

# Replication Report Draft

## Diversity in Digital Host-Parasite Systems

Selma and Isha

### Methodology of the Original Paper

The original paper, “*Rapid Host-Parasite Coevolution Drives the Production and Maintenance of Diversity in Digital Organisms*” by Zamal et al. (2011), investigates how host-parasite coevolution affects phenotypic diversity in digital organisms. The authors use Avida to simulate populations of self-replicating computer programs or otherwise known as “hosts”. These organisms could replicate, mutate, and evolve in a virtual environment. These hosts perform Boolean logic tasks to gain energy and reproduce. Parasites are digital organisms that cannot self-replicate independently, so they must infect hosts and attempt to exploit hosts, by co-opting their task-solving behavior.

The paper explores four different experimental conditions: (1) hosts only without mutation, (2) hosts only with mutation, (3) hosts and parasites without mutation, and (4) hosts and parasites with mutation. The experiments were run in a well-mixed environment for 200,000 updates with some runs stopping mutations halfway through to simulate ecological equilibrium. In parasite treatments, parasites were introduced after 3,000 updates. Mutation mechanisms included point mutations, insertions, and deletions, with hosts experiencing an average of one mutation every four offspring, and parasites every two offspring. Each run featured a population of organisms whose phenotypes were tracked in terms of which logic tasks they performed, and Shannon diversity index was used to quantify phenotypic diversity over time. The central hypothesis is that the coevolution between hosts and parasites drives and maintains higher phenotypic diversity.

### Summary of Results

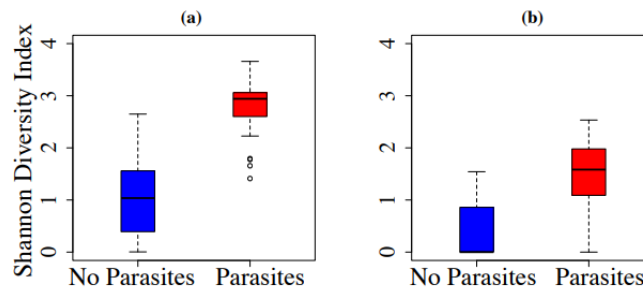
For our initial replication, we focused on the baseline condition from the original paper: host-only evolution with mutations and no parasites. In the baseline condition of the original study, where digital organisms evolved without the presence of parasites, the researchers ran 50 replicate Avida simulations for 200,000 updates. The main result was that these parasite-free populations evolved significantly lower levels of host phenotypic diversity by the end of the run,

as measured by the Shannon diversity index. That indicated that coevolutionary dynamics played a key role in maintaining diversity.

Without parasites or mutation, communities converged toward low diversity, with a mean of about 1.0 Shannon diversity index. However, introducing parasites significantly increased host diversity. After 200,000 updates, runs with parasites exhibited a 1.784-point higher Shannon diversity index than those without (95% CI: [1.506, 2.063],  $p < 0.001$ ). Even when mutations were stopped at 100,000 updates, parasite presence still led to significantly higher diversity (mean increase: 1.15; 95% CI: [0.933, 1.434],  $p < 0.001$ ).

To isolate the effect of novel genetic variation, the researchers replayed parasite inclusive simulations with and without continued mutation after 100,000 updates. Runs with continued mutation showed a significant additional increase in host diversity (mean increase 0.652; 95% CI: [0.321, 0.973],  $p = 0.00012$ ), demonstrating that novel variation further enhances the diversity maintained by coevolutionary dynamics. These results are summarized in Fig. 4 below.

These conditions serve as a control for testing the added impact of parasites and novel variation in later treatments. Any increases in host diversity beyond what is seen here can be attributed to the ecological pressure of parasitism or ongoing mutation.



**Figure 4: Host diversity in runs that evolved without parasites compared against host diversity in runs that coevolved with parasites. Subfigure (a) depicts host diversity when all 200,000 updates had mutations, and subfigure (b) depicts host diversity when mutations were stopped at 100,000 updates. Thus, subfigure (b) shows the *ecological* effects parasites have on host diversity.**

## Absolute Baseline

## Modification and Reimplementation

Our replication is built using SignalGP-Lite and the Empirical library. We implemented a digital evolution system in which grid-based populations of self-replicating digital organisms on a toroidal grid evolve to solve computational logic tasks. The world tracks task completions, reproduction, and mutations. Each organism has a genome executed by a virtual CPU and maintains internal state information for age, accumulated points, and task completions.

Each organism possesses a CPU and OrgState, with logic for task completion and phenotype recording. Tasks include all nine from AVIDA : NOT, NAND, AND, ORN, OR, ANDN, NOR, XOR, and EQU. Our simulation tracks how many organisms complete each task and how this changes over time.

The major elements of our implementations are:

- Organism architecture: Organisms are composed of a CPU object that processes input, produces output, and executes mutation logic. The OrgState stores task flags and statistics.
- World Management: The OrgWorld class governs organism updates, reproduction, task assessment, and data tracking.
- Task logic: Tasks are represented as polymorphic classes that check for correct output against recent inputs, and each successful task grants the organism reward points.
- Visualization: The web version of our system shows the grid that hosts our population, with color-coded organisms that correspond to the task the organism is solving. We further have a settings panel where the user can configure the system, and live readouts of the system statistics.
- Configuration System: When using the native version of our system, the user can adjust the settings in the MySettings.cfg config file, or directly in the command line without having to recompile. Similarly, the user can configure the system with the settings panel when using the web version.

The major difference between the original and our reimplementations:

- Logic engine: We used a simplified logic engine, SignalGP-Lite instead of Avida
- Point system: Rather than using spatial resources and awarded CPU-cycles, we implemented a simple point system where the organism gets a set amount of points for each task solved that it uses to produce offspring
- Metabolism: The original experiment implemented a metabolism in the organisms, where they have to spend resources to perform tasks. We removed this feature, leaving the production of offspring as the only way to spend points
- Mutation rate: We use a set mutation rate of 0.02 for our hosts, and will implement a doubled mutation rate in our organisms. The paper rather implemented mutation rates

corresponding to one mutation in every four offspring for hosts, and twice that in parasites

- Initialization: Our organisms are initially only capable of reproducing, while in the paper they were also initialized with the ability to perform the NOT task.
- Experiment duration: We ran our experiment for 100,000 updates, rather than 200,000.

## Description of Replication Results

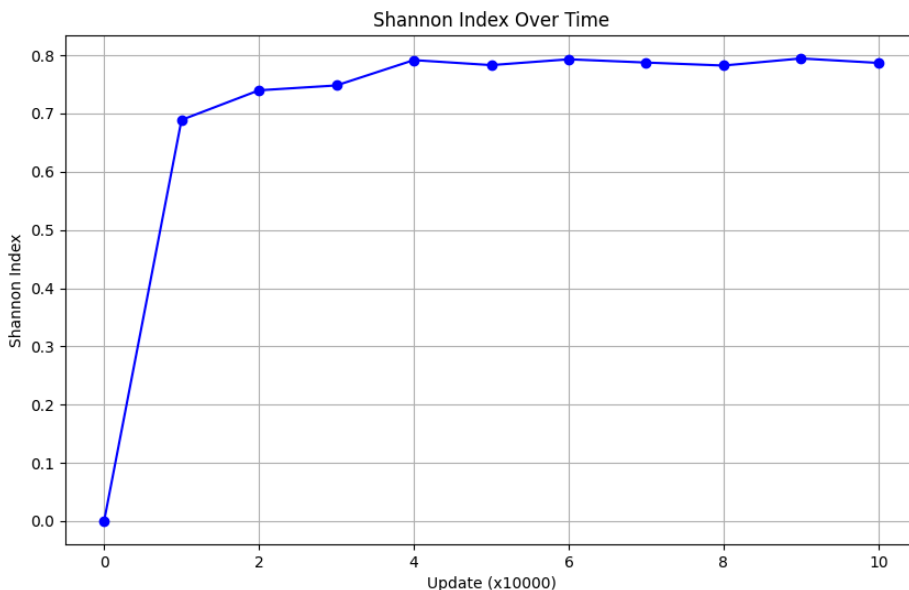
For our baseline test, we ran our system with the current configurations:

- Initial population: 10
- Mutation rate: 0.02
- Number of updates: 100,000
- Task reward: 10
- Cost of reproduction: 20
- World size: 80x80
- Max number of updates between reproductions: 1000

We chose to plot the Shannon index over time, along with a histogram showing the number of organisms solving each of the tasks in the final state of the system. Both figures are shown below.

We see that the Shannon index is initially rapidly increasing, before it flattens out at about 40,000 updates, and stays stable for the rest of the run. The Shannon index of the final state of the system is 0.79. Further, the histogram tells us that most of the organisms evolve to solve one of three tasks: AND, ANDN, and EQU, where EQU is the most prominent.

All organisms will randomly evolve to learn tasks, and it seems like these three are the easiest to learn. Once organisms learn how to solve tasks, there is no push towards learning new tasks, as there is no added reward in doing this. Thus, the produced offspring are likely to always solve the same task as its parent, meaning that the population will stagnate.

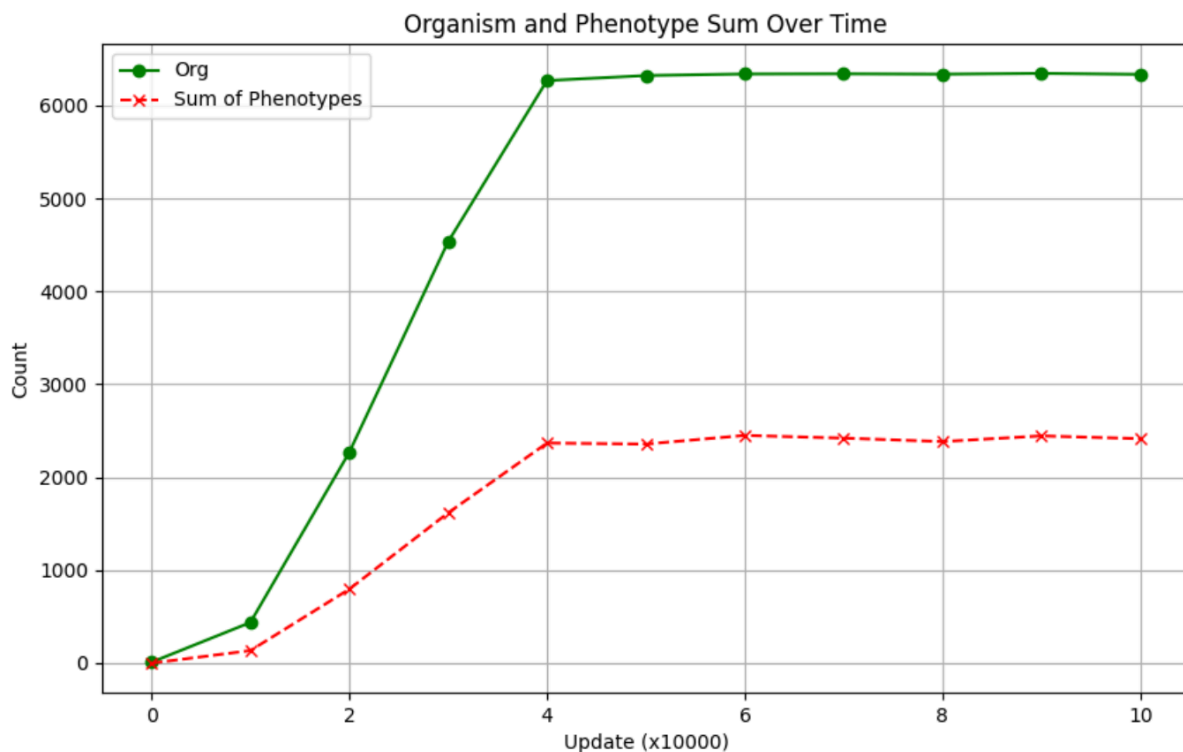


## Analysis of Differences

Our Shannon index ended up at 0.79, which is lower than the  $\sim 1.0$  mean that the paper found. Still, over the 100 runs in the original experiment, they found a range of Shannon indices between about 0.4 and 1.7 (read off from Fig. 4, no exact values are provided), and we are well within this range. We have thus successfully set up a system that mimics the system in the original experiment, even though we have made some simplifications.

We ended up with a population of 6335 organisms, which fills 99% of our grid. This is significantly higher than the 70% saturation they had in the initial experiment. This means that we might want to adjust the maximum time between reproductions so that the population stabilizes at a lower saturation. If this is difficult to balance, we know that our current results are still representative.

Out of curiosity, we also plotted the total number of organisms, and the total number of task-solvers over time.



We see that both flattens out at about 40,000 updates, meaning that we might get away with restricting our experiment to 50,000 or so updates in the future to save computational power. If the computational power becomes an issue, we may also consider limiting the grid size in the future.

We conclude that our baseline test was a success, where we were able to replicate a population diversity similar to the one found in the paper without any parasites.

## Basic Controlled Experiment:

### The Experiment:

Building on our baseline replication, we conducted a modified version of the host-parasite coevolution experiment described in Zamon et al. (2011). In our control group we let the Hosts evolve without the Parasites being injected. In this experiment Hosts and Parasites coevolve with mutation, but Hosts were introduced at time step 0, but Parasites were injected at update 1500. Both treatments ran for 12,000 updates, with all parameters held constant across the two groups. For an expansion for our paper, we also ran the experiment with different virulence values to see how the Hosts are able to evolve depending on the amount of resources the parasites are able to steal.

We systematically varied parasite virulence across a series of runs, with all simulations using mutation for both hosts and parasites. The primary aim was to analyze how different virulence levels affect host phenotypic diversity. For each virulence setting (including a no-parasite baseline), we ran three replicates using different random seeds (3, 4, and 6), and measured host Shannon diversity at the end of each simulation.

We use the following settings for the control and our base experiment without the expansion:

- **SEED**: varied across replicate runs for each experiment/virulence level (3, 4, 6)
- **NUM\_START**: 10 hosts
- **NUM\_PARASITES**: 100 (injected at update 1500)
- **MUTATION\_RATE**: 0.02 (hosts)
- **PARASITE\_MUTATION\_RATE**: 0.04
- **VIRULENCE**: 0.8 ( also varied for expansion experiment)
- **REWARD**: 20
- **LIFE\_SPAN**: 15 updates

- **NUM\_BOXES:** 50 (grid size 50x50)
- **BONUS\_UPDATE\_LIMIT:** 500

## **Modification and Reimplementation**

Our system differed from the original in a few ways which likely contributed to the diversity shifts we observed. While our system implemented virulence as a percentage of task reward stolen, the paper could have applied virulence to the host's entire point pool. If so, this would apply stronger parasitic effects in the original, further explaining why our parasites exerted less selective pressure.

Another key difference was in grid saturation: the original experiment maintained roughly 80% world saturation, while our system consistently approached 100%. This was a necessary trade-off to ensure enough host task-solvers were available to support parasite reproduction.

More broadly, our system incorporated several intentional changes to improve parasite survival:

- Parasite injection in our system occurs into random task-solving hosts, rather than being task-matched at birth due to the asynchronous nature of task-solving.
- A 500-update bonus phase follows parasite injection. During this time, parasites receive survival points, are rewarded for solving any task (not just the matching host task), and to not harm their host. This helps them establish a foothold and be able to survive and reproduce.
- Parasites that solve a task are added to the reproduction queue and receive points for each update they remain active.
- Reproduction costs were lowered for both hosts and parasites to encourage rapid population growth.
- Parasites can produce multiple offspring in a single update.
- All organisms being with the ability to solve ANDN task instead of NOT, which proved easier to solve.
- Hosts are granted survival points early on to help bootstrap reproduction
- Offspring placement avoids overwriting task-solvers when possible, preserving task-solving diversity and parasite habitat.
- Parasites are removed if they do not solve a task within a set number of updates, reducing the chance for task-matching

These adaptations were necessary because the host-parasite ecosystem in our system proved more fragile than in the original study. In early tests, parasite populations often declined quickly along with the task solving hosts and they weren't able to stabilize. To

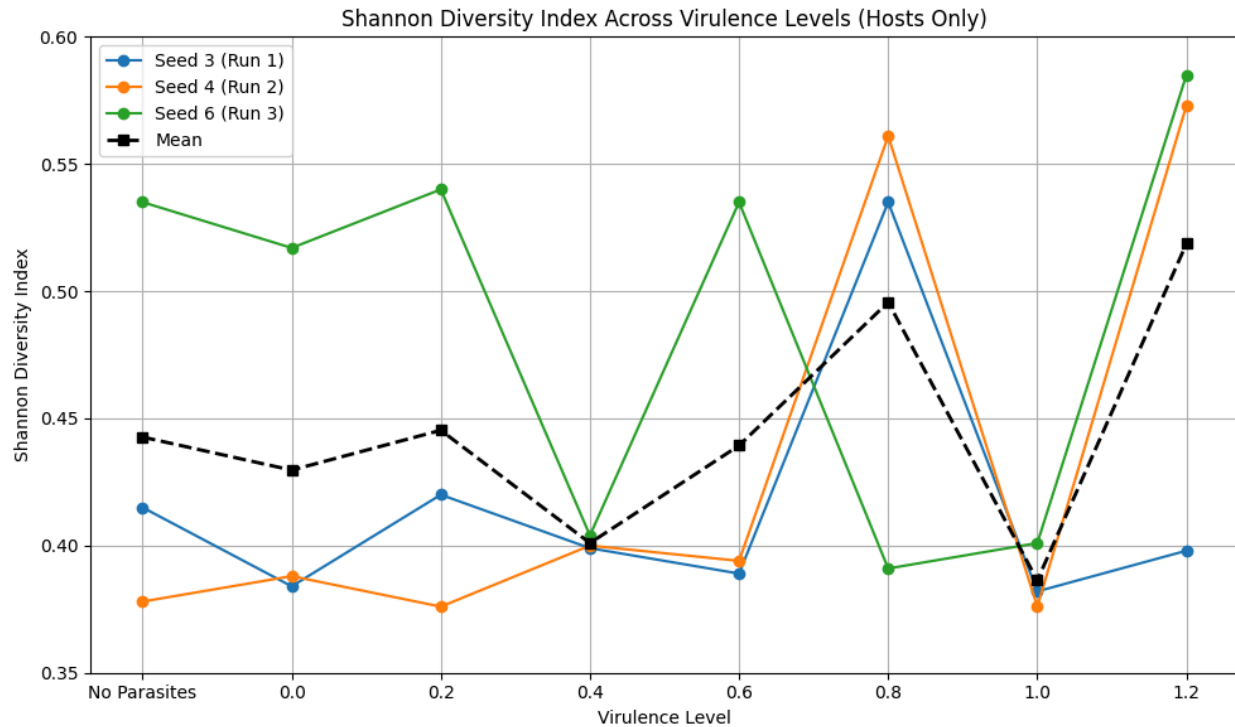
support parasite persistence, there were priority system configurations that maintained a large number of active hosts solving tasks, at the cost of some selective pressure and coevolutionary intensity.

## Results

We present the final Shannon diversity values across multiple virulence levels, averaged over three simulation runs (with seed values 3, 4, and 6). Mutation was used in all runs. The full results are shown in the table below, followed by a graph.

<b>Virulence</b>	<b>Seed 3 (Run 1)</b>	<b>Seed 4 (Run 2)</b>	<b>Seed 6 (Run 3)</b>	<b>Mean Shannon Index</b>
<b>No Parasites</b>	<b>0.415</b>	<b>0.378</b>	<b>0.535</b>	<b>0.443</b>
<b>0.0</b>	<b>0.384</b>	<b>0.388</b>	<b>0.517</b>	<b>0.430</b>
<b>0.2</b>	<b>0.420</b>	<b>0.376</b>	<b>0.540</b>	<b>0.445</b>
<b>0.4</b>	<b>0.399</b>	<b>0.400</b>	<b>0.404</b>	<b>0.401</b>
<b>0.6</b>	<b>0.389</b>	<b>0.394</b>	<b>0.535</b>	<b>0.439</b>
<b>0.8</b>	<b>0.535</b>	<b>0.561</b>	<b>0.391</b>	<b>0.496</b>
<b>1.0</b>	<b>0.382</b>	<b>0.376</b>	<b>0.401</b>	<b>0.386</b>
<b>1.2</b>	<b>0.398</b>	<b>0.573</b>	<b>0.585</b>	<b>0.519</b>

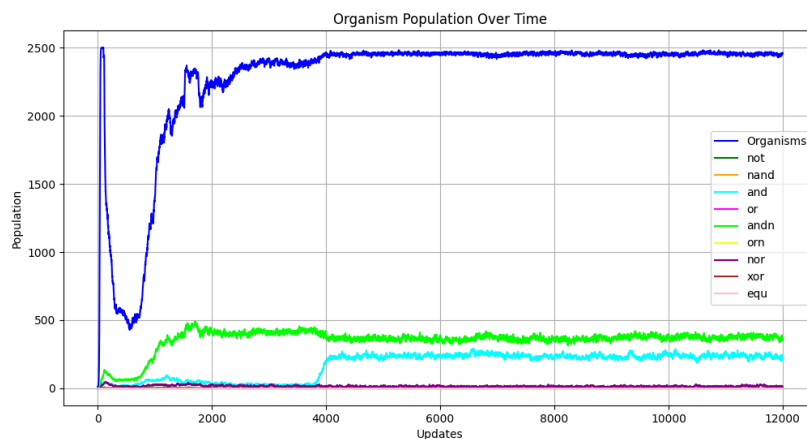




## Analysis of Virulence Effects and Results

The graph shows that the diversity doesn't consistently increase or decrease with virulence. Diversity hovers around the baseline (0.44) for most virulence values but exhibits peaks at 0.8 and 1.2 and a dip at 1.0.

- *Baseline Comparison:* The “no parasites” condition yielded moderate diversity (mean: 0.443), dominated by ANDN solvers.

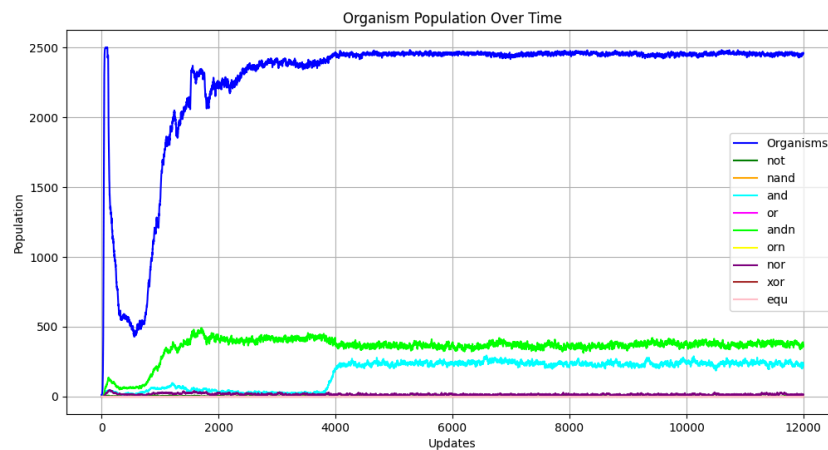


- *Low-to-Mid Virulence (0.0-0.6):* Diversity fluctuated slightly by staying close to baseline. There is a possibility that these levels were too mild to exert strong selective pressure, or parasites may not persist long enough to influence host evolution meaningfully.

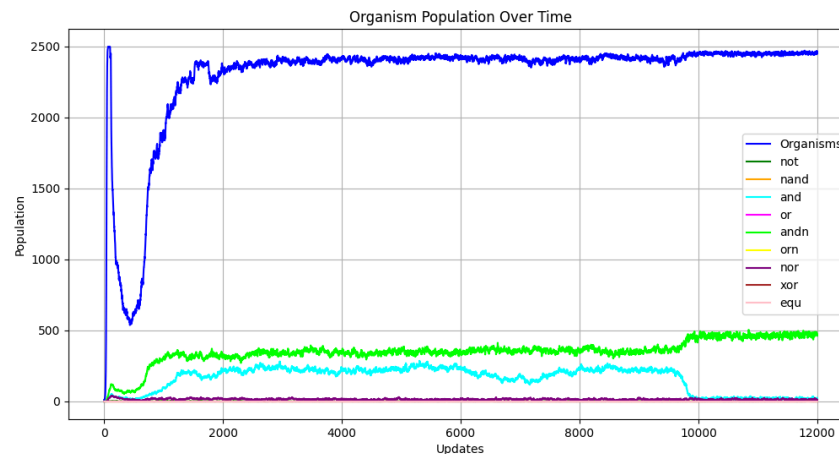
Compared to the no-parasite condition, these levels have small effects on host diversity, suggesting that mild parasitism exerts limited selective pressure. That could be due to poor parasite survival or low virulence impact.

- *Virulence 0.8*: This condition had a notable higher mean (0.496), suggesting that it is the best virulence where parasites persist and apply evolutionary pressure. However the Shannon entropy was higher in the first two runs, but saw a drop in entropy due to a collapse in AND solvers. Two out of the three runs performed strongly, indicating that parasites likely persisted and created enough ecological pressure to favor task diversification.

For the first run:



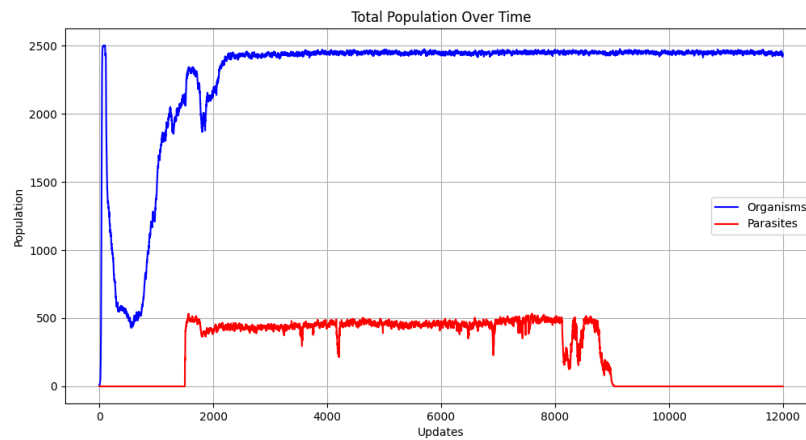
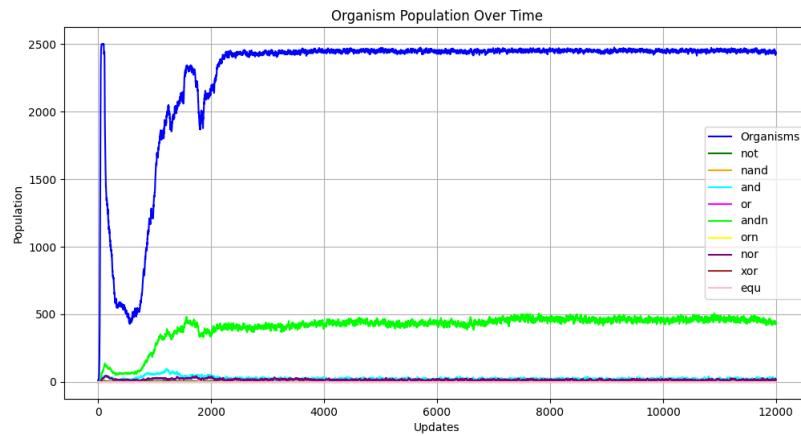
For the 3rd run:



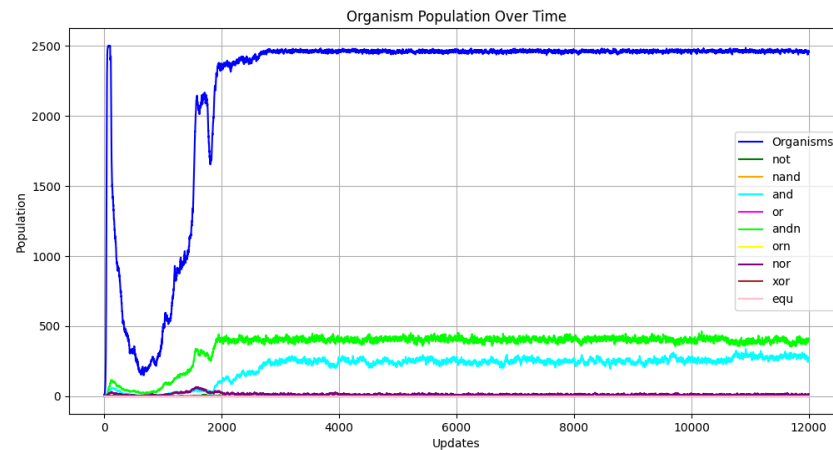
- *Virulence 1.0*: Host diversity decreased slightly (mean:0.386) potentially due to excessive virulence suppressing parasite evolution.

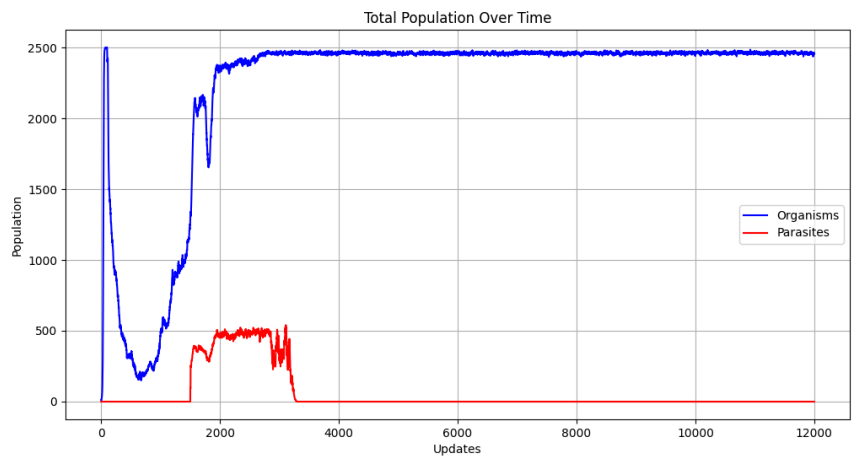
- *Virulence 1.2*: This condition was the most interesting, with the highest average diversity (0.519). Runs 2 and 3 showed sharp rises in AND solvers; parasites survived in run 3 but went extinct in run 2.

### Run 1

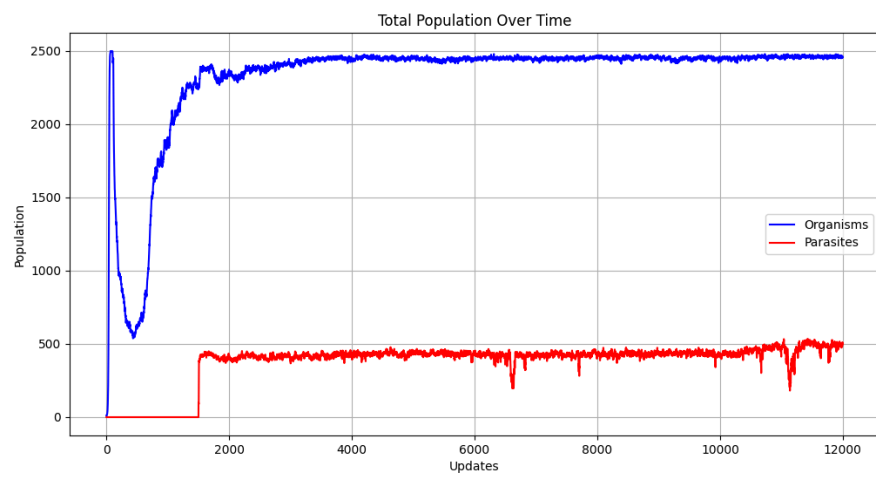
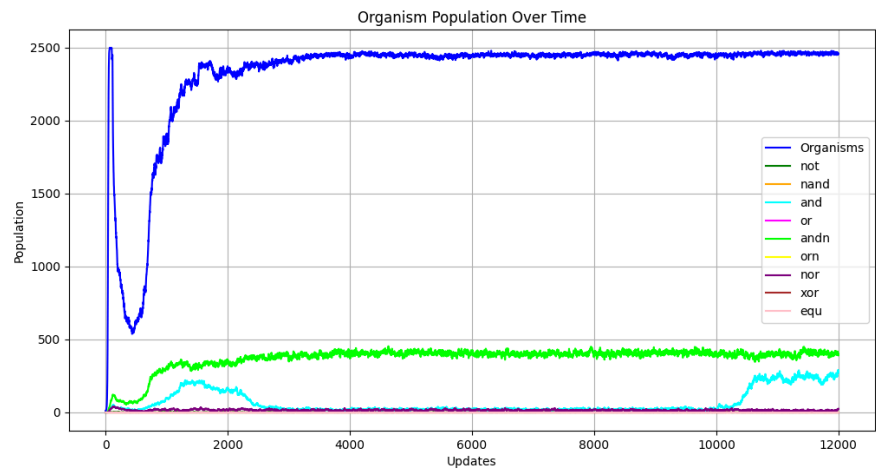


### Run 2





Run 3



There was also a lot of intra-group variability, which could suggest that other factors like early mutation and parasite survival timing play a major role in shaping diversity even under identical parameters.

## **Conclusion**

Through our replication and extension of the host-parasite coevolution experiment from Zaman et al. (2011), we demonstrated that even with differences in implementation, such as logic engine choice, mutation modeling, reproduction mechanics, and grid saturation, our system successfully captured the central dynamic: the host-parasite interactions can promote and maintain phenotypic diversity. Our baseline replication produced diversity levels within the expected range of the original study, and our controlled experiments exploring parasite virulence revealed a nuanced, non-linear relationship between virulence and host diversity. Notably, mid-to-high virulence levels (0.8 and 1.2) yielded the greatest increases in Shannon diversity, while extreme levels led to population instability or parasite extinction. These findings reinforce the importance of coevolutionary pressure in sustaining diversity, but also highlight the fragility of digital parasite-host systems, which depend on finely tuned parameters for survival and impact. Our modifications, including survival bonuses, relaxed reproduction costs, and altered injection logic, were crucial in supporting parasite persistence, albeit at the cost of reducing selective pressure. Going forward, further replicates, improved injection criteria, and alternative virulence definitions may help refine our understanding of coevolutionary dynamics in digital ecosystems.