Openlife

Sustained Complexity of Intelligent Systems

(Computer Science)
April 3rd, 2017
Sandia Preparatory School
Sandia-1

Team Members:

Eli Stone (12th, eli.stonium@gmail.com)
Lucas Saldyt (12th, lucassaldyt@gmail.com)

Teachers:

Neal Holtschulte (nholtschulte@sandiaprep.org)

Mentor/Sponsor:

Neal Holtschulte (nholtschulte@sandiaprep.org)

Languages:

C++, Python, bash

Summary

Openlife is a c++ simulation of social creatures. Creatures in Openlife follow deterministic rules to determine their movement and reproduction. Our experiments examine how changes to simulation parameters affect metrics such as number of networks and entropy. Learning how the parameters affect the system behavior will give us insights into rule-based systems in general and may have specific applications, i.e. swarm robotics. Our goal is to learn how and why complex patterns emerge from systems with simple rules, and also to investigate practical applications of agent-based automata.

Problem Statement

The goal of Openlife is to create a system of simple rules and parameters that can exhibit a wide variety of emergent behaviors by making minor changes to the parameter set. The discussed parameters are:

- Neighbor radius: radius around a creature that defines members of that creature's group (default = 400 pixels).
- Crowded limit: The threshold, in number of creatures, at which a creature attempts to leave a group due to overcrowding. (i.e. if crowded limit is four, and more than four creatures are in a group, then the creatures attempt to leave the group to form smaller groups, default=10).

Hypotheses:

A. As crowded limit increases, average group size will increase while number of groups decreases. Standard deviation in group size will increase as well. Specifically, when crowded limit is set to 10 the average group size will be significantly larger, the standard deviation in group size will be significantly larger, and the number of groups will be significantly smaller with 95% confidence compared to a crowded limit of 4.

B. When crowded limit is 4, (2 + the minimum number of creatures to prevent underpopulation), area will be significantly larger than when other thresholds are tested ($p < 0.05$). When the threshold is 70%+/-5% of the maximum network size, before overpopulation damage occurs, the population will be significantly larger than other thresholds ($p < 0.05$).

Methods

Data is collected by changing a parameter file for the simulation and then running the simulation for a constant number of timesteps. The raw data collected is a .csv file of the locations of each creature. From this data, metrics like entropy and standard deviation in group size are calculated. A Python script manages this process, automatically modifying the parameter file and repeatedly running simulations, saving output data in the form of box plot and sample mean hypothesis tests for further analysis.

Hypotheses of the form “varying parameter x positively/negatively varies metric y” are verified by a sample mean t-test between samples taken from simulations run with each parameter value. If a hypothesis claims a positive relation, then, at each step of x, the null hypothesis is that the value of y is equal to the previous value of y before x was changed. The alternative hypothesis is that y is greater than the previous value of y. This comparison is evaluated with a sample-mean t-test. For example, when varying crowded_limit, the first t-test uses ten samples with crowded_limit = 3, and then ten samples with crowded_limit = 4, and compares the means of each set of samples using the sample mean formula to see if they are significantly different. A similar t-test occurs between 4 and 5, 5 and 6, and so on. If each of these t-tests give values in the same direction (have the same sign) at a confidence level of 95% or higher, then it can be concluded that a given parameter varies a given metric.

Verification and Validation

Openlife, like many pieces of software, has a suite of unit tests. This is mostly used for verifying the low level functionality of the simulation (i.e. angle and distance calculations, limited rotation mechanics, etc). The higher level functionality of the simulation is what is being tested in our hypotheses.

Results

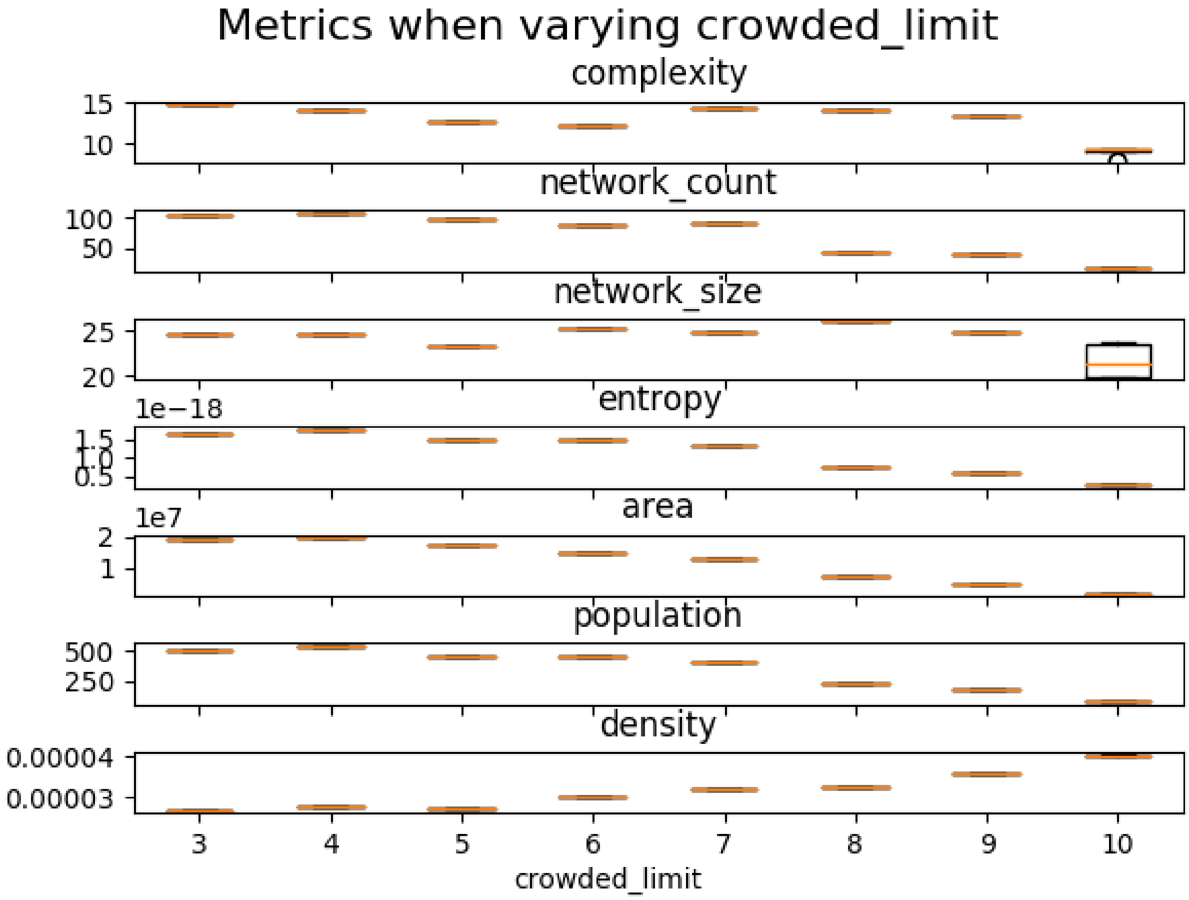
Each hypothesis is given its own page, for nicer formatting of plots.

A. When doing a t-test between each sample, it appears crowded limit has no direct variation with complexity, group size, or group count.

   However, crowded limit inversely varies both area and population after crowded limit is above four.

   If the last point (crowded limit = overpopulation damage threshold) is discarded (justified by the fact that creatures will take damage before leaving a group), then crowded limit is seen to directly vary group size (but only between the minimum and maximum threshold values).

   If only the first and last points are compared in a sample mean t-test, then crowded limit significantly decreases network count and complexity. The results from the middle points may be from choosing the wrong step size to increase crowded limit by.



[1]

---

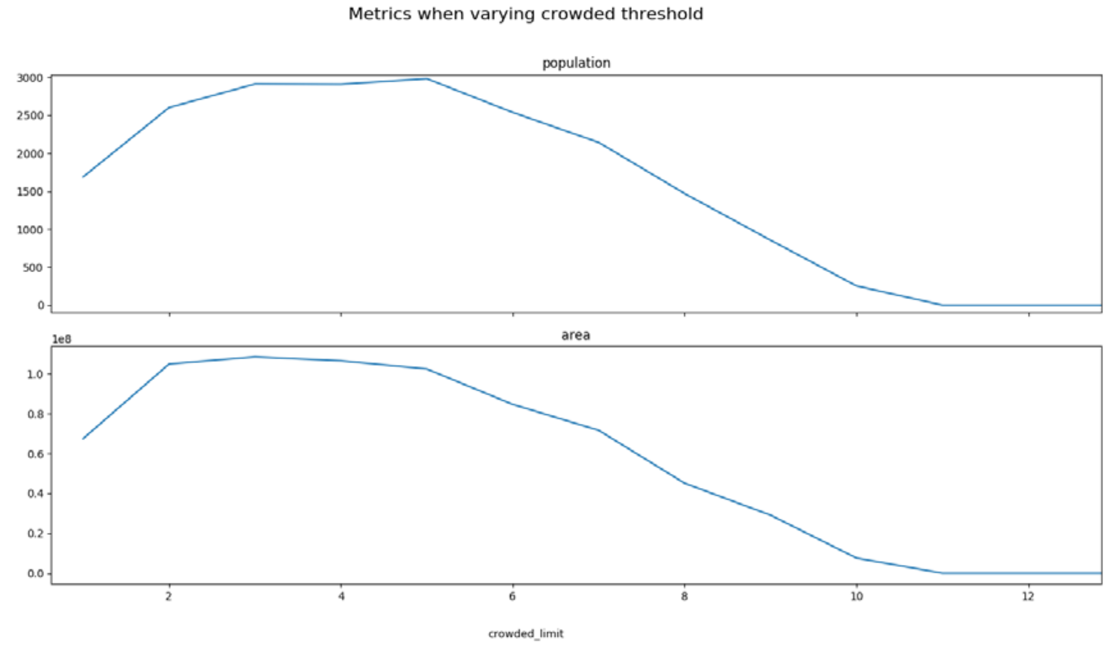
[1] Differences in results were usually caused by the rounding off of doubles in the simulation. Introducing infinite precision floating point numbers would make it completely deterministic.

B. When optimizing for maximum area the ideal threshold is 3 to 4 for maximum populations from 10-18, with 3 showing up 3 times and 4 6 times. Then the likelihood of the null hypothesis in "The ideal threshold for maximizing area = 4 given the other conditions remaining the same and a random maximum population between 10-18" being true is exactly 3/9[2] or $p = 0.333$. Thus we failed to reject the null hypothesis which requires $p < 0.05$. A more general hypothesis, IE that the ideal crowded limit is either 3-4 would have succeeded with $p = 0$ for all maximum populations between 10 and 18.

When optimizing for maximum population, the ideal crowded threshold is either 4 or 5 for maximum populations from 10-18. This outright fails the proposed hypothesis, that the ideal crowded threshold is determined by the maximum population, and that it is approximately 70% of it.

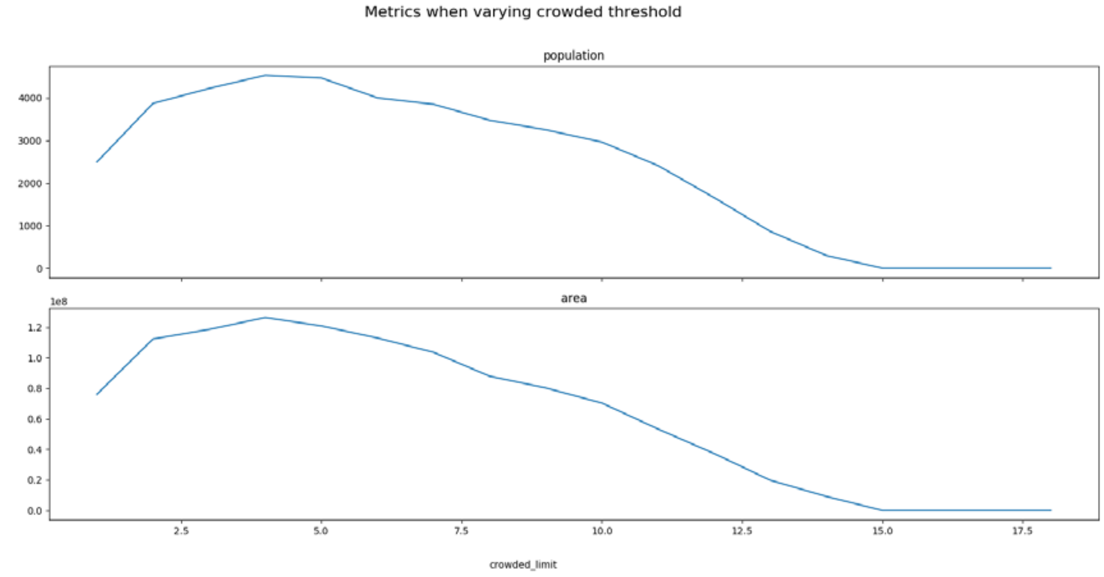
---

[2] Despite testing crowded thresholds 5 times for each combination of threshold and maximum population (between 1-18, and 10-18 respectively), my results for area and population remained exactly the same, even if other statistics like network size, slightly varied, so I could not use t-testing since area and population are completely deterministic.



Above, crowded threshold with a standard overpopulation of 10

Below, crowded threshold with a overpopulation of 14



Conclusion

Openlife's goal was to investigate the effects of parameter changes on the simulation's metrics. Specifically, group size threshold decreases network count, area, and population. If the variation step size is increased, group size threshold decreases complexity and increases group size but only when the maximum and minimum threshold values are not included. The effect of group size on entropy is insignificant in all cases.

The crowded threshold is one of the only real "intelligences" of the creatures, and including it provides them a large advantage. In an earlier version of the simulation, before crowded thresholds existed, creatures would only avoid others upon becoming overcrowded. This leads to vastly decreased growth, as is visible above when observing populations and areas at a crowded threshold of 14 (the same as the overpopulation limit). Here, population has only grown to 299 creatures in 25,000 seconds, compared to 4,534 creatures with a crowded limit of 4. Despite this, we inaccurately predicted at what points the crowded threshold would be most beneficial to the creatures' growth in both area and population, which happens at 3-5 regardless of the actual maximum population.

Doing a sample mean t-test between each set of parameters was not the best method of evaluating the effects of a parameter change. (It was overly dependent on choosing a good variance size for the parameter being changed). An analogy: When comparing test scores by amount of time spent studying, the scores of a student who studied for 15 minutes may vary insignificantly compared to the scores of a student who did not study at all, but if the variation in study time were increase to, say, eight hours, then the test scores might be significantly different. Also, doing an ANOVA t-test would be more effective, but would also suffer from an arbitrary parameter variance amount.

Openlife has millions of testable combinations of parameters and metrics, and it is an arguably simple simulation. Choosing good hypotheses is key to finding interesting results. In a future experiment, many more hypotheses could be designed and tested, and only the interesting ones reported.

Overall, the Openlife simulation could help many fields of science in the future. Like with the Crowded threshold, this simulation can be used to determine the efficacy of different evolutions in an ecosystem, which could be beneficial for Biologists looking to understand the purpose of certain mutations. Furthermore, the customizability of these creatures also makes the simulation ideal for discovering certain ecosystem niches, and creature traits that synergize together, which could be used to predict characteristics of future and undiscovered species. In Computer Science, Openlife is a valuable tool for modelling and developing swarm robots, machines that can perform complex tasks as a group with a simple list of individual instructions. These show promise for many fields, like that of medicine, where small robots could enter a patient and perform surgery or fight cancer more effectively than doctors. Small robots need to function on minute amounts of RAM and processing power, so they require a simple set of rules not unlike those present in our Openlife creatures.

The extension of the Openlife simulation and further investigation into parameter-metric combinations will likely help describe other systems and areas of science. Openlife has the potential to help learn more about the universe itself.

Significant Achievement

Openlife’s goal was to investigate the effects of parameter changes on the simulation’s metrics. Making a simulation from scratch with 100% modular parameters and robust automated parameter testing was by far the most difficult part of this project, but allowed us to find the most interesting results of our simulation. If we had used a more constricted method of simulation, our space for potential hypotheses would’ve been much more limited.

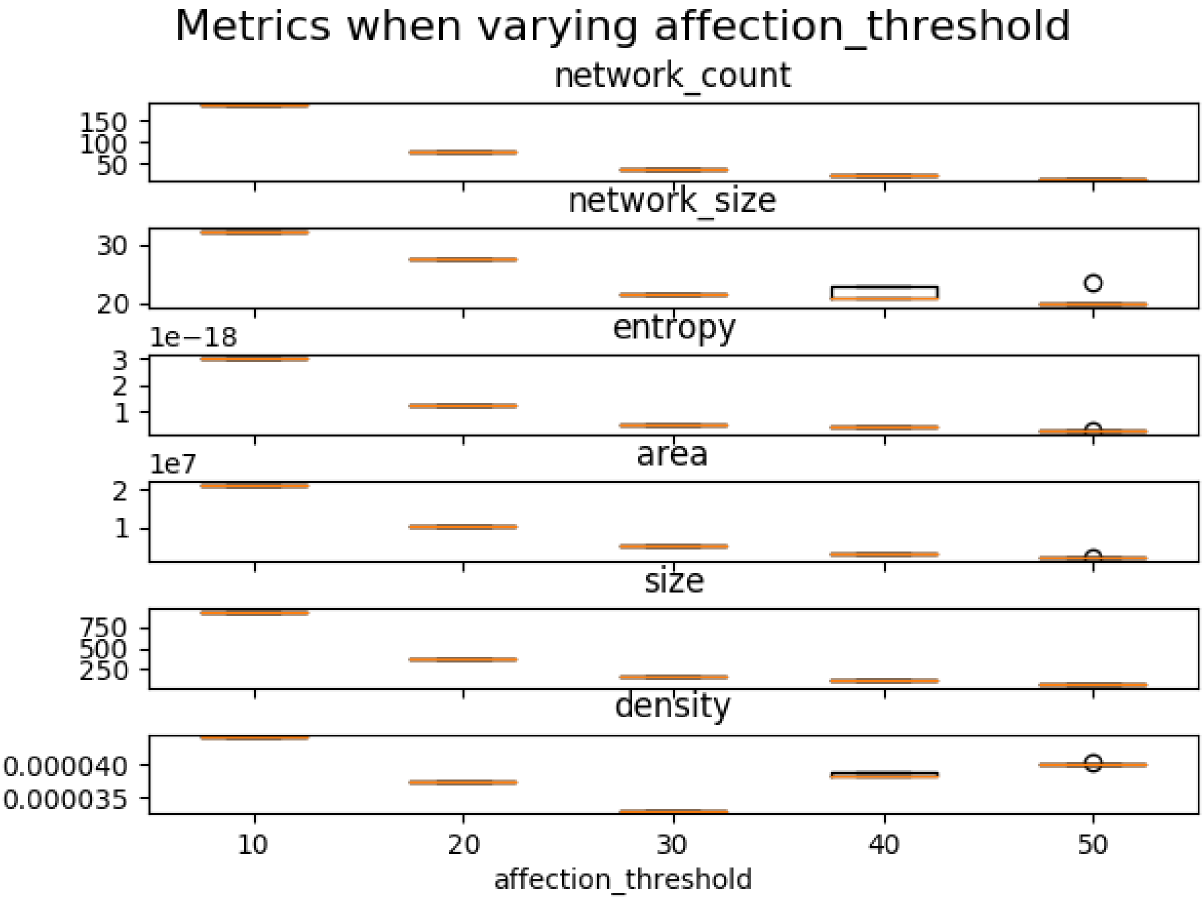
Acknowledgements

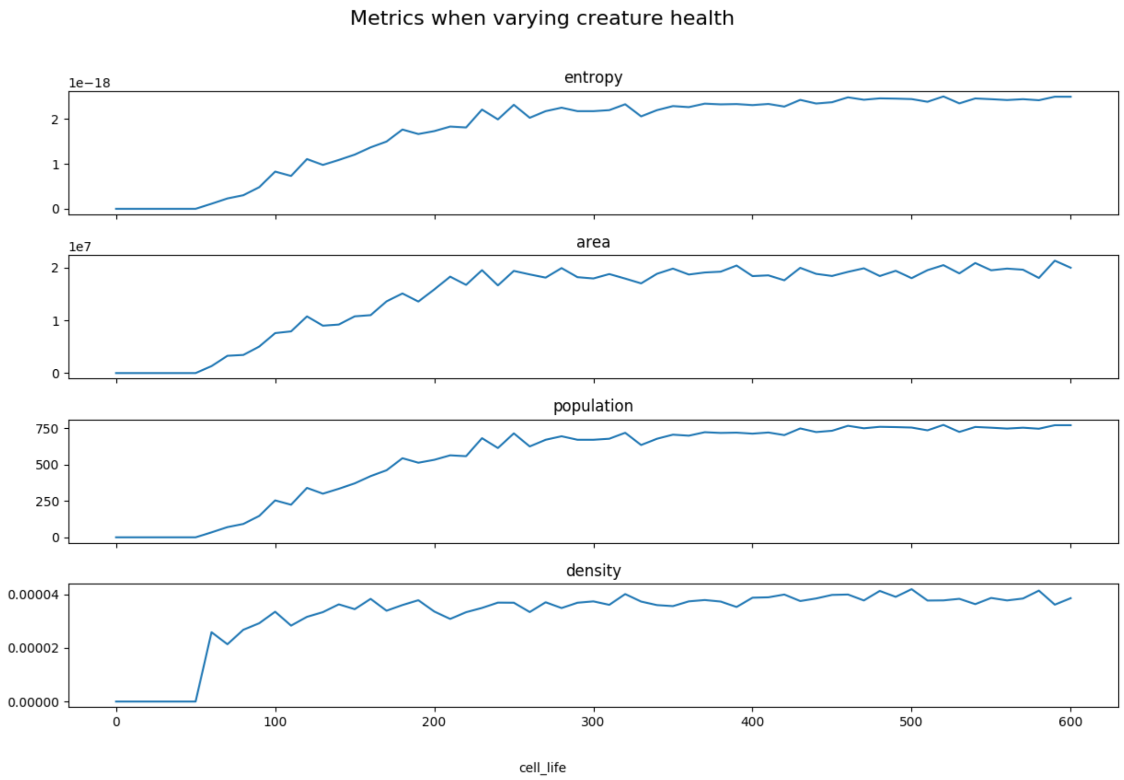
Special thanks to Neal Holtschulte, for encouraging us to take on this project and keeping us on track. His expertise and advice allowed us to turn Openlife into the simulation that it is today, and has paved the way for its future.

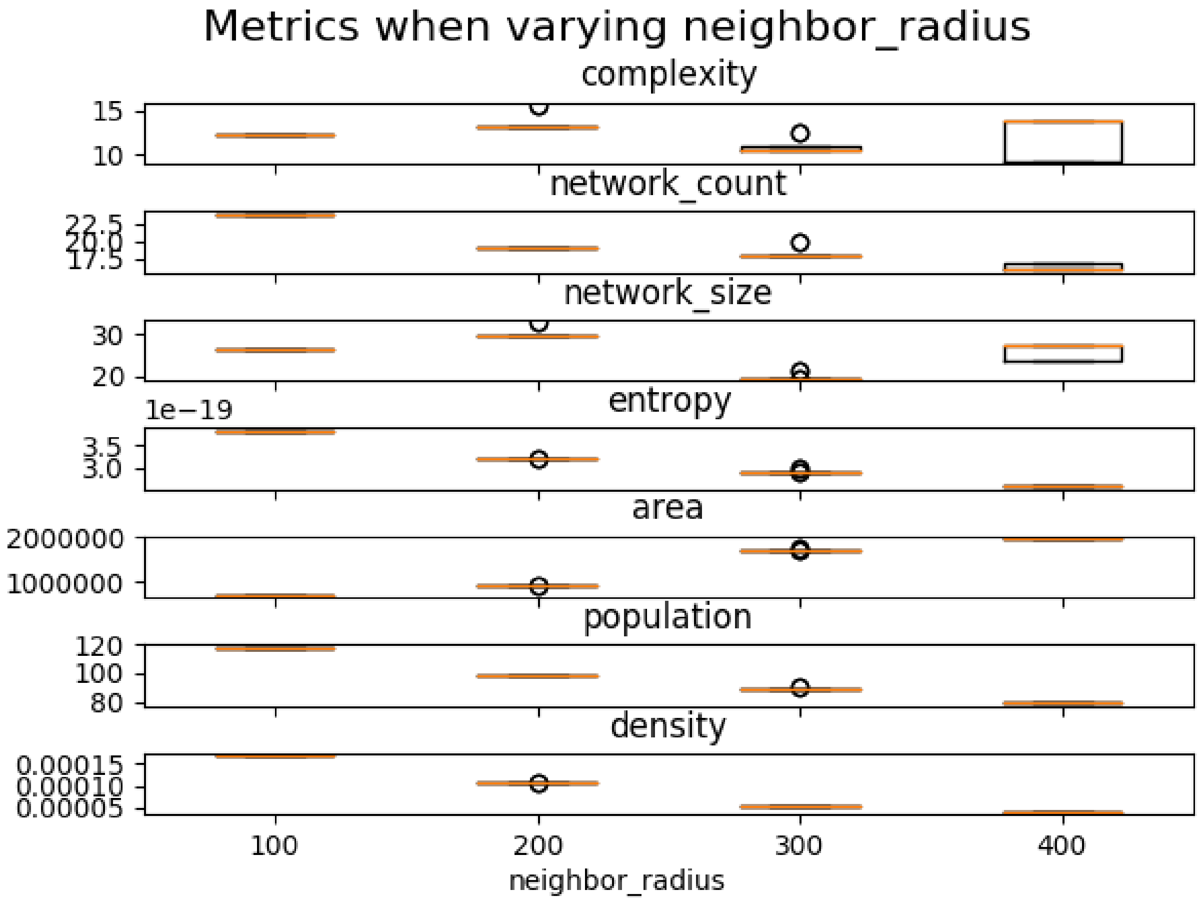
Appendix

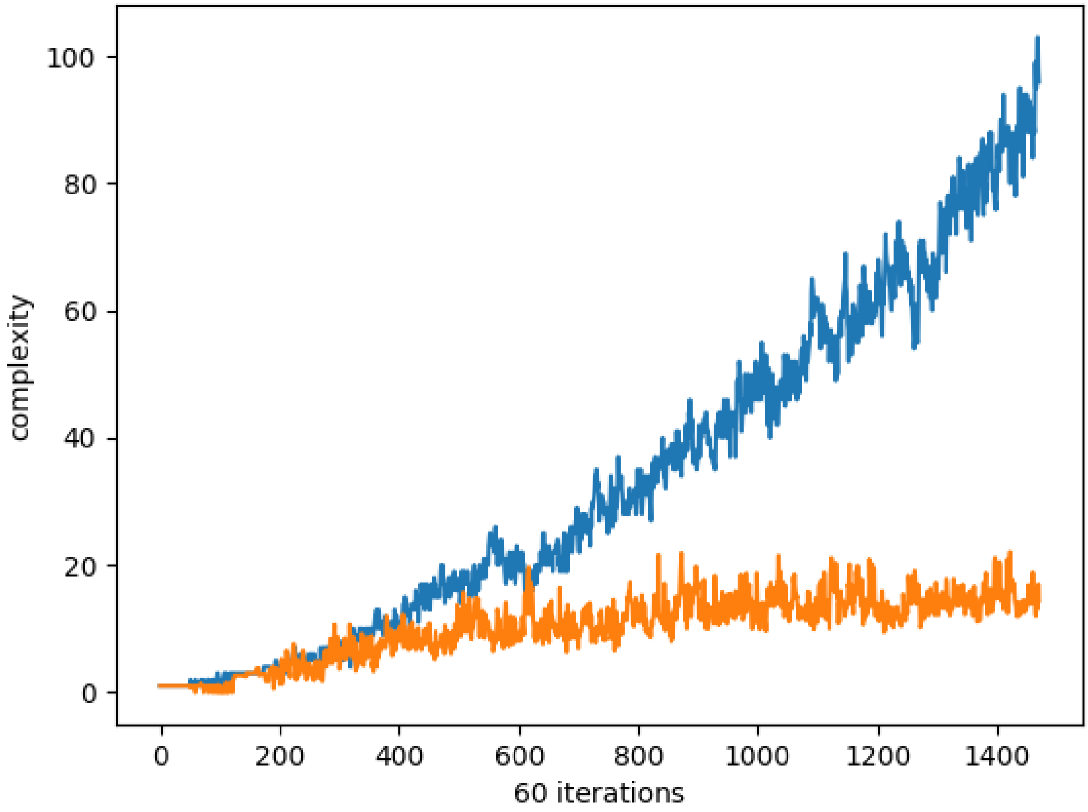
Github: https://github.com/natis1/openlife

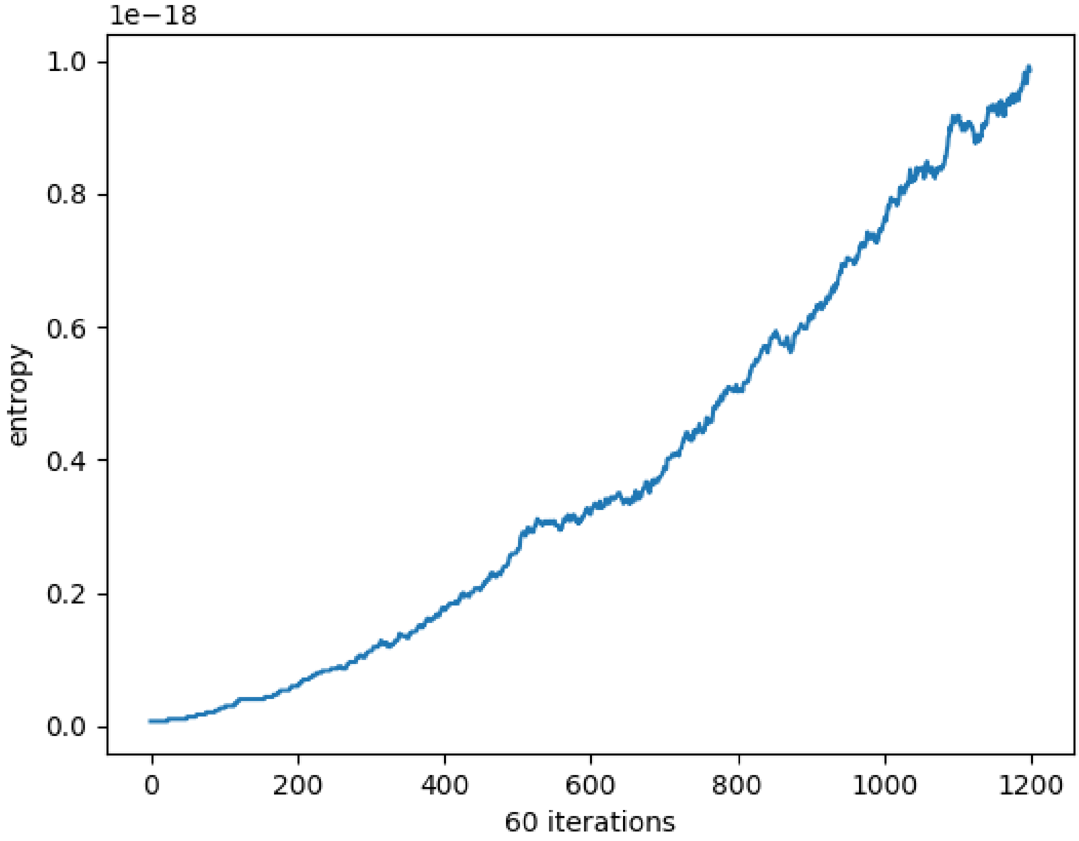
Potentially interesting parameter-metric combinations not mentioned above:

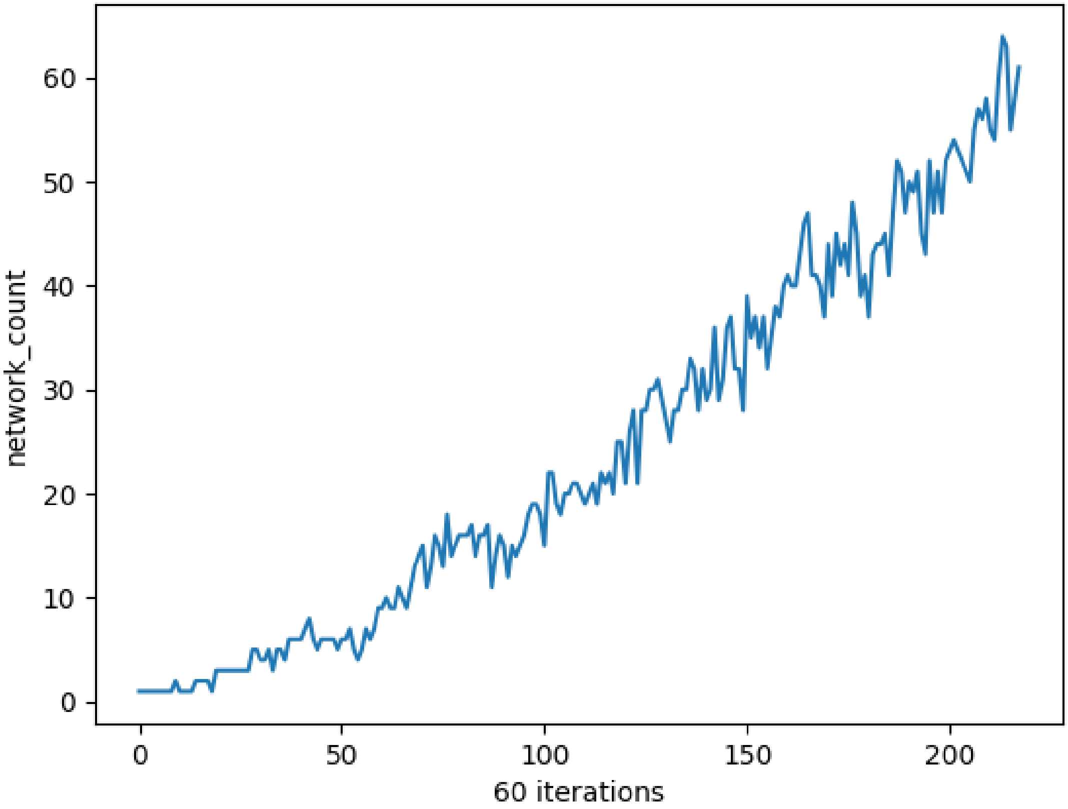


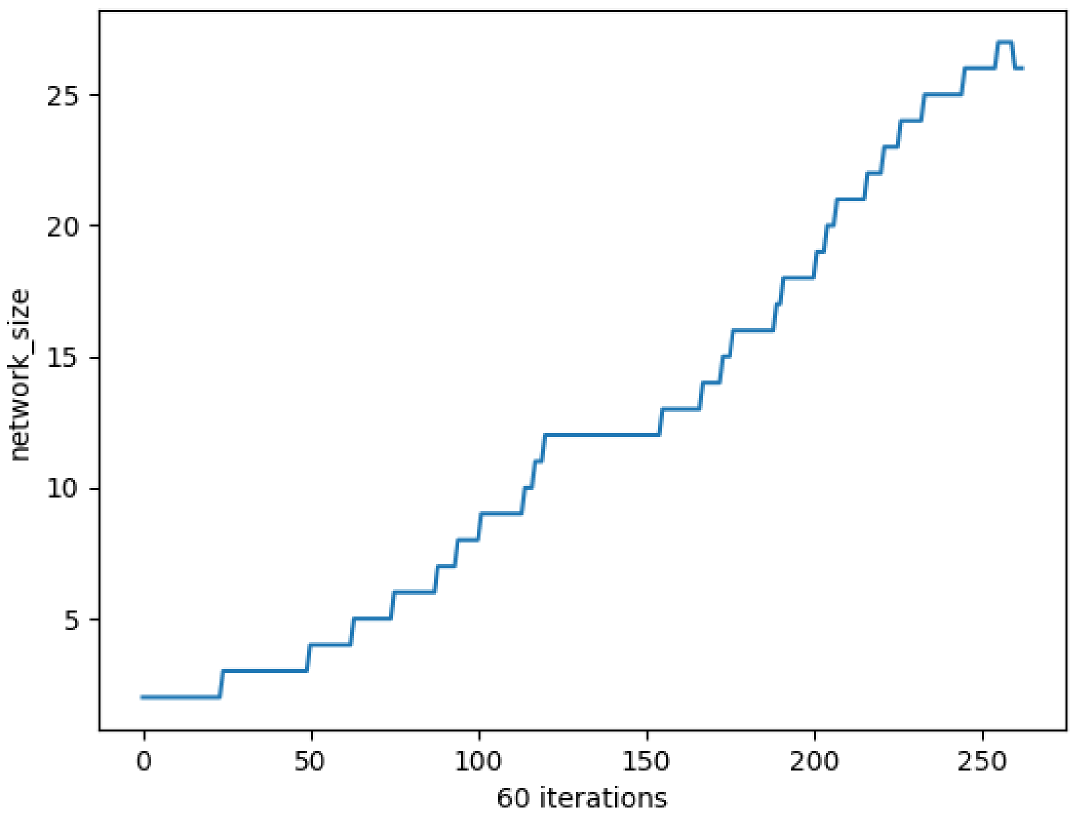

Metrics when varying creature health
entropy
1e−18
0
1
2
area
1e7
0
1
2
population
0
250
500
750
density
0.00000
0.00002
0.00004
0
100
200
300
400
500
600
cell_life


Metrics when varying neighbor_radius
complexity
10
15
network_count
17.5
20.0
22.5
network_size
20
30
entropy
1e−19
3.0
3.5
area
1000000
2000000
population
80
100
120
density
0.00005
0.00010
0.00015
100
200
300
400
neighbor_radius


complexity
100
80
60
40
20
0
0
200
400
600
800
1000
1200
1400
60 iterations


1e−18
1.0
0.8
0.6
0.4
0.2
0.0
entropy
0
200
400
600
800
1000
1200
60 iterations


network_count
0
10
20
30
40
50
60
0
50
100
150
200
60 iterations



Variables used for creatures in Openlife (unless otherwise stated)

mate_radius	100.0
neighbor_radius	400.0
search_radius	1000.0
move_modifier	10
underpopulation_limit	1
overpopulation_limit	10
crowded_limit	10
regeneration_amount	0.
underpopulation_damage	1.
overpopulation_damage	1.
affection_prime	1
affection_threshold	50.
kids_per_reproduction	1.
Turn_rate	50
creature_health	100
creature_size	10

Bibliography

[1] Aaronson, Scott. "Quantifying the Rise and Fall of Complexity in Closed Systems: The Coffee Automaton." http://www.scottaaronson.com/papers/coffee2.pdf

[2] Stanford University. "Game of Life." 07 Dec. 2016. http://web.stanford.edu/~cdebs/GameOfLife/

[3] Shapiro, Alex. “Sproutlife, evolving of conway’s game of life” https://github.com/ShprAlex/SproutLife/blob/master/README.md

[4] Ferreira, P. “*Tracing Complexity Theory*”. MIT, Oct. 2001. Ppt. http://web.mit.edu/esd.83/www/notebook/Complexity%20Theory.ppt

[5] Grobstein, Paul, and Laura Cyckowski. "Deterministic and Non-deterministic Emergence." *Deterministic and Non-deterministic Emergence / Serendip Studio*. Serendip Studio, 11 Dec. 2008. 07 Dec. 2016. http://serendip.brynmawr.edu/exchange/ca

[6] Bhatia, Aatish, “*Entropy Explained, With Sheep: From Melting Ice Cubes to a Mystery About Time.*” https://aatishb.github.io/entropy/

[7] Cornell University. "T-test reports" *Reporting the Results of a T-test*. 04 Apr. 2017. http://www.csic.cornell.edu/Elrod/t-test/reporting-t-test.html

[8] Stattrek. "Hypothesis Test for a Mean." *Hypothesis Test for Mean*. 04 Apr. 2017. http://stattrek.com/hypothesis-test/mean.aspx?Tutorial=AP