EvolveAgent4kids: a Novel Game-Based Tool for Teaching Genetic Algorithm in K-12

I. Abstract

The paper describes the process of designing and developing an interactive game-based tool called ”EvolveAgent” that focuses on introducing fundamental concepts of genetic algorithms(GA) to middle school students. As Genetic Algorithm (GA) is an advanced concept in today’s digital age, it is important to start the learning process from simple applications that explore the complicated ideas of GA with ease and simplicity. The goal of this project is to provide a high-level visualization of genetic evolution in a simulated environment where students will tweak parameters like population size, lifespan, and mutation rate to successfully evolve digital creatures. At the heart of this tool, the tool applies ”survival of the fittest” algorithmically.

Keywords: Genetic algorithms, AI education, educational games, K-12 learning, interactive learning tools

II. Introduction

In nature countless generations of living organisms have evolved from simple beginnings into complex forms starting from unicellular organisms to multicellular advanced species like humans. Genetic Algorithms (GAs) applied in the research borrow this inspiration from natural evolution-the idea that a population of organisms can adapt and improve over time through selective pressure.

GA algorithmically replicates biological evolution in nature and incorporates it inside a computer such as Darwin’s principles are replicated to guide search of simulated agents towards optimal solution [1]. In high level view, GA mixes and mutates the traits of the fittest individuals to breed new offspring. This meta-heuristic approach is the aesthetic charm of modern-day computing. It allows refined solutions over time for problems where the search space is very vast.

The approach of Genetic Algorithms is good at finding near-global optima where classical optimization techniques fail to improve over time or take up a lot of time as they get trapped in local optima. This provides GA an advantage to escape the dead-ends and discover more globally optimal solutions [2]. In fact, GA explicitly mimics the “survival of the fittest” process in nature, making the search significantly faster and more robust by continually favoring better candidates while still exploring new variations [2]. This balance between exploitation (selecting the best solutions) and exploration (injecting random mutations) allows genetic algorithms to solve tough problems ranging from scheduling and design optimization to machine learning tuning. Ever since the rise of GA in late 20th century, it has been applied as a general-purpose strategy for global optimization [2] [1] to problems where exact solution is hard to pinpoint with calculus or brute force. In such scenarios, GA evolves a good solution by trial, error, and adaptation.

A rich amount of research on GA highlights its versatility and effectiveness over traditional methods as well as incorporating with cutting edge technologies. In the research “Application of GA with integrated Machine Learning”, Raj et al. (2023) demonstrates how integrating GA with machine learning techniques can tackle both unconstrained and constrained optimization problems [3]. The research blended intelligent models and evolutionary search algorithms and yielded superior results in fields like predictive analytics and AI model training. In another research, Thong-ia and Champrasert (2023) combined a GA with the Ant Colony Optimization (ACO) algorithm to solve an NP hard problem – Traveling Salesman [4].

Traveling Salesman problem is a well-known NP hard problem that as per current technological capabilities is almost impossible to solve with increasing number of cities. So, while searching for the optimal path, their “Gene-Ants” system injected genetic crossover and mutation operations into the ant colony’s path-finding process to avoid premature convergence. The Gene-Ants algorithm outperformed a standard ACO in finding shorter routes with even faster convergence towards global optimum [4]. Such exemplary research makes a solid point that genetic algorithms, whether used on its own or incorporation with other methods, embody a powerful idea of learning through evolution in optimizing even powerful technologies like artificial intelligence [3] [4].

III. Motivation and Research Question

In contrast, the power and elegance of genetic algorithms can seem abstract and intangible to young students. Everyday experience of a young middle school student does not use the concept of a population of solutions evolving over generations. They are foreign to terms like mutation rate, crossover, selection pressure, or simply evolution. **How to convey the intuition of survival of the fittest in a way that a middle school student finds intuitive and fun?** Traditional slides and equations can leave the students even more confused and unconvinced that these algorithms will ever be seen in real life. The paper takes on the challenge to teach such abstract concepts to middle school students using creative and hands-on approach. Research on [5] an educational game, “SnakeAI” has found that game-based learning and interactive simulations can bridge the gap between complex theory and intuitive understanding. The playful experiment gives Students opportunity to turn their confusion into curiosity. The paper introduces a similar tool called "EvolveAgent" that is playful, interactive, and easy to use.

Studies have shown that concepts like AI ethics and machine learning are difficult to grasp for younger students [6], [7]. EvolveAgent is a biologically inspired game designed to bridge the gap between genetic algorithms and middle school students. EvolveAgent transforms complex evolutionary concepts into engaging interactive simulation where students can visualize agents evolving over time to find their goals. Students using EvolveAgent step into the role of "nature's experimenter": they set up a virtual environment where simple digital creatures or agents must adapt to survive and succeed at a given task. These agents start off performing the task rather clumsily (much like random initial solutions in a GA), but with each new generation the game shows how the agents improve.

The same principles of selection, crossover, and mutation that drive genetic algorithms are at work – except now the students can see them in action. For instance, learners can adjust the mutation rate or selection criteria via intuitive sliders, and then immediately observe how those changes affect the agents' evolutionary progress. A population of colorful little agents wander around on screen, some thriving, some failing, and gradually the whole group becomes better at their task. This dynamic visualization turns abstract terms into understandable events: mutation becomes that slight random tweak in a creature's behavior that sometimes leads to a breakthrough, and selection is witnessed as only the best performers get to parent the next generation. Students gain an intuition for how randomness and selection can solve problems over time. Thus, EvolveAgent promises to guide students to experiment, observe, and draw conclusions about evolution in computation.

Genetic algorithms take inspiration from nature's evolutionary genius and EvolveAgent conveys these ideas to learners in a simplified approach as these concepts require more than conventional teaching. EvolveAgent is an attempt to humanize and demystify artificial evolution by casting it into a game that is both reflective and fun. In the following sections, the paper discusses in detail the related works that guided creation of the tool in hand and the design of the tool [1].

IV. Related Work

A. AI Ethics and Societal Impact Tools:

In this paper, Rayavaram et al. introduces an interactive self-driving car simulation that helps K-8 students understand the ethical considerations and societal impacts of AI [8]. The tool allows students to first choose from a list of pre-trained car vision models and then place obstacles like traffic cones, animals, and pedestrians in the driving scenario [8]. Students observe how a well-trained model reacts to the obstacles versus how a poorly trained model is susceptible to missing the obstacles that it was not trained on. Students can see the biases of training data leading to unsafe outcomes [8]. The tool clearly emphasizes the need for comprehensive and diverse training data in AI systems like autonomous vehicles that humans might rely on in a few years [8]. In evaluations, over 84% of participating middle-schoolers understood that poor training choices can negatively affect an AI's behavior, and almost 96% recognized why using diverse data makes the AI fairer and more reliable [8].

B. Conversational AI in Educational Games:

In this paper, Chen et al. explored a different angle by integrating a generative AI (GenAI) agent into a middle-school learning game [9]. In their history-themed game, students first learned new material from a virtual mentor (an AI character) and then switched roles to teach a virtual student (another AI agent) named Sun Bin [9]. This role-play scenario was designed to immerse learners in a dialogue-based learning experience, where they practice knowledge by conversing with AI. Analyzing 365 recorded dialogues from 76 students, the authors found that learners' interactions fell into various categories (social chat, questions, ideas, etc.), but contained relatively few factual explanations – likely because the AI couldn't provide many authoritative answers for the students to build on [9]. Moreover, the majority of emotions expressed during these exchanges were negative (about 82% of emotional remarks) as students sometimes grew frustrated when the AI misunderstood them, repeated itself, or didn't meet their expectations [9]. Thus, the study creates interactive learning moments with GenAI in the classroom but faces challenges in maintaining student trust and positive engagement [9]. Previous research has shown that students often respond positively to AI-driven characters in educational settings, even preferring them over human instructors in some cases [10].

C. Game-Based Learning of AI Concepts:

In this paper, Priyanka et al. focuses on teaching young students the concept of reinforced learning through an interactive game-based tool called TrainYourSnakeAI [5]. The tool engages middle school students in training an AI agent to play a popular snake game. It illustrates students how AI can learn with trial and error and how certain rewards and penalties parameters can change the behavior of their AI agent. The snake game was successful in keeping the students excited and involved while the research team experimented with the benefits of learning in a game-based environment for students [5]. Eventually, students gained an intuitive understanding of reinforced learning concepts as most participants were able to later explain in their own words how an agent learns from feedback [5]. The study suggests that abstract concepts like RL can be introduced to middle school students with the right interactive approach.

V. Game Design

EvolveAgent is a genetic algorithm framework that simulates survival of the fittest among a batch of simulated agents that navigate toward a goal. Agents have a life expectancy of 5 generations of offspring, which implies that even the best surviving agents are regenerated with mutated DNA after 5 successful generations. The goal to be found which is a green-colored object in the grid, is found on the lower right corner of the screen. The agents are generated at the top left corner of the screen. The best-performing agents in their lifespan release pheromones if they are moving closer to the goal. When such best-performing agents reach their life expectancy, their final position will have a strong pheromone scent. The new offspring are instinctively attracted towards the high pheromone places and then continue their exploration from the high pheromone regions. In this way, the new offspring will not be stuck exploring the already explored regions of the simulation. In this manner, the agents will continue to reach closer to the goal by creating a trail of pheromones. This phenomenon is naturally occurring in ant colonies. There already exists ant colony optimization algorithms in computer science but in this tool, we are focusing on just utilizing the pheromone trail characteristic of ants to simulate a different species of agents. The design of the game is explained in detail in the following sub-sections.

A. Initial Setup: Agents, Environment, and Goal

The simulation area is a 60×60 grid, giving agents plenty of room to explore. All agents start at coordinates (0,0) in the top-left corner, while the goal is located at the bottom-right corner by default. This placement ensures that agents have ample opportunities both to fail and to succeed, since some cells are walls that force them to navigate around obstacles. The user's chosen population size determines how many bots are created, and each bot is assigned a random DNA sequence that governs its movements. With a large grid and the possibility of wandering into dead ends, it often takes several generations for at least some agents to move closer to the goal. In this way, we place agents as far from the goal as possible, ensuring the need for multiple generations of exploration before they can discover and successfully navigate to the destination.

B. Generational Turnover and Lifespan

As we have already discussed, each bot has a lifespan of 5 generations. If it survives that long in the simulation, it is effectively retired even if it was performing well. This adds a natural turnover and prevents occurrence of immortal agents that are always the chosen ones to find the goal. The idea is derived from the natural phenomenon of how every human is replaced by the next generations of humans with increased amount of information about innovation and knowledge. Every time a new generation of bots is produced, the best performing agent's age increases by one. Once it hits its lifespan limit, the bot is removed from the simulation. The best performing bots can also be removed from the simulation at ages 2 or 3 or 4 if they are surpassed by better performing offspring that outclassed them before they reached their lifespan of 5 generations. The turnover ensures that no single bot or lineage dominates forever. After every 5 generations of offspring, the entire simulation is replaced by fresh mutated agents that have the instincts of their ancestors. This keeps the simulation lively by increasing exploration, potential flaws and despair. It also ensures that innovation continues every generation without the population relying on a single immortal "super bot". Hence, the setup is idealizing how human societies move forward with fresh minds and evolving knowledge.

C. Evaluating Success and Selecting the Best

Only the first generation of agents are ignorantly wandering and exploring the whole grid. Soon, the agents realize that each one of them is programmed to find the goal and each agent has a fitness measure, which is measured by its Euclidean distance to the goal. Agents realize that the closer it is moving towards the goal, the better. The agents are sorted by distance to the goal; the top performers are recognized as the best agents. A certain number of these best performing agents are carried over directly into the next generation unless, of course, if they exceed their lifespan. The simulation keeps track of a global best distance that updates whenever a new bot outperforms the current best record.

D. Pheromone Release: Reinforcing Progress

So, one might ask, then how does the agent find the goal if they are replaced by their offspring? As we have already briefly mentioned about pheromone trails, the bots soon realize that each time they move they deposit a small amount of pheromone. If a top-ranked agent is moving closer to the goal, it drops extra pheromones, whereas if an agent is moving farther away from the goal, its pheromone drop decreases. This ensures that only good moving agents are rewarded with pheromone trait and agents that do not support finding the goal do not interfere with the new offspring that relies on the pheromone trail to find the goal. This is a possibility because due to mutation, some agents might have no idea about the purpose of finding the goal. Hence, this additional subtlety ensures that only the best performers are followed. After each step and at the end of each generation, pheromones gradually fade away. This is an extra caution to avoid stale trails from misleading new agents. The most surprising pheromone trait is that once an agent that has been performing well reaches its lifespan limit, it is removed from the simulation but it deposits a hefty dose of pheromone at its final position. This big pheromone boost makes that location more attractive for future offspring. So, even though the super agent candidate is gone, it leaves behind a strong guiding trail. It ensures that what the agent achieved does not just disappear - it becomes a stepping stone for further progress.

E. Inherited Instincts and Directional Learning

When new agents are spawned, there is an 80 percent chance that an agent is created by mixing the DNA of two survivors of the best parents. The remaining 20 percent of the time, the agent is spawned with a random DNA sequence. Agents are also diversified with a defined mutation rate that can be changed by the user during simulation. New generations retain partial knowledge from parents, but are also nudged to try unexplored approaches, ensuring that they do not get stuck in local optima. Like in nature, offspring are never

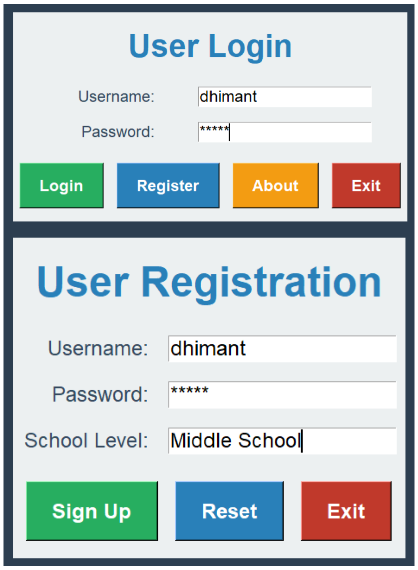

Fig. 1: *Login and registration interface.*

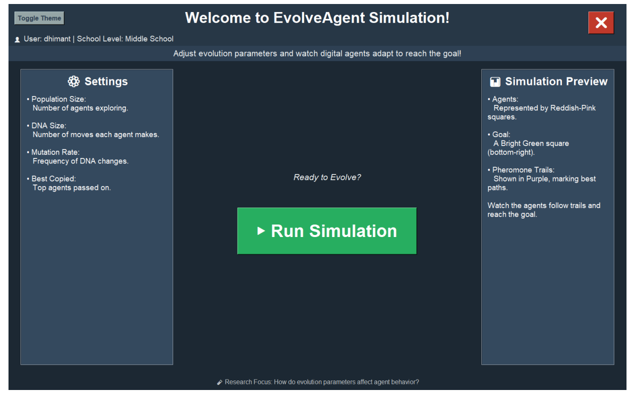

Fig. 2: *Simulation settings and preview screen*

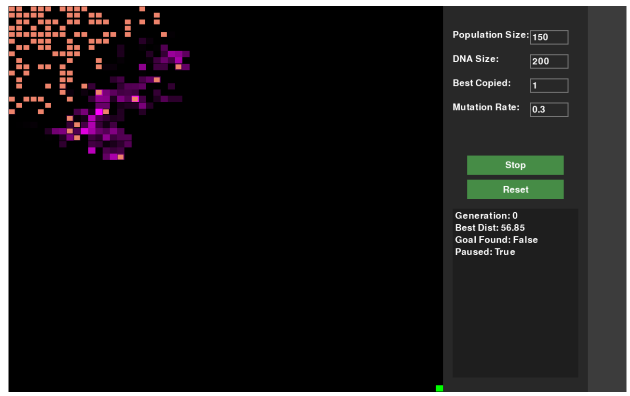

Fig. 3: *Agents evolving through pheromone trails.*

carbon copies of their parents. They are close enough to carry forward successful traits, but random differences bring genetic diversity.

F. Nature-Inspired Design

EvolveAgent is biologically inspired and uses natural analogies in its algorithm. The tool is inspired by the Ant Colony Optimization(ACO) algorithm but does not explicitly follow its framework. Implements only the ACO pheromone signaling from ACO and illustrates how nature-inspired concepts can create engaging and adaptive simulations. The resulting synergy between Darwinian selection and pheromone-based guidance is a highlight of the power of nature-inspired tool design. EvolveAgent aims to ignite curiosity in students to explore intersections between the computing world and the natural one.

VI. User Interaction and Experience

After launching the application, the tool requires users to register with their username, school level, and password. After signing up, users log into the tool using their credentials they just created. Upon logging in, users are navigated to the homepage, where they will have information about the simulated agents in action in the game. The homepage contains a detailed description that informs users about the mechanics of the simulation. After understanding the required information about the tool, the users can navigate to ”Run Simulation,” where users can create a population of agents, tweak their DNA size, mutation rate, and best copied agents. The changeable parameters in the simulation are responsible for directly affecting the successful and failed evolution.

The simulation allows for adjustment of key parameters that influence the behavior of evolving agents.

- **Population Size:** The total number of agents initialized at the start.
- **DNA Size:** The number of moves each agent can make before being evaluated.
- **Mutation Rate:** The probability that any given move in an agent’s DNA will be altered randomly.
- **Best Copied:** The number of top performing agents directly transferred to the next generation.

After simulation, the user can review the final results that explain the cause and effect of successful and failed digital evolution over generations. This approach helps ensure the tool remains relevant, practical, and easy to adopt in classroom settings as prior work has emphasized the importance of aligning educational tools with curriculum standards by working directly with teachers [7].

VII. Conclusion

EvolveAgent shows how an interactive game-based tool can be used to help middle school students learn about genetic algorithms in a fun and interactive way. Its use of a game-like simulation where digital creatures evolve over time is intuitive, simple, and easy to understand. Students interact by adjusting simple settings such as mutation rate, dna size, and, population size and observe how these changes affect the way agents behave.

EvolveAgent aims to make a complex topic clear and intuitive by conversion of confusion to curiosity. The simulation correctly shows how the ideas of selection, crossover, and mutation work in a step-by-step manner. The tool's simplicity broadens its reach to students that may not be expected to have a career in computer science.

Our study shows that interactive tools like EvolveAgent can improve student understanding of advanced ideas such as GA, AI ethics, or RL. This tool engages students and makes learning active and visual. In future work, we plan to add more topics and interactive features into the tool, and we also hope to work more closely with educators. EvolveAgent demonstrates that difficult subjects can be taught in a simple and engaging way. It proves that the idea of evolution and the naturally occurring algorithms of nature can be brought into the classroom.

VIII. Future Work

Future development of EvolveAgent will focus on expanding its educational scope and increasing interactivity. One major thrust will be to extend content beyond genetic algorithms and cover other foundational concepts of AI. Basic machine learning experiments or a few words on AI ethics would place genetic algorithms in the larger perspective of artificial intelligence. Incorporating a wider range of AI literacy topics could strengthen EvolveAgent's scope and curriculum. EvolveAgent can then help students become more informed about how AI operates in society and encourage them to think critically about technology.

The integration of a conversational virtual agent could guide students through the learning process by answering their questions, providing hints, and narrating the evolutionary steps occurring in the simulation. It would be helpful to students struggling with parameters and they could receive result analysis with the agent and figure out the potential reasons for failure in their simulation. They would not have to wait for the organizers to personally help them navigate through their problems. The students can maintain privacy for their problems and questions about the simulation. This helper AI friend moving along with the student during their simulation session could transform the tool from a static simulation to a highly engaging and effective learning tool as the number of good questions that the student can ask the AI helper is the deciding factor for their learning.

EvolveAgent can benefit from collaboration with K-12 teachers to align the tool with classroom needs and curriculum standards. It highlights the value of working closely with teachers to maximize student engagement. Thus, the research team will focus on broadening content coverage from GA to AI literacy, integrating conversational AI helper, and partnering with K-12 teachers, to make EvolveAgent a more powerful and versatile platform.

References

[1] A. Anwaar, M. Iqbal, A. Ashraf, and W. H. K. Bangyal, "Genetic algorithms: Brief review on genetic algorithms for global optimization problems," *IEEE Human-Centered Cognitive Systems*, pp. 1–7, 2022.

[2] Z. Si-ru, "Application of improved genetic algorithm in optimization computation," in *Proceedings of the Fifth International Conference on Natural Computation (ICNC)*, 2009, pp. 144–148.

[3] A. Raj, A. Kumar, V. Sharma, S. Rani, A. K. Shanu, and T. Singh, "Applications of genetic algorithm with integrated machine learning," in *Proceedings of the 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, 2023, pp. 1–6.

[4] S. Thong-ia and P. Champrasert, "Gene-ants: Ant colony optimization with genetic algorithm for traveling salesman problem solving," in *Proceedings of the International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*, 2023, pp. 1–5.

[5] C. Hinojosa, P. Kumar, P. D. Rajarajan, and F. Martin, "Trainyoursnakeai: A novel tool to teach reinforcement learning to middle school students," in *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, ser. SIGCSETS 2025. New York, NY, USA: Association for Computing Machinery, 2025, p. 506–512. [Online]. Available: https://doi.org/10.1145/3641554.3701907

[6] I. Lee, "Developing ai literacy in k-12," in *Proceedings of the Annual ACM Conference on Human Factors in Computing Systems*, 2021, pp. 1–10.

[7] D. S. Touretzky, C. Gardner-McCune, D. Seehorn, and F. Martin, "Teaching tech to talk: K-12 conversational artificial intelligence literacy curriculum and development tools," in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2021, pp. 15 358–15 365.

[8] P. Rayavaram, S. Narain, and F. Martin, "Let's go for a drive: Exploring ai's societal impact in k-8 education with an interactive self-driving car tool," in *Proceedings of the IEEE Frontiers in Education Conference (FIE)*, 2024, pp. 1–9.

[9] Z. Chen, Z. Xiong, X. Ruan, S. Jiang, W. Wei, and K. Fang, "Exploring learners' interactions with genai agents in educational games: Typologies and emotional factors in human-computer interaction," in *Proceedings of the 4th International Conference on Educational Technology (ICET)*, 2024, pp. 136–140.

[10] M. M. Chan, H. R. Amado-Salvatierra, R. Hernandez-Rizzardini, and M. Rosales, "Assessing student perceptions of video quality and effectiveness from ai-enhanced digital resources," in *Proceedings of the IEEE Digital Education and MOOCS Conference (DEMOcon)*, 2024, pp. 1–5.