two agents exist i.e. there is still hope for the population to flourish. The simulation can be stopped any time by exiting it and upon exit a log file will be generated which contains statistics of simulation after every 20 seconds, which can be used for further analysis.



```
STATISTICS AFTER  20 SECONDS
POPULATION   : 16
BIRTH   : 16
DEATH   : 0
BIRTH IN LAST 20 SECONDS        : 16
DEATH IN LAST 20 SECONDS        : 0
BIRTH : DEATH RATIO      : 0.0

STATISTICS AFTER  40 SECONDS
POPULATION   : 13
BIRTH   : 24
DEATH  : 11
BIRTH IN LAST 20 SECONDS        : 8
DEATH IN LAST 20 SECONDS        : 11
BIRTH : DEATH RATIO      : 0.4583333333333333
```

Fig.3 Sample Output File

2.2 Agent

There are typically twelve agents in the beginning of the simulation, each represented by a square of different colors. These agents move around the space in search for food to increase their energy and level. They compete with each other and learn through experiences in their life.

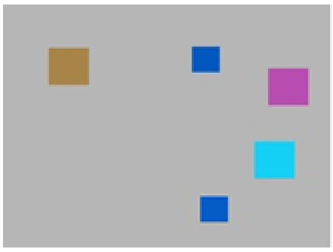


Fig.4 Agents

Each agent is made up of a single chromosome that decides its color. Each chromosome consists of three parts, indicating the percentage of the primary colors each i.e. red, green and blue. The length of chromosome for each agent is fifteen bits, which is further subdivided into three parts of size five bits corresponding to the primary colors in the sequence red, green and blue. Each color in the chromosome is represented by five bits which can be assigned an integer value ranging from 0 to 255. This combination of primary colors each using 5 bits and having 256 integer values makes a possibility of 256*256*256= 16777216 different colors for agents.

2.3 Food

ABEMJ environment contains food which helps the agents to survive in it. Mimicking the real world, some food here is beneficial for health i.e. has (+ve) 800 units of energy associated with it and green in color which will further be referred to as “positive food”, while others are harmful for health i.e. has (-ve) 200 units of energy associated with it and is red in color which will be hereafter referred to as “negative food”. Each type of food is circular in shape and has a color associated with it for ease of distinguishing the two types. The lifetime of food in seconds is displayed on it.

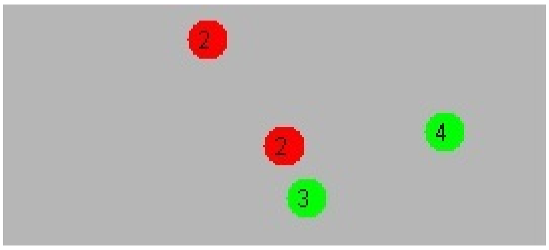


Fig.5 Food

In the beginning of the simulation there are fifteen positive food and five negative food scattered at random places in the environment. Each food has a timer associated with it which decides the lifetime of the food. With passage of time the energy of both positive and negative remains constant and after completion of lifetime the food is exhausted. When a type of food completes its lifetime and is exhausted, another food of same type is generated at a random position in the environment. The amount of food in the simulation is kept constant to fifteen positive food and five negative food, so as to keep a check on the population.

2.4 Agent's Energy, Memory and Level

In the Agent Based Evolution Model in JAVA, every agent is born with 1500 unit of energy, which is the basic initial amount that is required for sustaining its life and searching for food. An agent spends 1 unit of energy in every 10 millisecond as the cost of living in the environment. Each type of food in the environment is associated with positive or negative energy which remains constant till its decay. If the energy of agents crosses the maximum threshold value of 1800 units of energy, the agent no longer participates in the process of searching for food, they rather stay constant at a place and wait for their energy to reach below the maximum threshold value, and this is done so that every agents gets fair chance of sustaining their life. Also, the agents are optimized to keep their energy from reaching its minimum by constantly searching for right food so as to survive in the environment. Agents which cannot compete with others and increase their energy die when their energy falls to zero.

Every agent is associated with a single unit of memory which retains knowledge about the type of food it has eaten during its lifetime and the outcome of eating that food i.e. whether it has benefited from it or has it caused him any loss. The unit of memory associated with each agent is kept to a minimum of single unit for the sake of simplicity of the model. Whenever an agent eats a negative food and thus loses its energy, this memory unit helps it retain this incident and the agent learns that eating a negative food results in loss of energy. Any agent that has once witnessed the effect of eating a negative food will learn from its experience and never repeats this mistake.

Every agent in ABEMJ is associated with a level, for which the possible values are level 0, level 1, level 2 and level 3. The initial level of an agent at the time of its birth is level 0 and it increases with the number of food it consumes. While intake of positive food results in increase of level, consuming negative food has no effect on the level. Increase in level also results in increase of the size of the agent. Both level and size becomes constant after the agent reaches level 3, as it is the maximum level of the agents, no further food intake changes this level. The level of the agents signifies its readiness to serve as a candidate in the reproduction process. An agent which has consumed four or more positive food i.e. is at level 3 is considered fit for reproduction.

2.5 Search for food using Manhattan Distance

In an environment consisting of agents and different types of food, the agents have to continuously search for food as means of sustenance. With the agents continuously moving and no fixed position of food due to decay and growth, the agents need to find the food which is nearest to it so as to maximize its energy value. In ABEMJ the agents calculate its distance from every food in the environment by using the Manhattan Distance formula and proceeds towards the food nearest to it. If P is the position of organism and Φ is of the food, then Manhattan distance (Δ) can be calculated as,

$$P=(\xi 1,\psi 1) \qquad (1)$$

$$\Phi=(\xi 2,\psi 2) \qquad (2)$$

$$\Delta=|\xi 2-\xi 1|+|\psi 2-\psi 1| \qquad (3)$$

Manhattan distance is the distance between two points measured along axes at right angles [5]. The two points here being the coordinates of center of the agent which is a square and the coordinates of center of the food which is a circle. The coordinate system used here is the default coordinate system of JPanel in Java with upper left hand corner as the origin with point (0, 0).

2.6 Competition for Food

With each agent searching for food nearest to it, and huge difference between the number of agents and amount of food, multiple agents approaching the same food is a common scenario. To tackle such a situation the agents are programmed to be highly competitive while approaching its target.

As soon as an agent finds its nearest food, and fixes it as its target, it starts moving towards it. The nature of agent being protective and competitive, it perceives every other agent as a threat and to ensure its dominance it competes with every agent in its radius of 10 units and displaces any agent weaker than it i.e. having energy less than itself, randomly anywhere in the environment, thus ensuring its dominance over the food. All the agents compete with one another vigorously and the winner takes the food.

Whenever a weaker agent is defeated and displaced to another random place in the environment it again starts its search for the nearest food and competes for it, as to fulfill the ultimate aim of sustenance in the environment. While on the other hand, the winner who consumes the food does not immediately starts its search for new food, it rather assesses its need for more food or energy i.e. if its energy has reached its maximum threshold value of 1800 units, it waits for the energy to reduce by passage of time and only once its energy reduces below the maximum threshold value, it resumes its search.

2.7 Reproduction using GA

As reproduction is essential for survival and continuity of human race, so is it for our agents. The agents in ABEMJ follow the rules of genetic algorithm for reproduction.

2.7.1 Fitness

The fitness of the individuals are decided by its level, any agent that has reached level 3 is considered fit for reproduction.

2.7.2 Selection

As soon as an agent reaches level 3, its checks for a partner. If two agents are at level 3 simultaneously both serve as the parent to next generation and if a partner is not available, the agent waits for it.

2.7.3 Crossover

Two point crossover technique is used with fixed points at 5 and 10 in chromosome of 15 bit length.

After the termination of reproduction process, both parents as well as offspring become part of the environment and continue to survive. To mimic the natural process of reproduction, unlike the fate of parents in typical Genetic Algorithm, the parents continue to live.

3. EXPERIMENTAL RESULTS

The discussed evolution model was implemented with a help of a computer program in Java. The program takes into consideration an initial population of 12 agents, 15 green positive energy food and 5 red negative energy food. The total number of food is kept constant throughout the evolution process. The number of agents can increase of decrease depending on the availability of food and stored energy.

To facilitate the observation of variation of population with time, a graph is plotted on the basis of output obtained on running the simulation. GNU Octave is used to plot the graph illustrated in the fig 6.

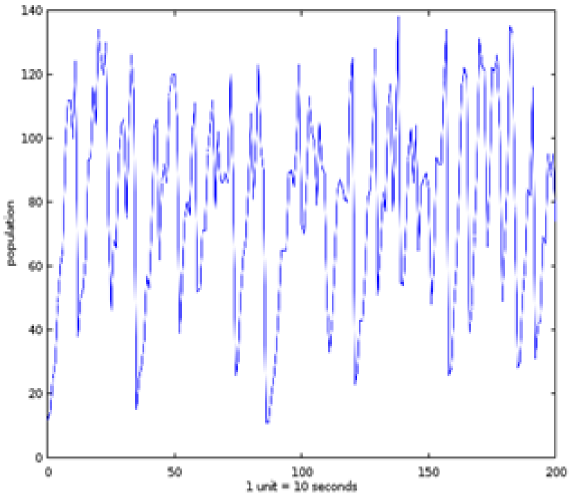


Fig.6 Graph depicting variation of population with time

The variance in the above population graph explains the struggle and competition for food among the agents. Since the total number of food is constant in the simulated environment, the probability of procuring the food by an agent is dependent on the population. If the population is low, the agents face less competition to obtain the food and hence, the probability of finding the food becomes higher. On the other hand, a disorderly state is reached when the population rises. Since the amount of food is limited in the environment and food is necessary for survival, all agents strive to obtain the food. In this process, they may be required to randomly change their path; if the food they are aiming for is eaten by some other agent before they reach the target. This creates a chaotic state and may leave many agents hungry for a longer period of time. Since they lose 1 unit of energy every 10 milliseconds, their energy reduces to zero and they eventually die. This explains the steep falls in the population graph.

The factors which influence the stability of the environment are initial energy of the agents, the positive energy of green foods and the number of green foods. A healthy and flourishing environment plays the most important role in the survival and evolution of the agents. The environment takes into account the initial energy of every agent as 1500, the positive energy given by green food as 800 and the number of green foods as 800. A graph depicting the variation of the ratio of the dead agents to born with time is illustrated below.

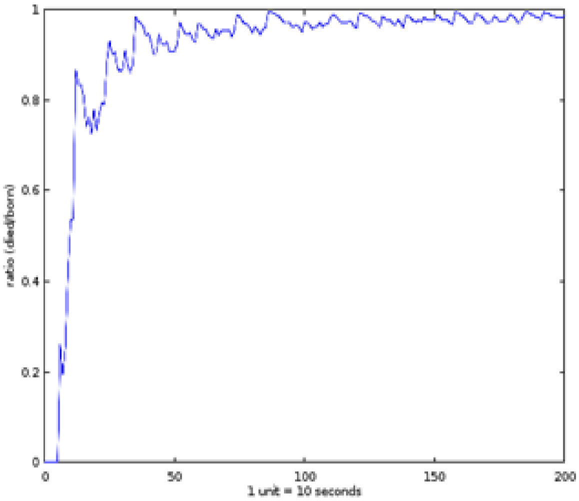


Fig.6 Graph depicting variation of ratio of dead to born agents with time

The behavior of the ratio graph is due to the fact that the death rate is very low initially as the population grows from an initial of 12 agents. As the population rises, the agents compete for survival. Those who fail to procure food eventually die. This explains the steep rise in the graph. When the population decreases, the probability of obtaining the food becomes higher and they become eligible for reproduction on reaching level 3.This leads to an increase in population. This process repeats again and again. But, with passing time, the ratio varies less which means that the environment tends to me more stable than before. The changing environmental factors thus, leads to changes in the graph.

4. CONCLUSION

This paper coherently describes the ecology of artificial agents in a two dimensional artificial media. This model depicts the way in which the agents evolve with passing time and generation. This study has a great potential for future advancement. It can serve as a test bed for unforeseen situations to pursue further evolutionary studies. One such future advancement in this study can be the retention of the memory of all the life experiences in addition to remembrance of consumption of food. This can help the simulated organisms to stay vigilant and make perceptive decisions.
The advent of artificial life can be a boon or bane to human life on earth. Humans can shape the progress of artificial life to prove beneficial to mankind. With the help of the study of artificial life, the elementary understanding of physics and biology can be perceived. The objective of AI is not dehumanization; rather it ought us to recognize the AI systems far beyond "machines". Classical AI underscores free choice as a vital characteristic of human beings.

5. REFERENCES

[1] Larry Yaeger, “Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or polyworld: life in a new context

[2] Farmer J. Doyne and Belin Alletta d’ A. , “Artificial life: the coming evolution” SFI WORKING PAPER: 1990—003 in Proceedings in Celebration of Murray Gell-Man's 60th Birthday.

[3] M. A. Bedau. 2007. Artificial life. In M. Matthen and C. Stephens, eds.,Handbook of the Philosophy of Biology, pp. 585-603. Amsterdam: Elsevier

[4] Bedau, Mark A., “Artificial life: organization, adaptation and complexity from the bottom up” Elsevier,TRENDS in Cognitive Sciences,Vol.7 No.11 ,November 2003

[5] Paul E. Black, "Manhattan distance", in Dictionary of Algorithms and Data Structures [online], Vreda Pieterse and Paul E. Black, eds. 31 May 2006.

[6] Harp, S., T. Samad, and A. Guha (1990), “Towards the Genetic Synthesis of Neural Networks” In Proc. Third Intl.Conf. on Genetic Algorithms, edited by J. D. Schaffer, Morgan Kaufmann, San Mateo, CA, 1990

[7] de Boer, M. J. M., F. D. Fracchia, and P. Prusinkiewicz (1992), “Analysis and Simulation of the Development of Cellular Layers” In Artificial Life II, edited by C. Langton, C. Taylor, J. Farmer, and S. Rasmussen. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. X. Addison-Wesley, Redwood City, CA, 1992

[8] Chalmers, D. (1991), "The Evolution of Learning: An Experiment in Genetic Connectionism" In Connectionist Models, Proceedings of the 1990 Summer School, edited by D. S. Touretzky, J. L. Elman, T. J. Sejnowski, G. E. Hinton, Morgan Kaufmann, San Mateo, CA, 1991.

[9] Nolfi, S., J. L. Elman, and D. Parisi (1990), "Learning and Evolution in Neural Networks", CRL Tech. Rep. 9019, Center for Research in Language, UCSD, La Jolla, CA, 1990.Humphries, M. D., Gurney, K., and Prescott, T. J. (2006). The brainstem reticular formation is a small-world, not scale-free, network. Proc. R. Soc. B, 273:503–511.

[10] Parisi, D., S. Nolfi, and F. Cecconi (1991), “Learning, Behavior, and Evolution”, Tech. Rep. PCIA-91-14, Dept. of Cognitive Processes and Artificial Intelligence, Institute of Psychology, C.N.R. - Rome, June 1991. (To appear in Proceedings of ECAL-91—First European Conference on Artificial Life, December 1991, Paris).

[11] Renault, O., N. M. Thalmann, and D. Thalmann (1990), “A Vision-based Approach to Behavioural Animation”, J. of Visualization and Computer Animation Vol. 1, 18-21.

[12] Cliff, D. (1991), “The Computational Hoverfly; a Study in Computational Neuroethology” In From Animals to Animats, Proceedings of the First International Conference on Simulation of Adaptive Behavior, edited by J.-A. Meyer and S. Wilson. A Bradford Book, MIT Press, Cambridge and London, 1991.

[13] McShea, D. W. (1996). Metazoan complexity and evolution: Is there a trend? Evolution, 50:477–492.

[14] Yaeger, L. S., Griffith, V., and Sporns, O. (2008). Passive and Driven Trends in the Evolution of Complexity. In Bullock, S. e. a., editor, Proceedings of the Artificial Life XI Conference, MIT Press, Cambridge, MA.