Comparative Analysis of Movement Strategies in a Simulated Grid World

(Genetic and Reinforcement Learning Approaches)

Daniyaruly Danial
School of Information Technology and Engineering
Kazakh-British Technical University
Almaty, Kazakhstan
Email: d_daniyaruly@kbtu.kz

I. Introduction

Agent-based simulations are useful for studying emergent behavior in complex systems. In our simulation, agents navigate a 25×25 grid, search for food, expend energy based on movement and intrinsic traits (speed and size), and reproduce when energy exceeds a threshold. This paper compares two approaches:

1) **Genetic Algorithm (GA) with Food-Driven Movement:** Agents check neighboring cells for food and move to the one with the highest energy. If no food is found, they randomly decide to move or stay.
2) **Reinforcement Learning (RL) Approach:** Agents have 9 options for actions (8 directions plus "stay"). The RL agent updates its policy based on food rewards and energy costs.

II. Methodology

A. Environment Simulation

The environment is a 25×25 grid containing a fixed number of agents (20) and food items (80). Food items provide energy (values 25, 35, or 45) and respawn randomly after each generation. Multiple agents or foods may occupy the same cell simultaneously. Food can also spawn in the same cell as the creature.

Each creature is characterized by the following parameters:

- **Speed:** An integer value (1, 2, or 3) determining how many movement steps the creature can take.
- **Size:** An integer value (1, 2, or 3) that influences both the creature's maximum energy and the cost of movement.
- **Energy:** Initially set to the maximum energy, computed as
$$E_{\max} = 150 + (\,\text{Size} - 1\,) \times 50.$$

Energy is expended during movement and reproduction. The movement energy cost is computed as follows:

- A baseline cost is given by $10 \times \text{Size}$.
- The maximum cost is $(\text{Speed} + \text{Size}) \times 10$.
- The actual movement cost is calculated as:

$$E_{\text{move}} = \text{baseline} + \\ + \text{round}\left(\frac{(\text{max\_cost} - \text{baseline})}{\text{Speed}} \times \text{num\_moves}\right) \tag{1}$$

Reproduction occurs if a creature's energy exceeds a threshold defined by:

$$E_{\text{threshold}} = 100 + (\text{Size} - 1) \times 50.$$

The reproduction cost is set as $15 \times (\text{Speed} + \text{Size})$ both in the GA model and in the RL model. Upon reproduction, the parent's energy is reduced by this cost, and the offspring is generated with 70% of its maximum energy. Offspring traits are inherited from the parent with potential mutations governed by a mutation rate of 0.1.

B. Movement Strategies

1) Genetic Algorithm (GA) Strategy: In the GA approach, each creature determines its movement by scanning adjacent cells (including the current cell) for food. The creature moves toward the cell with the highest food yield, or, if no cell has food, it moves randomly. After movement, creatures consume any food available in their new cell to replenish their energy. The GA evolution is conducted in epochs; each epoch consists of 50 generations (movements). At the end of each epoch, the population is evolved using classical GA operators:

- **Selection:** A tournament selection based on offspring count.
- **Crossover:** A uniform crossover is used to combine parental traits.
- **Mutation:** Each trait (speed and size) is mutated with a probability of 0.1.

The new population is then reinitialized with the same size (20 creatures) and assigned new random positions for the subsequent epoch.

2) Reinforcement Learning (RL) Strategy: In the RL approach, each creature is provided with an initial RL policy, which is a probability distribution over 9 possible actions (8 directional moves and one *stay* action). In every movement step (up to the creature's speed), the creature:

1) Scans the eight neighboring cells as well as its current cell to evaluate potential food rewards.
2) Combines the scanned food values with its current RL policy (weighted by a scan factor of 0.25) to determine effective action probabilities.
3) Selects an action based on these probabilities.

4) If the target cell contains food, the creature consumes it and gains energy equivalent to the sum of food values.
5) Updates its policy immediately using a learning rate of 0.1 and the immediate reward received.

Reproduction in the RL model is triggered under conditions analogous to the GA model. The simulation is similarly structured in epochs, and at the end of each epoch the best-performing policy (measured by the number of successful reproductions) is used to generate a new population (with mutation) for the next epoch.

III. Experimental Results

Under the default parameter settings and energy computation methods, the experimental results indicated that the Genetic Algorithm (GA) outperformed the Reinforcement Learning (RL) approach in terms of population dynamics, as illustrated in Figure 1. However, the RL method is closely followed in overall performance.

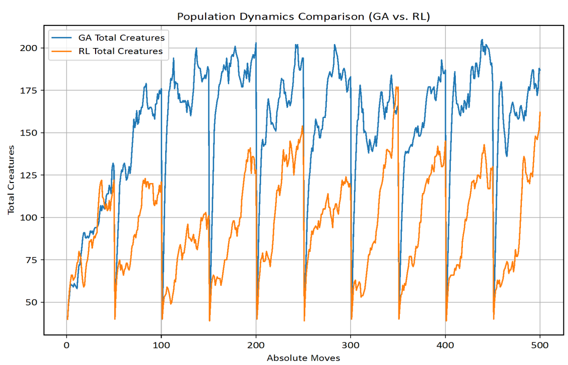

Fig. 1. Population trends over 10 epochs.

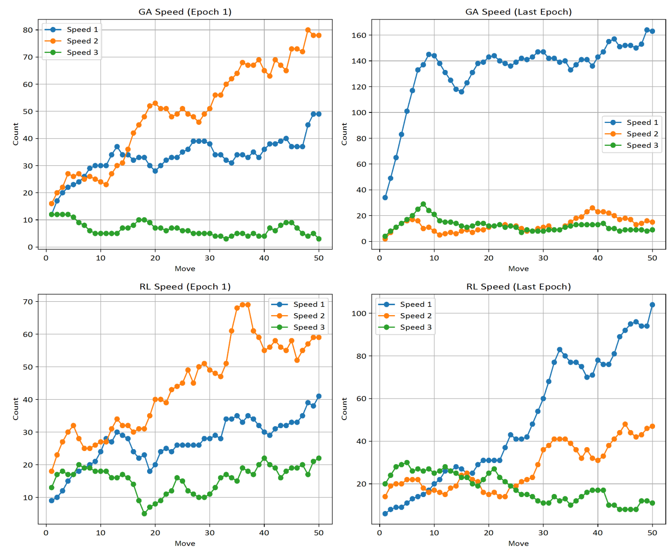

Fig. 2. Distribution over speed types.

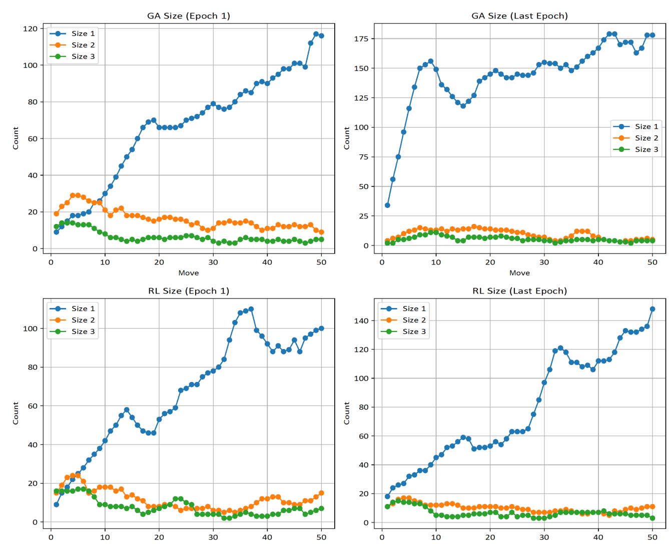

Fig. 3. Distribution over size types.

Figure 2 and Figure 3 presents the distributions of creatures based on speed and size. Both algorithms favored creatures with a speed of 1 and a size of 1. Notably, in the final epoch, the RL method produced a higher proportion of creatures with speeds of 2 and 3 compared to the GA.

Figure 4 shows the cumulative counts of reproduction and death events per epoch. Because the GA maintained a larger population in each epoch, it also exhibited higher numbers of both reproduction and death events.

Further experiments were carried out by changing key parameters:

- **Increasing the number of moves** per epoch from 50 to 80 improved performance for both algorithms, yielding an increase of approximately 25 creatures per epoch.
- **Changing the number of epochs** from 10 to 30 did not lead to significant differences in the results.
- **Reducing the amount of available food** from 80 to 40 substantially decreased the maximum number of creatures per epoch (by approximately a factor of 3), with both models continuing to favor creatures with speed 1 and size 1.
- **Increasing the food respawn duration** from 1 to 3 produced results similar to those observed when reducing the food quantity.
- **Reducing the energy value of all food types** by 5 points resulted in a significant decrease in the maximum number of creatures per epoch, approximately by a factor of 1.5. Additionally, there was a notable change in the speed distribution of the creatures (RL) in the final epoch, leading to an equalization in the number of creatures across different speed types in that last epoch.
- Conversely, **increasing the food energy** by 5 points increased the maximum number of creatures per epoch by roughly 1.5 times, although both algorithms still favored creatures with speed 1 and size 1.

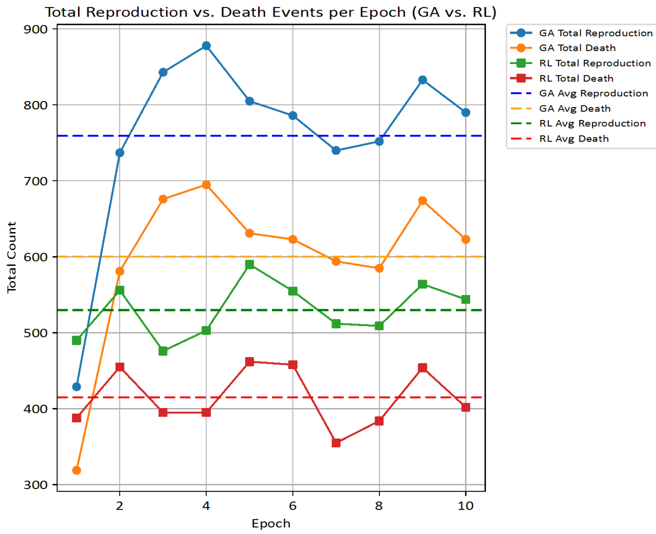

Fig. 4. Total Reproduction vs. Death Events per Epoch (GA vs. RL).

- **Reducing the movement cost** from $(\texttt{self.speed} + \texttt{self.size}) \times 10$ to $(\texttt{self.speed} + \texttt{self.size}) \times 5$ doubled the number of creatures per epoch. The GA showed a slight increase in creatures with speed 2, while the RL method exhibited a significant change: In the final epoch, the number of creatures with speeds 2 and 3, as well as sizes 2 and 3, reached a 1:2 ratio relative to those with speed 1 and size 1. Additionally, the GA experienced a notable reduction in death events.
- **Increasing the movement cost** to $(\texttt{self.speed} + \texttt{self.size}) \times 15$ decreased the maximum number of creatures per epoch by roughly a factor of 3. Interestingly, in this scenario, the RL method showed a higher number of creatures with speed 2 compared to the other speeds.
- **Increasing the mutation rate** from 0.1 to 0.25 improved the diversity of creature speeds, although creatures with size 1 remained dominant.
- **Reducing the reproduction cost** from $15 \times (\texttt{self.speed} + \texttt{self.size})$ to $10 \times (\texttt{self.speed} + \texttt{self.size})$ yielded slightly better results; In the final epoch the RL method did not exhibit a wide spread in speed categories.
- In contrast, **increasing the reproduction cost** to $20 \times (\texttt{self.speed} + \texttt{self.size})$ slightly reduced the maximum number of creatures per epoch.

IV. Discussion

The experimental results show that, under default parameters, the GA approach consistently produced larger population sizes compared to the RL strategy. In the GA framework, a greater number of creatures per epoch led to more reproduction and a higher overall number of deaths, showing its aggressive evolutionary pressure.

Although the RL approach had slightly fewer creatures overall, it still demonstrated competitive performance with a distinct pattern in trait evolution. Both methods favored creatures with a speed of 1 and a size of 1. However, in the later epochs, the RL method displayed a higher proportion of creatures with speeds of 2 and 3.

Parameter sensitivity experiments further revealed that:

- Increasing the number of movement steps per epoch (from 50 to 80) improved the population growth for both algorithms.
- Reducing the food availability from 80 to 40 or increasing the food respawn duration from 1 to 3 generations significantly decreased the maximum population size (by about a factor of 3), with both methods still favoring lower speed and size values.
- Adjustments in food energy values had a proportional effect on the maximum population—reducing energy values decreased population size by approximately 1.5 times, whereas increasing them had the opposite effect.
- Lowering the movement cost from $(\texttt{Speed}+\texttt{Size}) \times 10$ to $(\texttt{Speed} + \texttt{Size}) \times 5$ doubled the population size. Notably, the RL strategy exhibited a pronounced shift in its trait distribution, with the final epoch showing a higher proportion of creatures with speeds 2 and 3 and sizes 2 and 3, compared to those with speed 1 and size 1.
- Higher mutation rates and modifications in reproduction costs further affected diversity and stability, with a higher mutation rate enhancing speed diversity and a lower reproduction cost slightly improving overall performance.

These findings suggest that while the GA approach remains robust in ensuring higher population numbers under default settings, the RL strategy offers greater adaptability in trait evolution. Moreover, the sensitivity of both methods to parameter variations highlights the importance of careful parameter tuning in evolutionary simulations.

V. Conclusion

This study compares Genetic Algorithms (GA) and Reinforcement Learning (RL) in a simulated evolution environment. GA generally leads to higher population growth and reproduction rates, while RL enhances adaptive behaviors in traits like speed and size. Both strategies are sensitive to changes in food availability and movement costs, with reduced food and increased costs resulting in lower populations, while higher mutation rates foster more diverse movement strategies. Overall, GA excels in growth, whereas RL provides superior trait adaptation, suggesting potential for hybrid methods that combine their strengths in complex environments.

Code Availability

The source code for the simulation is available online at: https://github.com/daniyarulydanial/evolution_simulation.git