Yeabsira Gebreegziabher, Elhadji Amadou Touré

Professor Anya Vostinar

CS 361: Artificial Life and Digital Evolution

November 21st, 2022

Replication Report

Summary of the methodology of the original paper

Experimental Setup

AVIDA was used to configure a single population of 100 demes each comprising 25 digital organisms. Each deme is configured in a 5 × 5 torus. Experiments were run for 100,000 updates with an average of 30 CPU instructions. To account for the stochastic nature of evolution, 20 separate runs of AVIDA were performed for each experiment. Only one organism was put in each deme at the start of each program. Instructions in this paper were executed by the organism's virtual CPU. The architecture used in this study contained a circular list of three general-purpose registers {AX, BX, CX}, two general-purpose stacks {GS,LS}, and four special-purpose *heads*. Heads may be thought of as pointers into the organism's genome and are similar to a traditional program counter or stack pointer. The instruction-head points to the next instruction to be executed. The flow-control head points to a location in the genome to which the instruction-head may be moved upon execution of certain instructions. The read-head and write-head were used during replication, and enabled the organism to read and write instructions within its genome.

Environment and Communication

In the paper, the environment comprised a number of *cells*, each of which can contain at most one organism; organisms cannot live outside of cells. Each cell has a circular list of directed *connections* to neighboring cells; these connections define the topology of the environment. The topology is configurable by the user. Each cell in the environment has a *facing*, a single connection selected from its connection list that defines the orientation of the resident organism. The facing of a cell may be used by the organism in a number of different ways. For example, an organism can send a message to the neighbor it is facing. The organism can also read and manipulate the facing of its cell via a “get-facing” and variants of a “rotate” method, respectively. Each cell has two associated identifiers, one that is allocated sequentially over all cells in the population, termed the *cell-sequential ID*, and another identifier that is a random 32-bit integer, termed the *cell-random ID*. A resident organism may obtain its cell identifier via a “get-id” instruction, which was configured to refer to either the cell-sequential ID or the cell-random ID. These IDs were used to identify the organism to its neighbors. When an organism replicates, a target cell that will house the new organism is selected from the environment. Different models to select this target cell are available, including “MASS-ACTION” which selects a random from among all selects, or “NEIGHBORHOOD”, which selects from cells adjacent to the parent, among others.. In every case, an organism that is already present in the target cell is *replaced* (killed and overwritten) by the offspring. Organisms could also communicate with their neighbors using “send-msg” and “retrieve-msg” instructions. Each message contains a *data* and a *label* field, both of which are 4-byte values. A third instruction, if-inbox-empty, returns a boolean value representing whether or not there are messages in the organism’s inbox. To send or receive a message in this experiment, first, the sending organism must execute a “send-msg” instruction, which marshals two registers into a message and sends the message in the direction currently faced. If the sending organism is facing a neighboring organism, the message is deposited in that neighbor’s inbox. If the sender was facing an empty cell, the message is lost. Finally, the recipient of the message must execute a “retrieve-msg” instruction to extract the message from its inbox and place its two fields into registers.

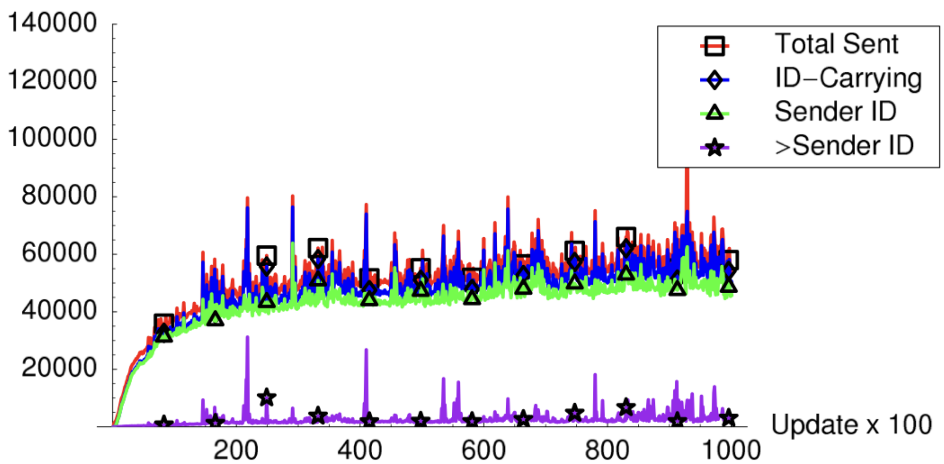
Reproduction

Copy mutation rate was set to 0.75%, while the insertion and deletion mutation rates were set to 5.0%.

Summary of the result from the original paper that we attempted to replicate (all figures are on averages of 20 runs)

1. Message-based leadership, without deme competition

In their first experiment using group selection, demes were not competing against each other; so no group level selection was performed. This experiment was designed to determine if simply splitting the population into demes would have a substantial effect upon behavior. Their results showed that of the 50,000 Total Sent messages, approximately 45,000 were Sender ID messages, thus indicating that the leader election did not take place. This essentially meant that not greater than 95% of all messages sent carried the ID of the leader, which is the organism that has the numerically-largest ID. Their results also indicated that Sender ID messages were not increasing during the run, implying that leader election would not occur even if the experiment were run longer. Plot results are shown below, where “Total Sent” represents the total number of messages sent, “ID-Carrying”, represents the total number of messages that contained valid IDs, “Sender ID” represents the total number of messages sent that were the organism’s own ID, and “>Sender ID” represents the total number of messages that were greater than the sender’s ID.



.

Description of our reimplemnetation and any changes we needed to make from the original approach (description of how our program works and how it differs from the original methodology)

Experimental Setup, Environment and Communication

Our experimental setup is identical to that of the study's, except that we did not use registers, heads, and the likes of to facilitate communication. Instead, we utilized a list that was a part of each organism's class and served as a point of storage of IDs, incoming and outgoing. SGP-Lite was also included, alongside reimplemented tasks and instructions. Furthermore, while the paper utilized the known number of cells to determine what would be the highest ID in the world, we assigned the highest targeted ID to be "99" and framed the program to work around organisms attempting to achieve leader selection via the organism that carried that ID. Next, for the plotted data of the results, we kept the author's number of updates (100,000) and number of CPU instructions (30). These were performed one time instead of 20 (in the interest of time, as they took more than what we could afford to finish setting up data files), and shown through a plot (refer to plot in our results). Also, since we were replicating results that involved no deme competition, we only made one single deme instead of multiple ones (it is safe to assume that they would all perform the same if there is no interaction between them). Additionally, to avoid dealing with manipulating cells in the world and adapting this study to Empirical, additional changes had to be made to account for the cells' main role: rotations and random sequential IDs. The following were implemented:

1. In terms of instructions, the AVIDA logical 9 tasks were not implemented. We do understand and acknowledge that it does bring a difference in the ability of organisms to perform tasks; however, the instructions in our project were utilized to find a random index to choose an ID in the inbox, and did not affect the rewarding of organisms. Instead, SGP-Lite instructions were used, and instructions were modified through changes in our Task and World classes so that IDs would be taken as inputs in our Tasks, and an index that referred to a place in organisms' inboxes was outputted. Organisms were rewarded appropriately based on which message they chose to send (explained further below)

2. Assigning an ID to organisms (instead of cells), and utilizing Empirical's built-in functionality of getting random neighbors instead of giving a facing to cells, as mentioned in the paper being analyzed here.

With that in mind, to reward the organisms, the following specifics were implemented:

1. An organism getting a specific number of rewards set by the user for sending their own ID (between 0.1 and 1)
2. An organism getting a specific number of rewards set by the user, plus 1, for sending the highest ID in their inbox
3. An organism getting a specific number of rewards set by the user, plus 2, for sending the highest ID that exists in the world
4. An organism being penalized by a specific number set by the user for sending an ID that does not exist.
5. No additions or deductions were made for all other types of IDs sent.

Replications

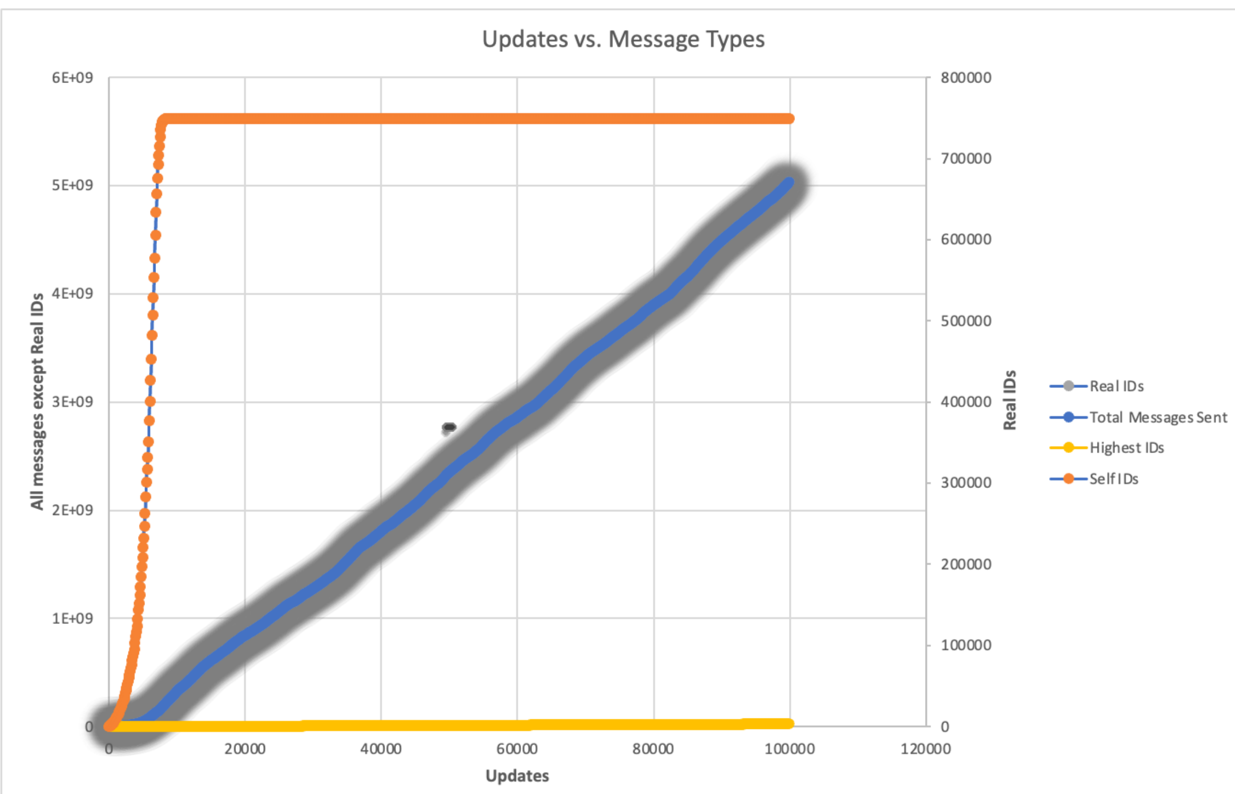
In our program, the copy of a mutation is set at 0.02% (while we also give the user the ability to take it up to 1%). The study we are attempting to replicate has a copy mutation rate set to 0.75%. Our experiment does not have insertions and deletions, only substitutions based on our rate.

Extensions

An extension that we implemented that should affect the evolution of the organisms towards their goal of electing a leader dealt with the elimination of organisms when they sent their own ID numerous times(we arbitrarily set it to 80). We also made sure that the organism with the highest ID was not subject to this extension, as we would not want it to be eliminated. By eliminating those organisms, achieving the goal of electing a leader might have been more probable.

Description of our replication results

Our replication results ended up being very similar to the paper's. No leader selection was made (although numerous organisms were in fact sending the highest ID in the world). With a start of 70 organisms, a reward of "1", our expansion turned off, and the mutation rate set at 0.02%, our results showed that of an average of 5.03E09 Total messages sent, approximately 748900 were Sender ID messages, 2.47E7 were the highest IDs in the world and the rest were non-existent IDs, thus indicating that the leader election did not take place. This essentially meant that not greater than 95% of all messages sent carried the ID of the leader, which is the organism that has the numerically-largest ID. Our results also indicated that Sender ID messages were not increasing during the run, implying that leader election would not occur even if the experiment were run longer. Finally, we also observed that the percentage of the presence of the highest ID in organisms' inboxes barely approached 0.49%, a result that is far off from the 95% that would have been an indicator of leader election. The results were very similar even with our expansion employed, with differences of a couple of thousands of messages which proved themselves to be irrelevant in the bigger scheme of things. Plot results are shown below:



Slight Differences in Results

Differences in numbers of messages in all categories with fairly similar setups

Our experiments outputted a far greater number of messages sent in all categories (with greater differences) while somewhat similar setups were made in terms of updates and number of instructions per process. We hypothesized that this is mainly due to the fact that the instruction sets were different, and when our CPU processed at each update, more than one message per organism was being sent per update.