Kendra Winhall
CS 361

Final Replication Report

Summary of the methodology of the original paper

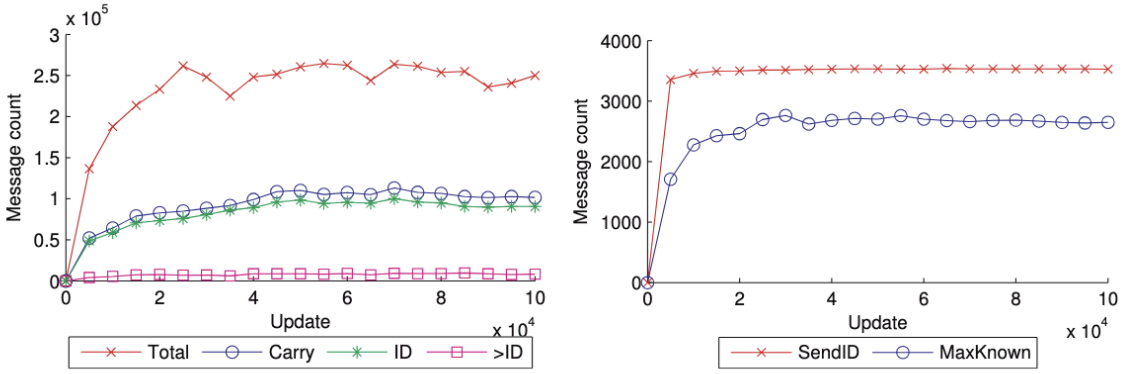
AVIDA is a digital evolution software platform. It enables users to create a population of self-replicating computer programs, control certain parameters and tasks, and study the evolution of the organisms over time. Organisms are subject to random mutations, rewards and punishments, allowing for interesting behavior to evolve. Organisms can also communicate with each other by sending and receiving messages. The original paper uses AVIDA for its experiment, with parameters matching the default AVIDA configuration.

The original paper studies the evolution of cooperative behavior and distributed problem solving in digital organisms. Specifically, the original paper studies the evolution of message-sending behaviors and attempts to reach a benchmark of 95% of messages carrying the largest cell-ID.

Summary of the results from the original paper that you attempted to replicate

Since I did not do this experiment in AVIDA, I did not have the built in tools that AVIDA has. The absolute baseline was to create a message sending and retrieving instruction, and verifying that it works.

The easiest, basic experiment in the paper is about filtering messages. The researchers rewarded organisms for sending messages containing cell-IDs greater than the organism's own cell-ID to see if all organisms in the population would evolve to send the largest cell-ID. While the organisms did not evolve to do that, they did send messages containing cell IDs that were larger than their own. The graphs below show their results.



(a) Messaging Behavior (b) Task Performance

Fig. 2. Data filtering with MAX-KNOWN and SEND-ID tasks; average of 20 trials.

Description of your reimplementation and any changes you needed to make from the original approach

Since I did not do my experiment in AVIDA, I had to write instructions for sending and receiving messages. I looked at AVIDA Github to try to make these instructions similar to AVIDA actual implementation, but the way my organisms handle sending and receiving messages is still a bit different than the way AVIDA does it. For example, in the original experiment, organisms can send messages to other organisms they face, but in this case I just choose a random neighboring cell.

I added attributes to each organism such as a unique cell-ID, an inbox of all received messages, an index that keeps track of the next message that can be retrieved (but has not been read yet), and a deque that holds the 100 most recently sent messages for that organism, which is likely different than how AVIDA handles these attributes under the hood.

My parameter values for processes like mutation or reproduction are likely different from AVIDA's default parameters, adding another difference between my work and the original paper.

Description of replication results

I was able to successfully replicate the absolute baseline result. I was able to write instructions for sending and receiving messages and verify that it worked as expected. Even without tasks rewarding organisms for sending larger message values, it seems that sometimes they evolve to do so anyway.

I spent days working on implementing the basic experiment, but unfortunately my code kept getting different errors. After unsuccessfully trying to solve the below error for an embarrassing amount of time and getting nowhere, I reverted my changes back to the absolute baseline.

```
Unset
Process 51102 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = EXC_BREAKPOINT
(code=1, subcode=0x194fa4248)
    frame #0: 0x0000000194fa4248 libsystem_malloc.dylib`free_small.cold.1 + 28
libsystem_malloc.dylib`free_small.cold.1:
->  0x194fa4248 <+28>: brk    #0x1

libsystem_malloc.dylib`small_check_region.cold.1:
    0x194fa424c <+0>:  brk    #0x1

libsystem_malloc.dylib`small_madvise_free_range_no_lock.cold.1:
    0x194fa4250 <+0>:  brk    #0x1

libsystem_malloc.dylib`xzm_metapool_alloc.cold.1:
    0x194fa4254 <+0>:  stp    x20, x21, [sp, #-0x10]!
Target 0: (native_project) stopped.
```

Analysis of replication results

My analysis for why I was unable to replicate the basic controlled experiment is that I did not dedicate enough time to the project due to personal matters and mild discouragement. It took a lot of time to try to write and debug my own versions of the functionality that AVIDA already has, so most of my time was spent writing code for the digital evolution framework rather than implementing the tasks that the original paper focuses on. As a result, even though I poured a lot of time into this project, my code didn't compile most of the time, so it felt like I wasn't making progress.

In the end, I chose to spend less time on trying to make this experiment work and more time revising past assignments for this class because the other assignments were going more smoothly. However, if I had had more free time to work on this project, I would have asked for help with interpreting and solving the bugs that I was getting. Regardless, I imagine my implementation varied enough from AVIDA's implementation that I would have had some major differences in my results.

Expansion

I ran my code with a variety of seeds and number of starting organisms, which is a simple extension but an extension nonetheless. Changing these parameters on the GUI does visibly change the max sent values for cells in the population, but I was not able to detect a strong pattern or reason as to why. I also saved some data files with different parameter configurations, but I did not have the time to analyze them. (Some example files are included in the repository if you'd like to take a look). In hindsight, the average value is not the best one to choose because the average is dragged down by newly born organisms. In general, I see a lot of extreme values in both the data files and the web GUI, but not values in the middle, which is odd–I thought they'd be more evenly distributed. If I had more time, it would be interesting to run various summary statistics aside from the average.