Application of the coevolution strategy to solve the problem of autonomous navigation through the maze

Iaroslav Omelianenko, Anatoliy Doroshenko, Yevheniy Rodin

Institute of Software Systems,
National Academy of Science of Ukraine

14-15 May 2024

Introduction

The utilization of an evolutionary algorithm (EA) for navigation through a complex environment essentially involves determining the most efficient route to reach a specified goal. This process hinges on formulating an objective function (also known as a fitness function) that is designed to be either minimized or maximized.

Earlier research has highlighted a significant issue encountered in practical applications. Although the goal of a task might be clearly established and well known, the definition of the objective function could be misleading. This issue is exemplified by the challenges faced when constructing a control model for an autonomous agent tasked with navigating a two-dimensional maze.

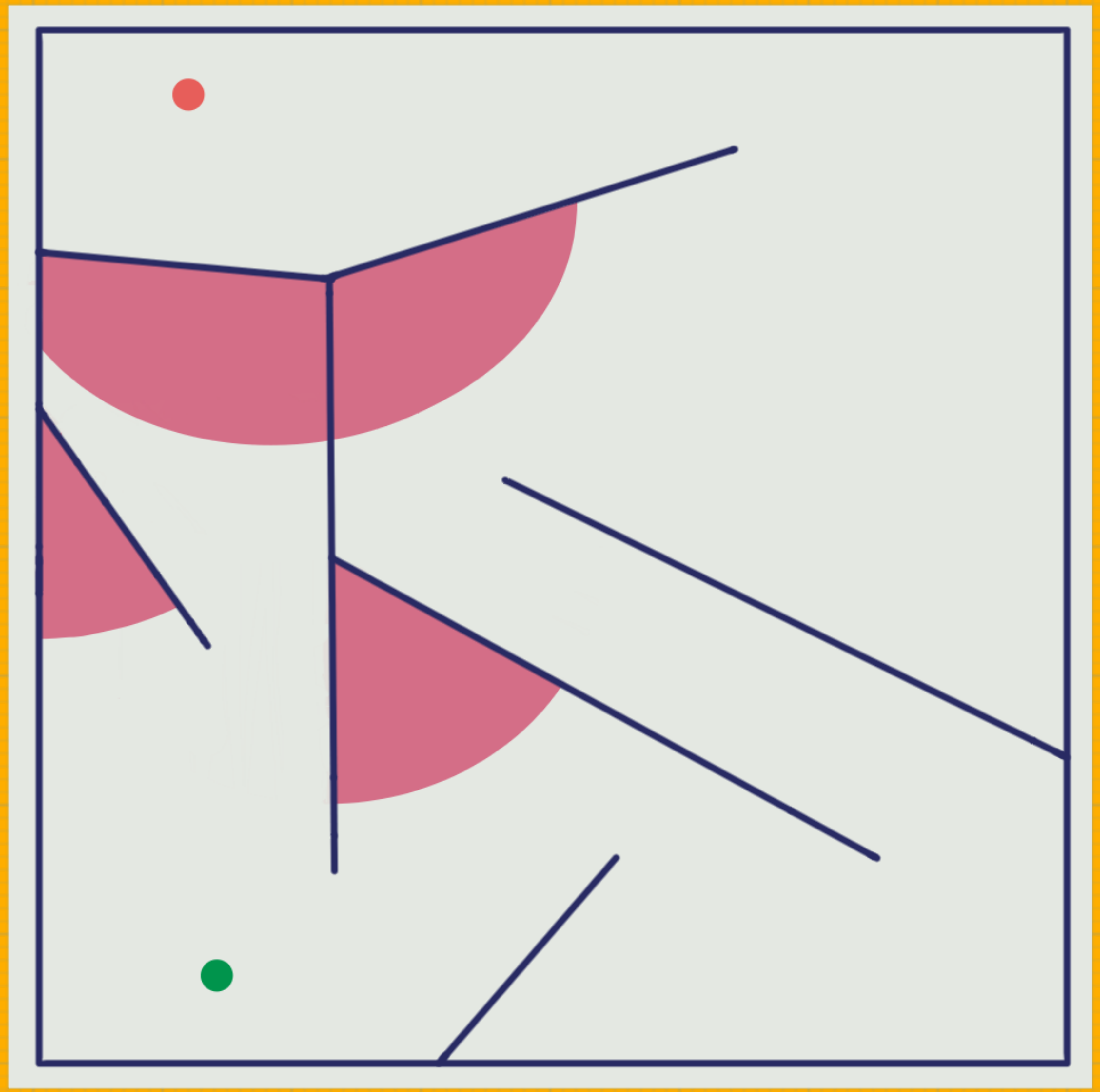
Here, the primary objective of the control model is to guide the robot so that it navigates the maze from the starting point to the exit within a predetermined number of moves.

Introduction

One might instinctively create a goal-driven objective function based on the robot's proximity to the maze exit. However, in a complex maze, the solver agent could increasingly struggle to make appropriate directional choices solely based on exit proximity.

The figure at the right demonstrates that a minimal distance to the exit doesn't necessarily indicate a clear path to it. The fitness function may present local optima within the maze's dead ends, marked by steep fitness score gradients.

In this case, we have a deceptive landscape of fitness score values, which a goal-oriented objective function can not solve.



Introduction

It can be asserted that while the control model may identify the optimal solution, it's not always discoverable through evolution with a straightforward goal-oriented objective function. Furthermore, the conventional wisdom in defining the fitness function is questionable, as it conflates the end goal (exit of the maze) with the design of the objective function (proximity to the exit).

To address this issue, the SAFE (Solution And Fitness Evolution) method was suggested, which decouples the control model's optimization from that of the objective function. Essentially, the solver agent's goal to find the maze exit doesn't imply that the objective function must be based on the distance to the exit. The proposed solution involves the co-evolution of two populations: the control models and the candidates for objective functions.

This work introduces an innovative application of the NEAT algorithm for evolving the solver agents (control models) and combines the NEAT algorithm with the Novelty Search (NS) method for evolving the objective function candidates. Additionally, it introduces a modified version of the NS method designed to narrow the search space for solutions, thereby enhancing overall efficiency.

Key features of coevolution

Coevolution can be described as a synergistic process where multiple lineages of distinct organisms evolve together in a way that is advantageous to each. Concurrently, the evolutionary trajectory of one species is inextricably linked to the presence of others. Throughout their evolutionary journey, coevolving species engage in interactions that influence and define their respective evolutionary tactics.

Coevolution encompasses three primary forms:

- mutualism, where multiple species live in harmony, each gaining advantages from the others;

- competitive coevolution:

    - predation, where one organism preys on and utilizes the resources of another;
    - parasitism, where one organism exploits another's resources without causing its demise;

- commensalism, a relationship in which one species benefits while the other species is neither helped nor harmed.

The final category of coevolutionary tactics has attracted interest among scholars for its potential in developing an effective approach to train autonomous agents.

Key features of the SAFE algorithm

The SAFE method, as suggested by its designation, revolves around the simultaneous evolution of both the solution and the objective function, which steers the solution search optimization.

It's predicated on the commensalistic coevolution strategy involving two distinct groups:

- a population of potential solutions that undergoes evolution to address the immediate problem;

- a population of objective function candidates that evolves with the purpose of refining the evolutionary path of the solution population.

This work advocates for the application of the NEAT algorithm to orchestrate the evolution of the population of potential solutions, combined with the NS method to enhance the evolutionary process within the population of objective function candidates.

Fitness function of the solver agent

During every stage of the evolutionary process, each maze solver agent undergoes assessment by all the potential objective functions from a separate commensalistic group. The highest fitness score achieved from these evaluations for each solver agent by any of the objective function candidates is then considered as the fitness score of the solution they represent.

The maze solver's fitness function integrates two measures: the distance to the maze exit (an estimation of proximity to the goal) and the uniqueness of the solver's end location (an estimation of novelty). These evaluations are arithmetically combined using a pair of coefficients derived from the output data of a specific member within the population of objective function candidates.

$$O_i(S_i) = a \times \frac{1}{D_i} + b \times NS_i \,, \quad (1)$$

Where $O_i(S_i)$ – is the fitness score obtained by evaluating the candidate solution S_i wrt objective function O_i. A pair of coefficients $[a, b]$ is the output of a specific candidate for the objective function. This pair determines to what extent the distance to the maze exit (D_i) and behavioral novelty (NS_i) of the solution affects the ultimate fitness score of the maze solver at the end of the trajectory.

Fitness function of the solver agent

The distance to the maze exit (D_i) is defined as the Euclidean distance between the final coordinates of the maze solver on its trajectory and the coordinates of the maze exit.

$$D_i = \sqrt{\sum_{i=1}^{2}(a_i - b_i)^2}\ , \qquad (2)$$

Where a – final coordinates of the maze solver and b – coordinates of the maze exit.

The novelty score NS_i of each maze-solving agent is determined by its final position in the maze (point x). It is calculated as the average distance from this point to the k-nearest neighboring points, which are the final positions of other maze solvers.

$$NS_i = \frac{1}{k}\sum_{i=0}^{k} dist(x, \mu_i)\ , \qquad (3)$$

Where μ_i – is the *i*-th nearest neighbor and $dist(x, \mu_i)$ – distance between x and μ_i.

The distance between two points is a novelty metric that measures how different the current solution (x) is from another (μ_i) produced by different maze solver. The novelty score is calculated as the Euclidean distance between two points:

$$dist(x, \mu) = \sqrt{\sum_{j=1}^{2}\left(x_j - \mu_j\right)^2}\ , (4)$$

Where μ_j and x_j – values at index *j* of coordinate vectors with coordinates of pints μ and x, respectively.

Fitness function of the candidates for objective functions

The SAFE method operates on the principle of commensalistic coevolution, which means that one of the coevolving populations is neither benefited nor harmed during the evolution. In our experiment, the commensalistic population is a set of candidates for objective functions. It's necessary to establish a fitness function for this population that is independent of the performance of the maze-solving agents (control models).

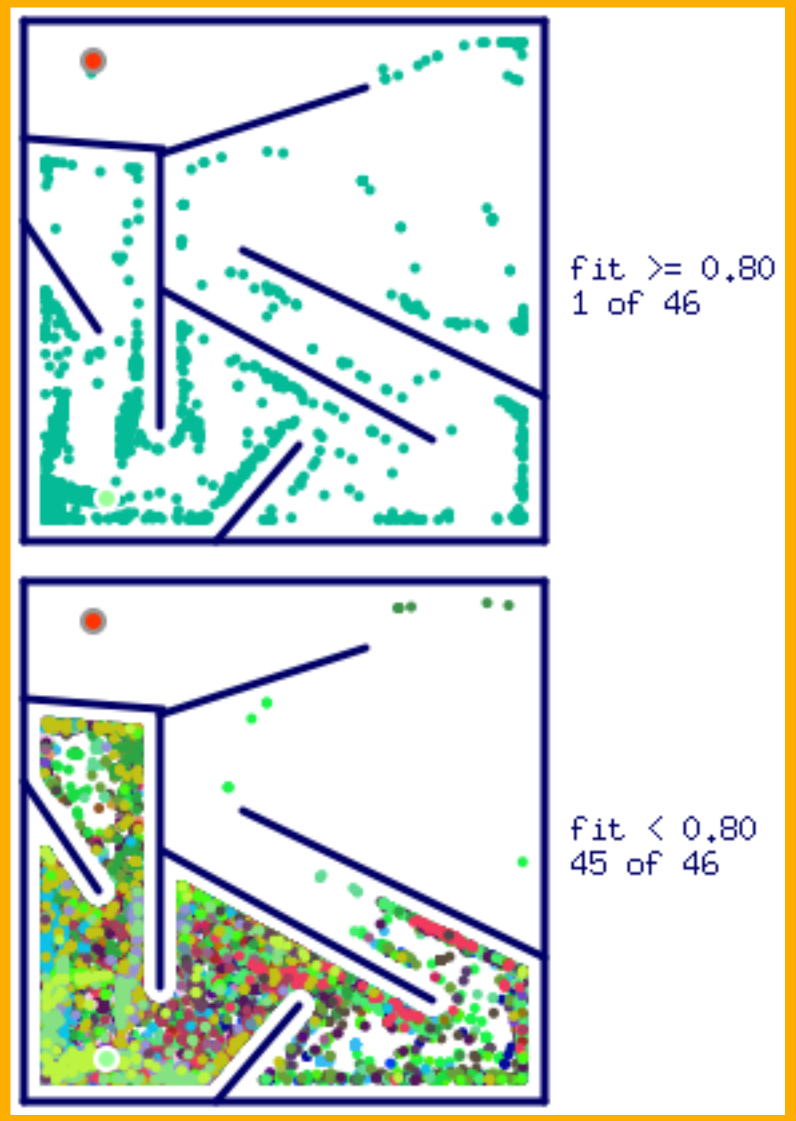
An appropriate choice in this context is an objective function that employs a novelty metric for determining fitness scores (NS_i). The formula for calculating the novelty score of each objective function candidate is the same as for the maze-solving agents (3). The only difference is that in the case of objective function candidates, we calculate the novelty score using vectors containing the output values $[a, b]$ (1) from each individual of the population. Subsequently, this calculated novelty score is utilized as the fitness score for the individual.

Experiment results

A successful solver agent was found after 211 generations of evolution and has the following configuration - 22 nodes connected by 47 links.

During the coevolutionary process, the optimal coefficients for the objective function [a, b] were identified. These coefficients, a = -0.53283 and b = 0.95889, were used to train an effective solver agent. Thus, formula (1) can be rewritten by substituting the found coefficients as follows:

$$O_i(S_i) = -0.53 \times \frac{1}{D_i} + 0.96 \times NS_i, \quad (5)$$



According to formula (5), it can be concluded that found objective function emphasizes training on the search for the most innovative solutions (NS_i), paying much less attention to the value of the distance to the maze exit (D_i). This confirms our thesis given at the beginning of this work that in a complex environment, a successful objective function is not the same as the distance to the goal (maze exit).

In the figure, we also see that most evolutionary losers are stuck in the local optima. At the same time, the successful solver agent showed the greatest innovativeness, giving preference to the research of new areas, as opposed to the use of already known ones (exploration vs exploitation).

Conclusions

This work shows how the use of the method of coevolution of two populations - a population of decision-making agents and a population of candidates for the objective function can be used to solve the problem of navigation in a complex maze. It has been experimentally proven that this method is more effective compared to the methods considered in previous works.

In contrast to earlier works, a novel method for executing the SAFE (Solution and Fitness Coevolution) method was introduced. This method involves utilizing the NEAT algorithm to manage the evolution process of two populations engaged in a commensalistic coevolution relationship. Also, the Novelty Search optimization method was used to guide the search within a deceptive landscape of possible solutions.

Finally, a software library was developed to facilitate a coevolution experiment using the GO programming language, as well as visualization tools that allow visual evaluation of coevolution outcomes as the input variables are modified.

https://github.com/yaricom/goNEAT_NS

Thank you!