273 Project 2 Report

Authors: Natalia Rivera, Nathalie Murphy, Yashesha Kothari, Joseph Gill, Andy Ho, Sibhi Sakthivel

Introduction | Escherichia coli (E. coli) are single-celled bacteria that are often found in the lower intestine of humans and other animals. Most strains are harmless, while some can cause serious illness and infection. E. coli is highly studied in the field of molecular biology and genetics because its high reproduction rate makes it very efficient to grow and work with in laboratory settings. E. coli's applications in medical and industrial use, such as in biopharmaceuticals, make it very important to understand cellular mechanisms and be able to predict its behavior. E. coli moves through a process called chemotaxis, a random walk that is biased by the location of a food source. This behavior is a stochastic process due to the randomness of the movement; its runs and tumbles vary based on the immediate environment. By writing a code that simulates a random, biased E. coli walk, taking into account a food source gradient and E. coli's short-term memory (about four seconds), we learn how to model biological behavior, a skill that is crucial to various fields such as drug design and synthetic biology.

Methods | In order to accurately simulate behavior that modeled E. coli's walk, we coded a biased random walk that alternated between tumble and run phases. This was plotted on a chemical gradient where the gradient maximum represented the highest concentration of the food source. The simulation cycled through three different steps. In the tumble step, the bacterium took four consecutive steps, randomized to represent the stochastic process of E. coli's movement. The next step estimated the change in concentration gradient by finding the difference in the chemical concentration at the current time from that of the previously recorded chemical concentration four timesteps earlier. Accounting for the change in chemical gradient every four timesteps mirrors how E. coli senses changes in concentration over short periods of time, in this case, approximated to be four seconds. Finally, the run phase is biased towards the gradient peak, at which the maximum food concentration is found. The direction of the run is dictated by the second step's gradient detection, in which it is determined if the bacterium finds itself closer to the food source from the last-recorded location. Moving towards the maximum food concentration creates a bias in the bacterium's movement. This overall behavior is known as gradient ascent because the gradient bias leads to an overall net movement towards the maximum concentration point.

A key component of this methodology is the randomization of movement in the x-y plane that represents the unpredictability of E. coli's movement. During the tumble phase, changes in direction are always random. For the run phase, only when the gradient estimation shows an increase in the chemical concentration at the current location will the run phase have intentional, biased movement. Another key component is the presence of a continuous concentration field, which drives the whole simulation and is the bias behind E. coli's chemotaxis.

Implementation | We established various concentration fields representing continuous food source gradients, with the gradient maximum representing the peak food concentration. We used init to initialize input parameters, such as each agent's current position and simulation parameters, such as tumble step size. The simulation proceeded until a specific number of iterations were run.

All agents were initialized with random positions and empty arrays preallocated to store positions at each timestep.

After initializing the previously mentioned metrics, we proceed into our 3-phase loop. For 4 consecutive time steps, each agent takes a small step in both the x and y directions, which is determined using a normal distribution. We also make sure to update the position history arrays every time an agent moves to a new position. In the next phase, we calculate the change in food concentration between the agent's original position and the agent's position after the tumble phase. We compute the gradient vector in two ways. The user may select to use the analytical partial derivatives of the concentration function to calculate the gradient by marking the "analytical_Grad" input argument "True." Conversely, they may choose to estimate the gradient using the change in the concentration function at two given coordinates over the change in the coordinates by marking the same argument "False." Additionally, the user may select which gradient ascent optimizer they choose to implement in their simulation.

Finally, in the run phase, we use the calculated gradient, whether it be from the precise partial derivatives or estimations, to move the agent in the direction of increasing food concentration. The distance the agent travels in this phase is a function of the magnitude and direction of the gradient vector and the selected optimizers.

The user may choose to optimize the gradient ascent using a selection of modular optimization options (which include momentum, Adam momentum, AdaGrad, or RMSProp) through the selection of keyword arguments, all of which have both overlapping and unique impacts on the path the E. Coli will take to the food source. All of the optimization options modify the learning rate applied to the run phase of movement. Either momentum or Adam momentum may be used as the numerator, and either AdaGrad or RMSProp may be used as the denominator in a ratio by which the learning rate is multiplied. The specific observed effects of the different optimization options are discussed further in the evaluation section of this report. The new positions after each run are also included in the position history array. The loop cycling over the run and tumble phases repeats for the specified number of iterations.

The movement of E. Coli in the x-y plane is visualized by overlaying the continuous concentration gradients, using a contour plot, with the coordinates of each step, using a scatter plot. Additionally, different colored markers indicate the starting, intermediate, and stopping points of each path. A second plotting routine visualizes the distance of each E. Coli in a given simulation from the food source in a histogram with plots representing the starting point, end point, and intermediate iterations.

Evaluation | The path of E. coli is shown across many simulations, with varying concentration profiles. In all cases, the E. coli travels from the starting point to the food source, but the specific parameters implemented in each simulation impact the specific path followed. See figures below for examples of a variety of concentration functions, gradient ascent techniques, and optimizers, and descriptions of the nuances between each of their outputs.

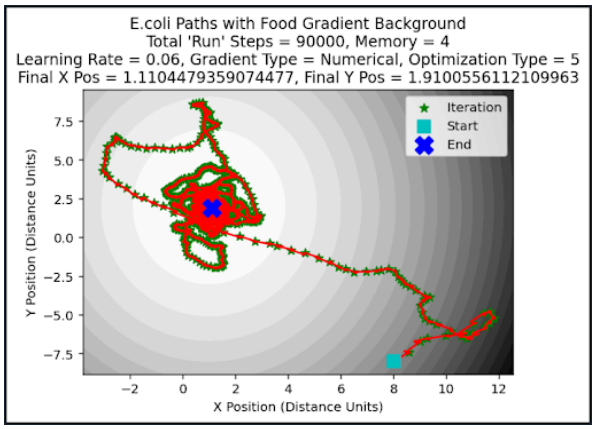


Figure 1. Food source from "Polynomial 2" function using estimated gradient and Adam Momentum and RMSProp optimization

The technique for applying momentum causes the E. coli to move past the food source during the run phases that occur in close proximity to the source. The potential to overshoot the local maximum is a limitation of this optimization technique; however, it has the advantage of avoiding getting stuck at local maxima.

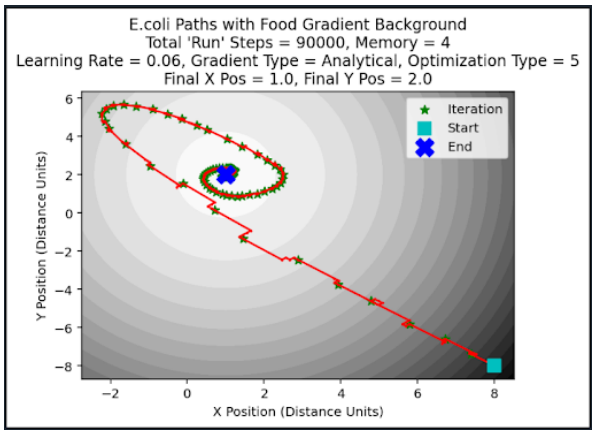


Figure 2. Food source from "Polynomial 2" function using numerical partial derivatives and Adam Momentum and RMSProp optimization

Compared to Figure 1, the E. coli have more precise run phases because they are based on the exact gradients (calculated from partial derivatives).

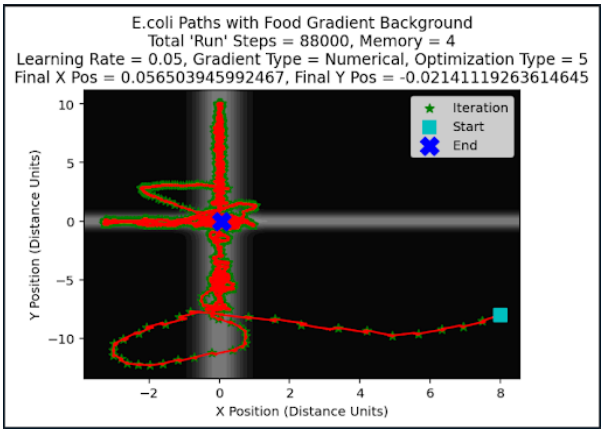


Figure 3. Food source from a normal distribution using the estimated gradient

When in a normally distributed concentration gradient and using the estimated gradient, the E. coli find the food source but follow an indirect path.

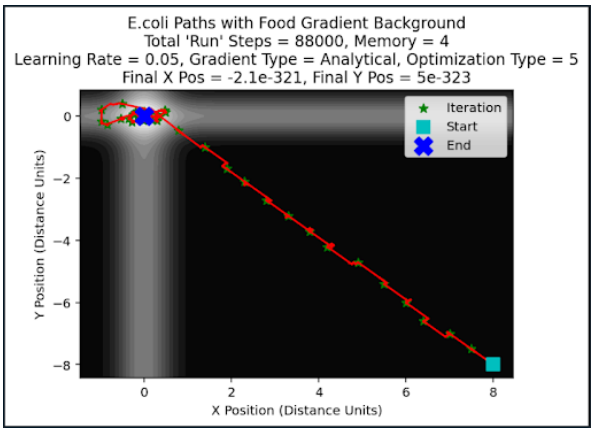


Figure 4. Food source from a normal distribution using numerical partial derivatives

In contrast to the results shown in Figure 3, when searching for the food source using the numerical partial derivatives, E. coli follows a fairly direct path with only minor adjustments close to the maximum.

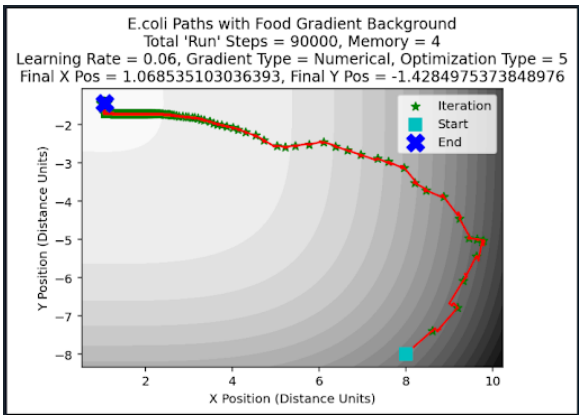


Figure 5. Food source from the "Polynomial 4" function using an estimated gradient

Similar to previous results, the E. coli follow an indirect path to the food source when using the estimated gradient. However, dissimilar to previous results, this simulation was run twice with the exact same parameters and converged at one of the multiple maxima of the function (shown).

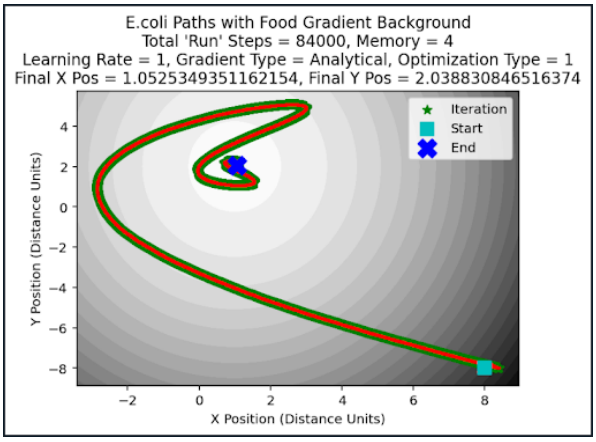


Figure 6. Food source from "Polynomial 2" function using numerical partial derivatives and Momentum optimization

Despite using the numerical partial derivatives in this simulation, deploying the momentum optimizer allowed the E. coli to move past the maximum multiple times before converging at the source.

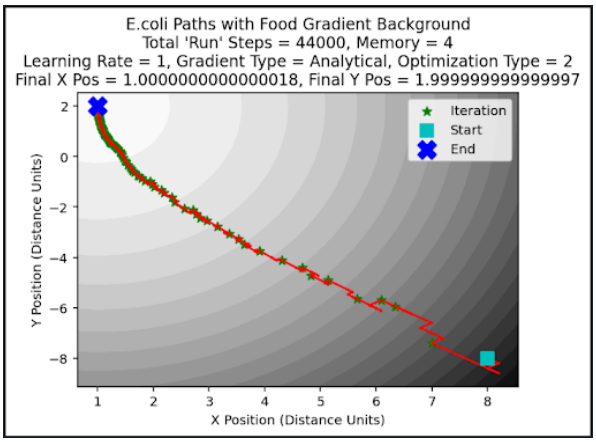
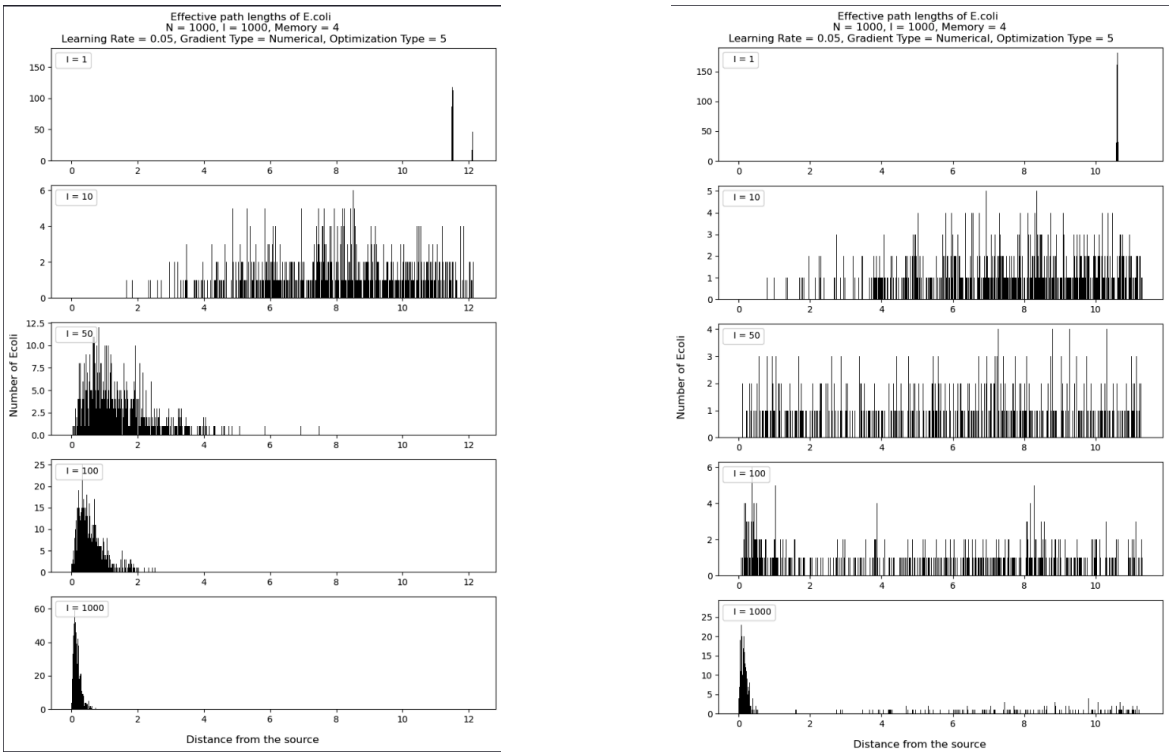


Figure 7. Food source from "Polynomial 2" function using numerical partial derivatives and AdaGrad optimization

Using adaptive gradient optimization with all other parameters kept constant, compared to the run displayed in Figure 7, causes the E. coli to find the food source faster and with more precision without moving past the maximum, as it did with non-adaptive momentum optimization.

Polynomial 2 **Normal Distribution**



Figures 8 & 9. Histogram of E. coli distances from the food source over increasing iteration counts(I = 1, I = 10, I = 50, I = 100, I = 1000) for N = 1000, learningRate = 0.05, gradientType = Numerical, OptimizationType = 5, using a Polynomial 2 food concentration profile (left) and a normal distribution (right).

As the number of iterations increases, the majority of E. coli agents converge toward distances near 0, showing proximity to food sources. Earlier iterations show a wider spread, with many far from the target. By I = 1000, most have clustered at minimal distances. This confirms convergence under these parameters. Inaccuracies in the gradient estimation for normal distribution increases the time it takes the agents to converge.

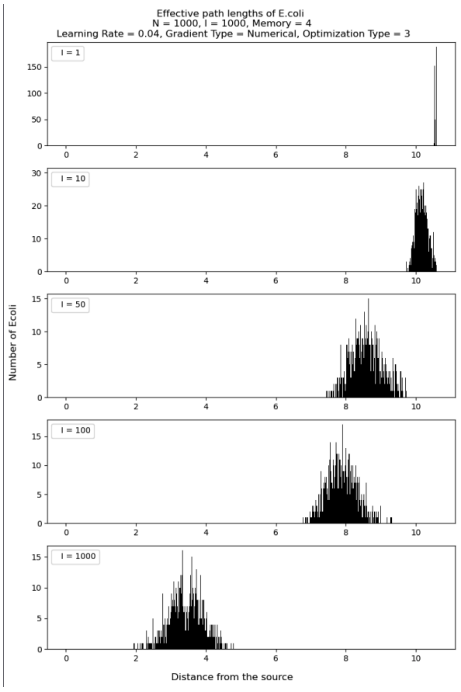


Figure 10. Histogram of E. coli distances from the food source over increasing iteration counts(I = 1, I = 10, I = 50, I = 100, I = 1000) for N = 1000, learningRate = 0.04, gradientType = Numerical, OptimizationType = 3, using Polynomial 4 food concentration profile.
Poorly optimized simulation parameters for this distribution resulted in the E. coli not converging to the food source by the end of the gradient duration.

Discussion | Our results show that the choice of gradient estimation method, concentration profile, and optimization technique affects the general path of the E. coli toward the food source. In all scenarios, they exhibit movement toward the concentration maximum; however, the directness and stability of the path vary depending on the parameters used.

This system is inherently very complex, and the choice of the concentration profile, gradient ascent technique, optimizers, and other parameters all interplay in a very sensitive balance. Small changes in any of these components can significantly alter the E. coli behavior and convergence patterns. Therefore, the user must ensure careful parameter selection to identify the communication that would yield the best biologically plausible results. In addition to the visual trajectory, the simulation function outputs the final location of each E. coli, which can be compared to the known position of the concentration maximum to objectively determine if they reached the target.

Effect of Optimization Techniques. Momentum accelerates movement across the concentration gradient because it introduces velocity to the movement vector, allowing E. coli to maintain direction. This helps avoid getting trapped at local maxima; however, it occasionally causes overshooting near the global peak. Adam momentum allows for a broader exploration of the space because it combines momentum with adaptive learning rates, giving the E. coli the opportunity to find multiple maxima if they exist. AdaGrad enables faster convergence in directions that are consistent with the gradients since it adjusts the learning rate independently in each dimension. This is very advantageous in anisotropic concentration fields; however, the continually decreasing learning rate can eventually lead to slow movement in later stages, particularly ones that are closer to the maximum. RMSProp counteracts this decreasing learning rate issue by weighting the recent

gradients more heavily, which helps maintain responsiveness throughout the simulation. This would produce smoother convergence without the very abrupt slowing we would see in AdaGrad.

The shape of the food concentration field also affects E. coli behavior. For more smooth and symmetric profiles like the normal distribution, E. coli converges more predictably towards the center with minimal changes along the way. In contrast, higher-order polynomial profiles can create multiple local maxima, which would then require using an optimization technique that allows for stronger exploration, like the Adam momentum, to avoid a premature convergence on peaks that are not the most optimal.

Limitations. One main limitation of the current implementation is the absence of constraints that would be present in real biological systems. Additionally, in the numerical gradient method, once the calculations fell outside of an accuracy level of $1e^{-3}$ to $1e^{-4}$, the values became excessively large or small, causing the simulation to blow up. As a result, the numerical approach is less reliable than the analytical solution. Future improvements could include adding concentration profiles that vary over time, introducing noise into the gradient sensing to better mimic biological systems, and adding constraints for more realistic conditions.

References |

Jiang, L. (2020, June 7). *A visual explanation of gradient descent methods (Momentum, AdaGrad, RMSProp, Adam)*. Towards Data Science. https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c

Marken, R. S., & Powers, W. T. (1989). Random-walk chemotaxis: trial and error as a control process. *Behavioral neuroscience*, *103*(6), 1348–1355. https://doi.org/10.1037//0735-7044.103.6.1348

Wikimedia Foundation. (2025, August 11). *Normal distribution*. Wikipedia. https://en.wikipedia.org/wiki/Normal_distribution

Scherfgen, D. (2019). *Derivative Calculator*. Derivative Calculator. https://www.derivative-calculator.net/