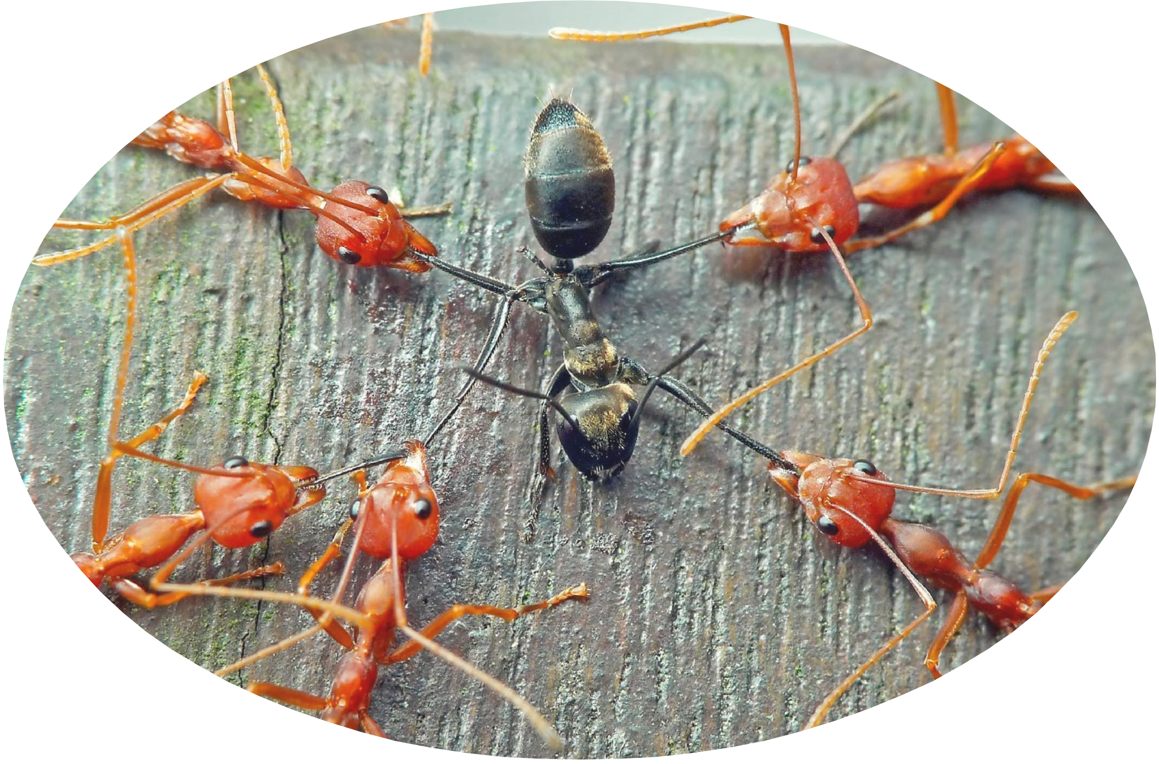
Institut Supérieur d'Informatique,
de Modélisation et de leurs Applications

Discrete Event Simulation course

"Multi-Agent Based Simulation: Ants"



Rim Shayakhmetov

Clermont-Ferrand 2016

Table of Contents

Introduction

When designing systems that are complex, dynamic and stochastic in nature, simulation is generally recognized as one of the best design technologies for analyzing systems. A wide range of existing application domains rely on the agent paradigm (such as social sciences, economics, robotics, complex systems, etc.). Luck et al. [1] make a distinction between two main multi-agent system paradigms: multi-agent decision systems and multi-agent simulation systems. In multi-agent decision systems, agents participating in the system must make joint decisions as a group. This project explores an artificial multi-agent simulation system based on some traits of ant behavior. In this model ants do not make joint decisions as a group, but simple rules and autonomous decisions can lead to the remarkable self-organization of ants. Moreover, the simplified model of ants behavior can show that the whole system changes its behavior in response to its environment (change in resource locations). General characteristics of multi-agent systems and a soft introduction to the field can be found in [2].

The project simulates a system with two competing teams, which contest for resources. The model does not represent an ant society in its beauty of form and function. Although, the model includes similar phenomenons as food trail pheromones and an ant’s kiss of food (which are not unique for ants in general). The mechanism of these phenomenons are simplified in this simulation, but the goal is to show how an ant colony can achieve complex behavior using simple rules for one ant. One can find an overview of recent results in a study of real-world ants building complex structures with a few simple rules in the article [3].

Model

World/Environment

Description

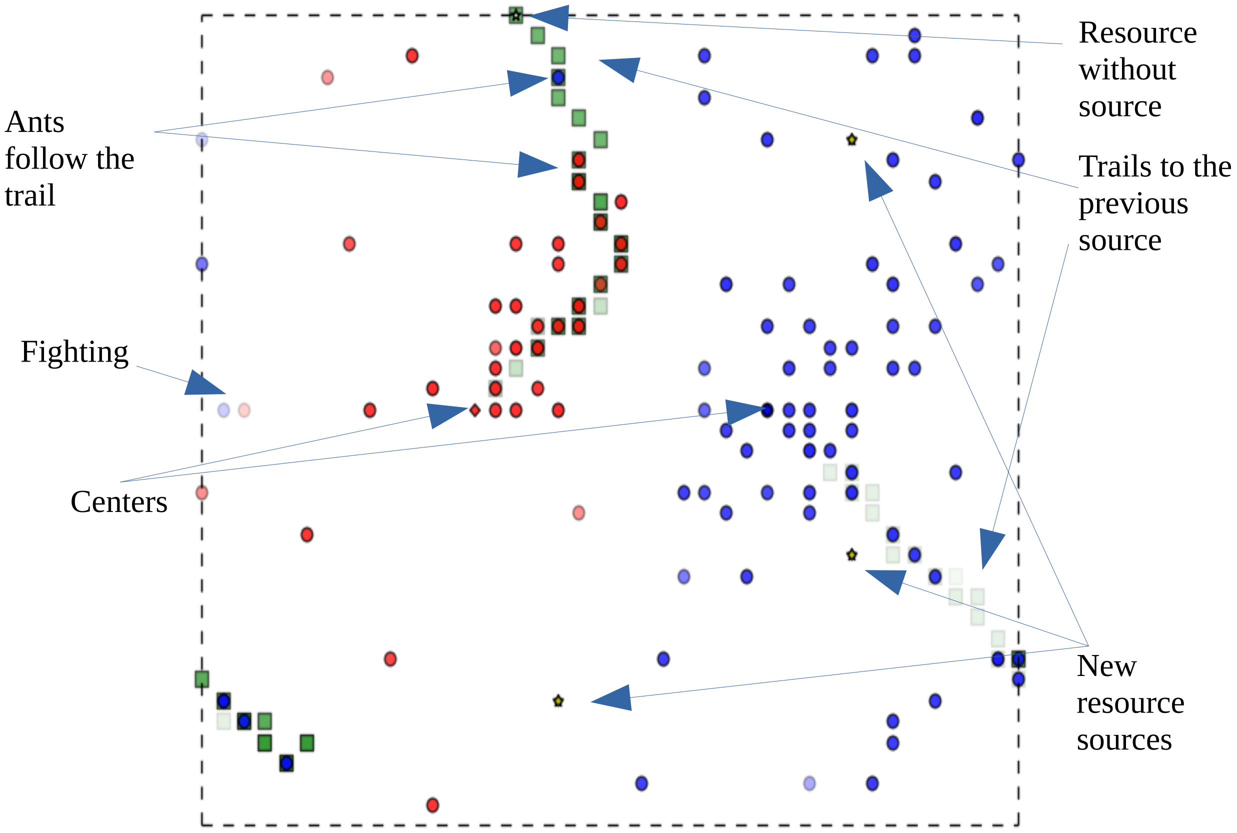
The world is a two-dimensional discrete torus with a defined size (for example, 40*40 cells), discrete time and 8-neighbor connections. Each cell can be empty, occupied by an ant or consist of a resource. In addition, each cell has trail pheromone concentration (value between 0% and 100%), which an ant can increase. Pheromone concentration decreases each turn (for example, by 3).

The world consists of two competing ant breeds (red and blue colored in the visualization). Each colony has its center in the torus, which is tied to some cell during the whole simulation. Each turn there is a possibility of creating a new ant for both ant breeds. Each turn, if the cell of center is not occupied by any ant, there is a probability (for example, 40%) to create a new ant in the center of the colony.

The world contains several sources of resources (for example, three). The locations of sources are changing with a defined period (for example, 100 turns). Each cell, corresponding to the source location, can produce a resource with some probability (for example, 80%) each turn if it is not occupied by any ant.

Visualization

The torus visualized as two-dimensional square, where agents can move from one border (black dotted line) to the opposite one. Each ant visualized as a colored circle (with transparency related to his health/power). Stars in the visualization are resources. Colored diamonds represent centers of each team. Green tiles represent pheromone concentration of cells (with transparency related to its value). Yellow diamonds show current location of resource sources (which are frequently accompanied by stars). One can see below the world after changing resource sources:



Simulation step

Simulation impose a strict order of executing actions (therefore, resolution of conflicts). One step of simulation consist of the next procedures:

- Try to add new ants
- Try to add resources / change sources
- Execute actions of each red ants according to their age (the oldest is first)
- Execute actions of each blue ants according to their age
- Decrease pheromone concentration in cells according to the defined decay rate

Ant

Each ant has strength/health/power (positive value with the maximum value of 100). When the value reaches 0 or below, the ant is considered dead. His strength changes when the ant found a resource, attacked or was attacked by an opposite ant, or participated in a food kiss with an ant of the same breed. Each ant can be either in a state of wandering (seeking for a resource, fighting) or in a state of found resource.

States

When an ant wanders, first, he tries to find a resource nearby. If there is no resource in the neighborhood and at least one opposite ant, he attacks him. Otherwise, he chooses an empty cell with the maximum amount of pheromones in it and moves in that direction. In a case when all his neighbor cells are occupied by his teammates, he initializes a food sharing kiss. In all cases when there are several equal possibilities the ant chooses the direction randomly.

In the second state after the ant found a resource, he eats it (receive boost in his strength/health) and leaves tail pheromones. Then the goal of the ant is to reach his center while leaving pheromones during his path. He chooses among empty cells those which have smallest distances towards his center. The distance is the Minkowski distance (p is infinity, can reach all cells in a square in one move, or 8-neighbor connectivity) with only extension of space being torus. Among empty cells the ant chooses those with the maximum amount of pheromones in order to keep and strengthen old trails. In this state he does not engage in a fight and his goal is to reach the center as soon as possible. When the ant reaches a neighbor cell of the center it switches to the default state of wandering.

Trail pheromones

When an ant finds a resource he receives immediate boost in health/strength (for example, up to 40 out of 100). Then he leaves pheromones all the way to his center. Each ant while wandering follows the direction of the maximum pheromone concentration. In order to achieve the goal of communicating the location of a new found resource to other ants, after each step towards the center he leaves slightly slower amount of pheromones.

In addition, an ant cannot increase concentration more than this cell had just before the action. The ant increases the current amount of pheromones to the level that he was supposed to leave before knowing the current concentration of the cell. More particular, if the cell had lower concentration, the ant increases it up to *100 - #steps_after_found*(decay_rate + ε),* where ε is much smaller than the decay rate and it ensures that the ant will create a path with slightly increasing pheromone concentration towards the resource location. This helps other ants (all ants have the same pheromone sense) find the resource quickly using only simple rules that were described before.

Ant-Ant interactions

If a food kiss initiated by some ant towards an ant of his own breed, they remain on their cells after the action. Their average health/strength is assigned to each of them after the action completed.

If there is an interaction between two opposite ants, they engage in a fight. The attacker receives advantage of executing the action first, but looses some strength afterwards (for example, 5 out of 100). The ant, being attacked, looses much more health (for example, 20 out of 100). If some of them reaches non-positive strength, he immediately dies. If the attacker dies, the other ant stays on his cell, but if the attacked ant was killed, the attacker moves to his cell.

Making several remarks, if two opposite ants are in the wandering state, they will probably continue to fight all next steps until someone will die (or both). Secondly, if one ant in a found resource state, he will continue to construct the path to his center, even being attacked. Finally, the fighting was purposely made to be locally in order to simplify the model whereas in the nature ants have different pheromones for different events (warlike alarms). This simplification has led to relatively more peaceful life among two breeds.

Tools

Programming

For programming, Python 3 was used (Jupyter Notebook was used only for convenient graphs plotting and statistical analysis of results). Dependencies of the simulation program are:

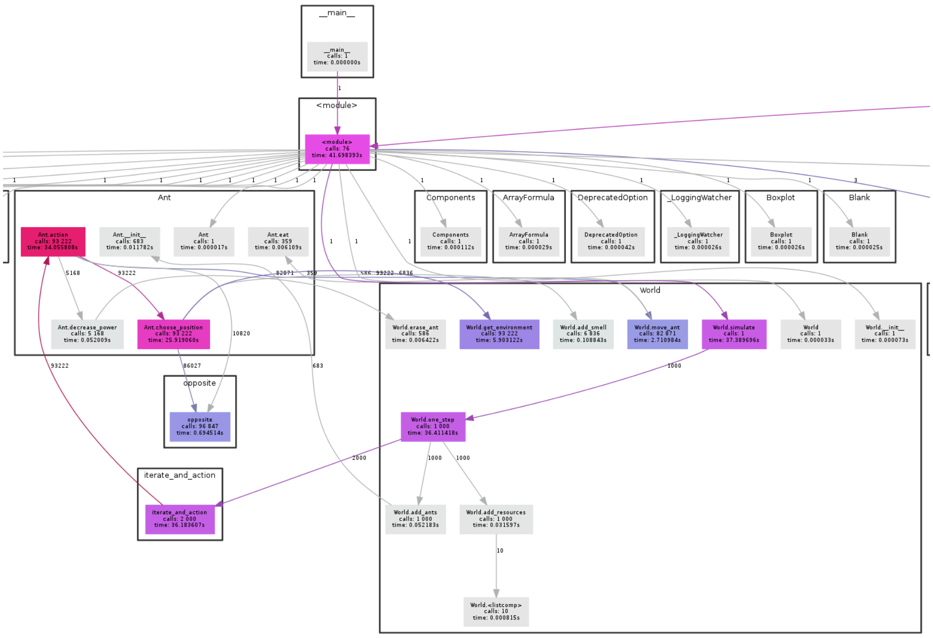
- NumPy: fundamental vector operations, random number generation (Mersenne Twister pseudo-random number generator), etc.
- Matplotlib: graphical visualization of the world (redrawing the grid two times in one turn after each team's move), graphs plotting.
- Pandas: convenient for processing statistics accumulated during the simulation.

Git

During all steps of development (prototyping, testing, adding new features, etc.) Git was used as a convenient way to store new versions of program and restore old code. Apart from being used locally, public repository was created on GitHub. One can find the whole project available at [4]. The main program in the ".py" file, whereas the code, analyzing statistics of several simulations, located in the ".ipynb" file.

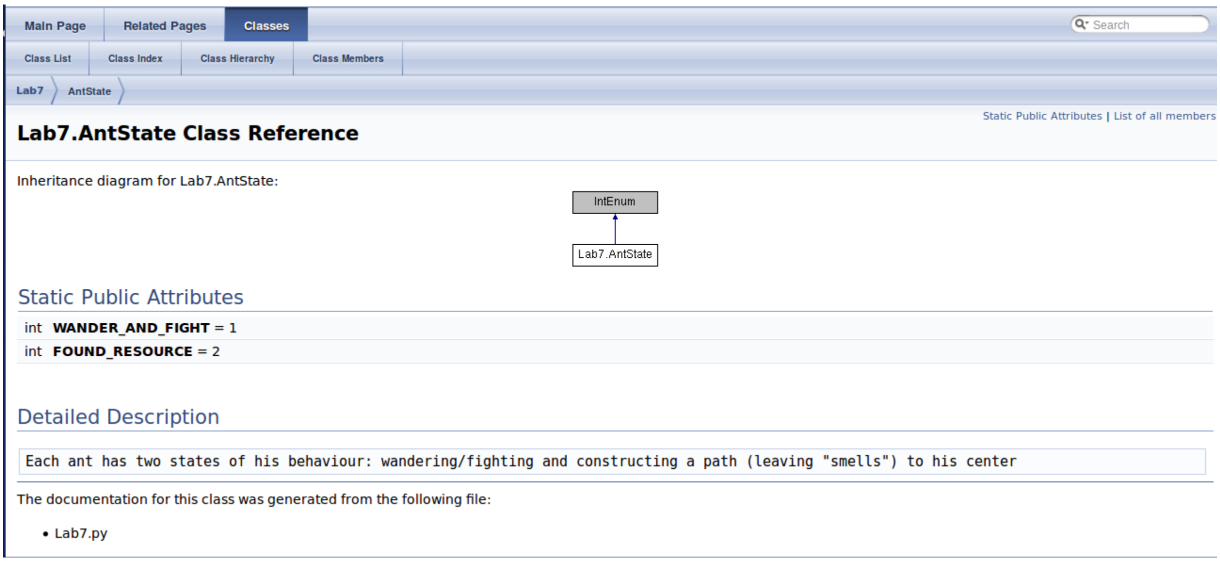
Profiling

For profiling, Python Call Graph was used [5]. It uses graphical presentation of a call graph and biggest bottlenecks can be seen easily and investigated thoroughly. One can see below a part of a call graph, containing all main classes. Here we can see the bottleneck is the *action()* function of the Ant class. It encapsulates all ant's logic in the states defined above. This graph was generated for only the simulation without analyzing GUI.


__main__
__main__ calls: 1 time: 0.000000s
1
<module>
<module> calls: 76 time: 41.698393s
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 3
1 1 1
Ant
Ant.action calls: 93 222 time: 34.055808s
Ant.__init__ calls: 683 time: 0.011782s
Ant calls: 1 time: 0.000017s
Ant.eat calls: 359 time: 0.006109s
5168
93222
82871
359
586 93222 6836
Ant.decrease_power calls: 5 168 time: 0.052009s
Ant.choose_position calls: 93 222 time: 25.919060s
10820
93222
86027
683
opposite
opposite calls: 96 847 time: 0.694514s
iterate_and_action
iterate_and_action calls: 2 000 time: 36.183607s
2000
Components
Components calls: 1 time: 0.000112s
ArrayFormula
ArrayFormula calls: 1 time: 0.000029s
DeprecatedOption
DeprecatedOption calls: 1 time: 0.000042s
_LoggingWatcher
_LoggingWatcher calls: 1 time: 0.000026s
Boxplot
Boxplot calls: 1 time: 0.000026s
Blank
Blank calls: 1 time: 0.000025s
World
World.erase_ant calls: 586 time: 0.006422s
World.get_environment calls: 93 222 time: 5.903122s
World.add_smell calls: 6 836 time: 0.108843s
World.move_ant calls: 82 871 time: 2.710984s
World.simulate calls: 1 time: 37.389696s
World calls: 1 time: 0.000033s
World.__init__ calls: 1 time: 0.000073s
1000
World.one_step calls: 1 000 time: 36.411418s
1000
1000
World.add_ants calls: 1 000 time: 0.052183s
World.add_resources calls: 1 000 time: 0.031597s
10
World.<listcomp> calls: 10 time: 0.000815s

Documentation

For documentation generation, Doxygen tool was used [6]. It requires only proper comments for every function and class. Some UML diagrams were generated automatically. The documentation available at [4] in the “html” directory. One can see below an example of a page describing small class AntState:



Lab7.AntState Class Reference

Inheritance diagram for Lab7.AntState:

Static Public Attributes

int **WANDER_AND_FIGHT** = 1
int **FOUND_RESOURCE** = 2

Detailed Description

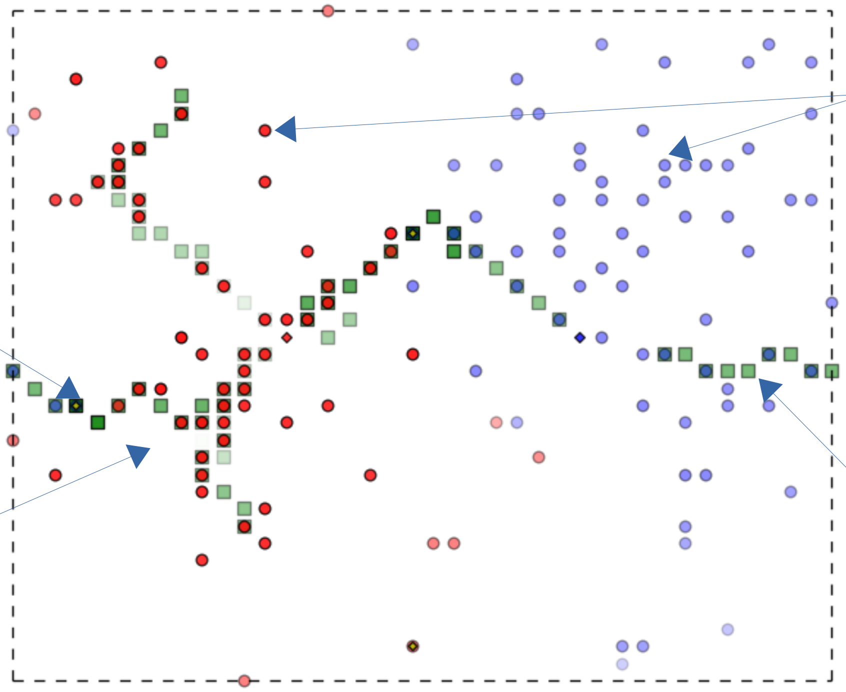
```
Each ant has two states of his behaviour: wandering/fighting and constructing a path (leaving "smells") to his center
```

The documentation for this class was generated from the following file:

- Lab7.py

Visualization

For visualization, only Matplotlib primitives were used. The axes of canvas were disabled and each agent or object is visualized using *scatter()* function.



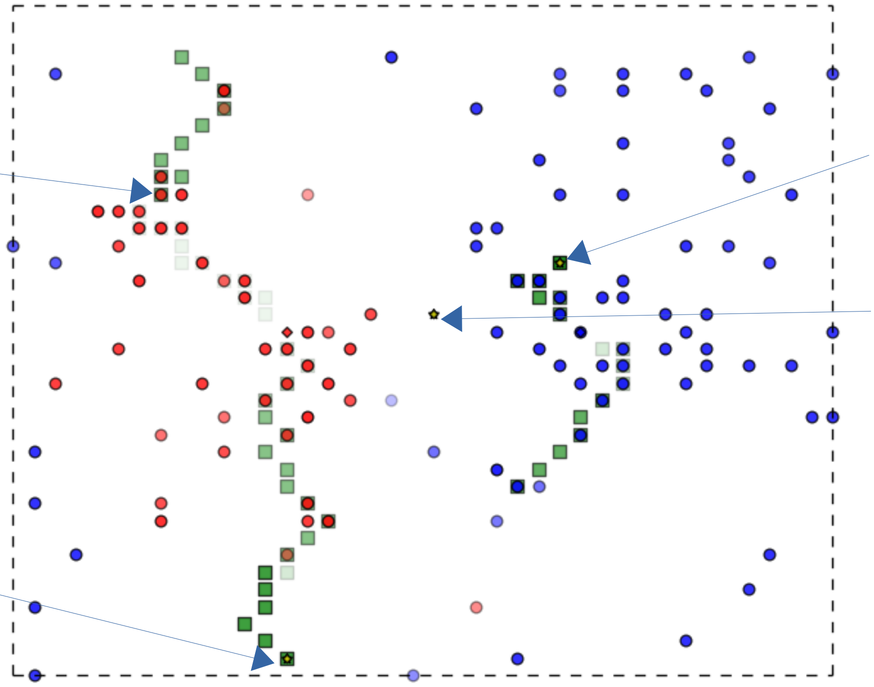
Yellow diamonds with a star correspond to a source with a resource. It is visible atop of agents

Red and blue color transparency represents strength of agents

Complex pheromone trails connecting both colonies (8-neighbor connectivity)

Green color transparency represents concentration of pheromones

Simulation



The ant has taken the last resource from the old location

Blue ants were lucky to find a new resource nearby

New source of resources has not been found yet

The new source of resource has been found quickly by red ants

Will blue ants get dominance over red ones? For how long?

The program in Python using graphics allows visualization of one simulation which is useful for deeper understanding of the system behavior. But in order to draw some solid conclusions it is necessary to conduct several independent experiments. The time limit of 10000 steps was set in all experiments. During this period there were one hundred changes of resource sources on a 40*40 torus. This number of steps is considerably enough to draw some asymptotic conclusions. The number of experiments was set to 40 (each experiment contains simulation of 10000 steps).

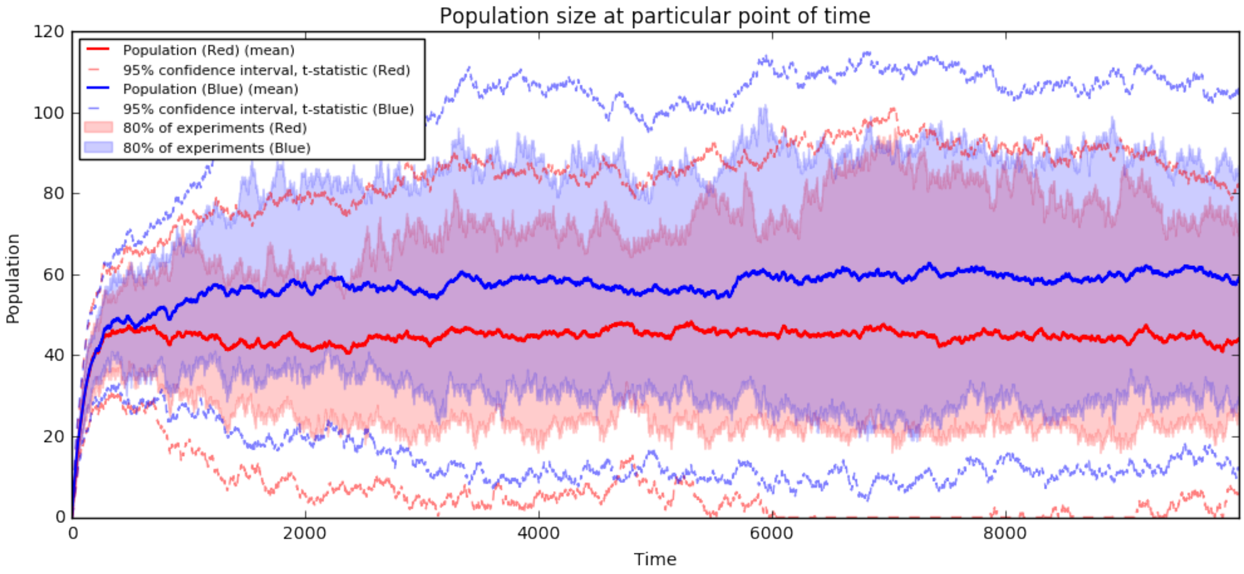
The class World stored step-by-step evolution of the next characteristics:

- Population size of each team
- Number of deaths (all deaths occur during fighting)
- Number of absorbed resources by each team

Same breed

First, all experiments were conducted using the same characteristic of breed. Both breeds were identical in birth probabilities, power, etc. One can expect that all characteristics must be the same at least on average. Some interesting question arise before experiments. For example, do population sizes converge to some number by 10000 steps? If yes, are they close to the upper bound of 1600 (all cells)? This can happen even in the case of the same breed. If there is an occupied center the ant colony cannot produce new ants. In other hand, parameters of the model and rules of constant reproduction and fighting can lead to eternal (at least big number of steps) struggle between two breeds.

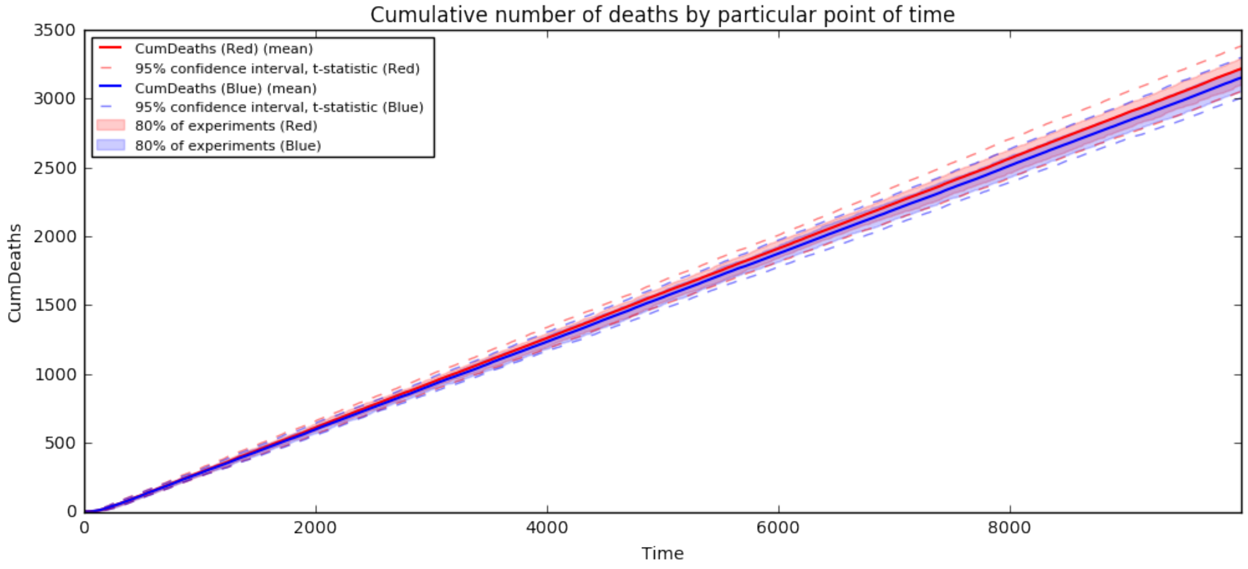
40 experiments show the next graphs of described characteristics. One can see below the population size evolution. Red and blue dotted lines above and below the filled area represents 95% confidence interval based on t-statistic (we can assume that each distribution in a particular time are almost normal, it converges by the Central Limit Theorem). The filled area represent an interval between 0.1 quantile and 0.9 quantile with linear interpolation (80% of all values are inside it).



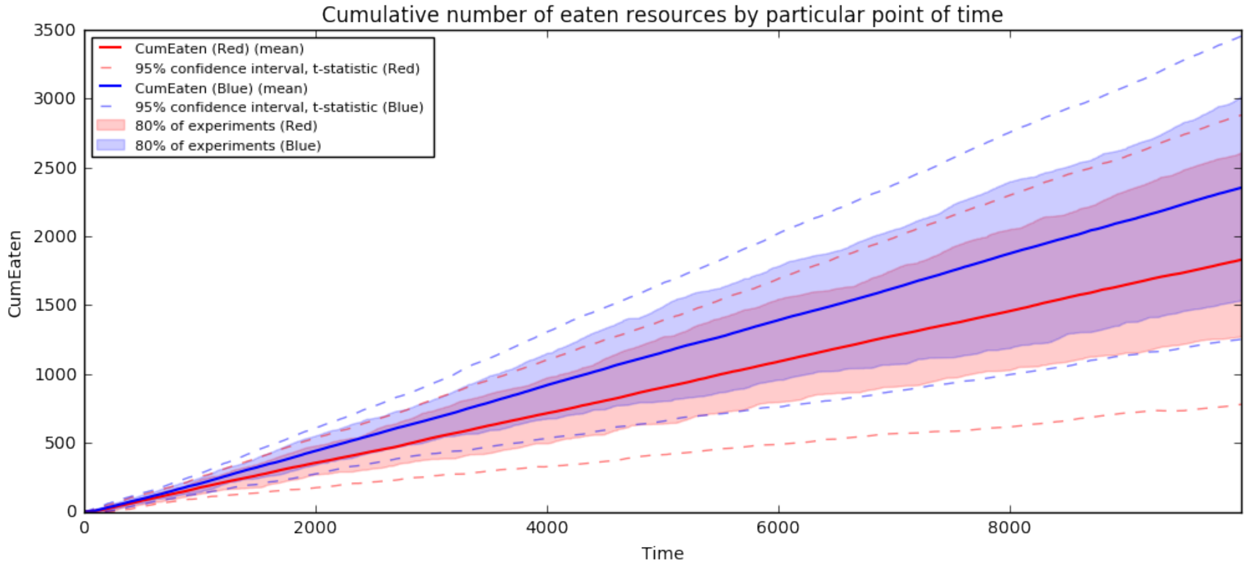
First, we can see that neither red population mean nor blue one lies outside of 120 (the upper bound of confidence intervals). This means that during first 10000 steps there were no situation of total dominance (the control of 1600 cells), although sometimes blue or red ants can almost disappear for some time. But the recreation and new born ants will not fall into total dominance of other ants. If we take one random cells in the environment, on average, the prior probability that it is not occupied by any ant is almost 97%.

The small rate of birth and only local fights are the main reasons for explaining the result. It is worst to note that slightly differences between population sizes of red and blue are not statistically significant and might be due to the small number of experiments (the number of experiments is much lower than the number of simulated steps). In addition, we can see that population sizes means almost converge to approximately 50.

The next characteristics are represented by cumulative sums up to a particular time. One can see the cumulative deaths of each team. There are no significant difference between two teams. The upper limit of the mean gives us almost 3500 cumulative deaths per team by 10000 steps. Combining the fact that cumulative deaths is almost a linear function of time and the fact that the average population size remains the same, we can predict number of deaths by some time very easily.



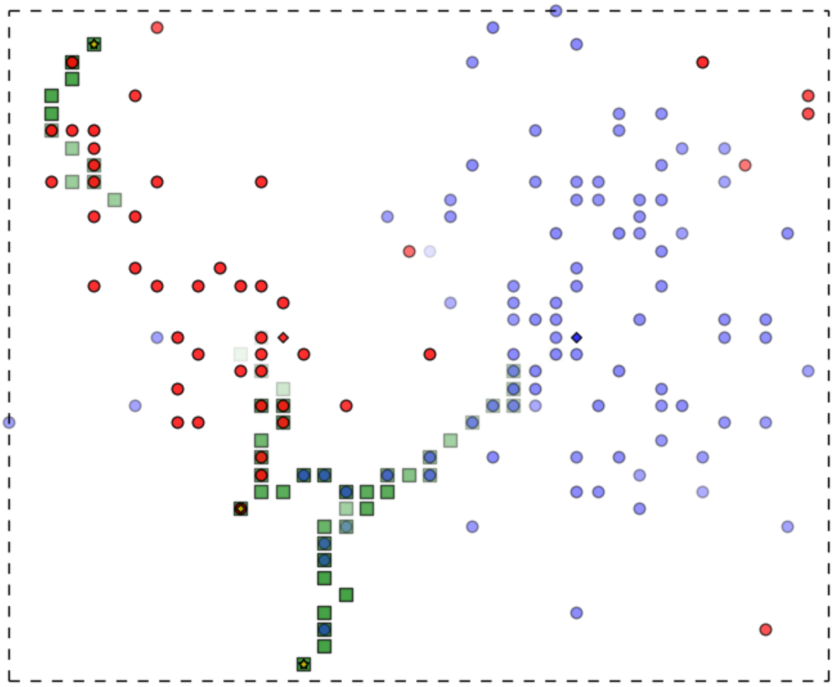
Last, there is arguably the most important characteristic – number of absorbed resources. In this model ants were designed to give first priority to resources rather than fighting. Therefore, the total number of accumulated resources can be a good indication whether a team was more successful (more adaptive, strong, etc.). One can see below the evolution of cumulative number of absorbed resources:



Although, we can see some changes in the total number of absorbed resources on average, the difference is not significant.

Different breeds

“Twice as strong, the red ones are, born twice as fast, the blue ones are”

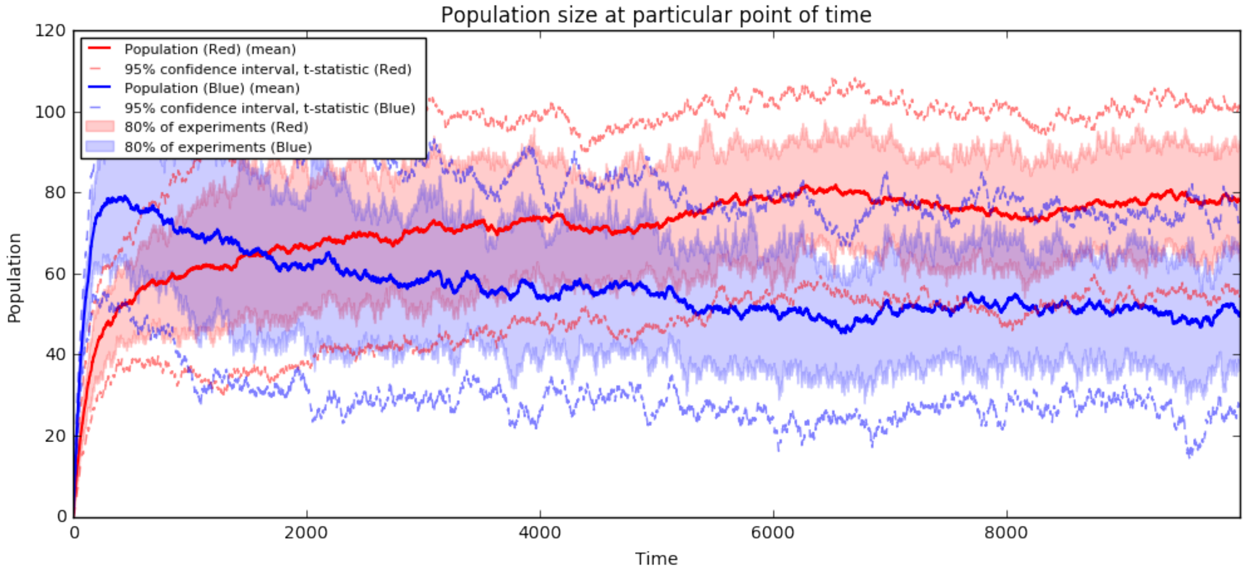


Will blue ants get dominance over red ones?

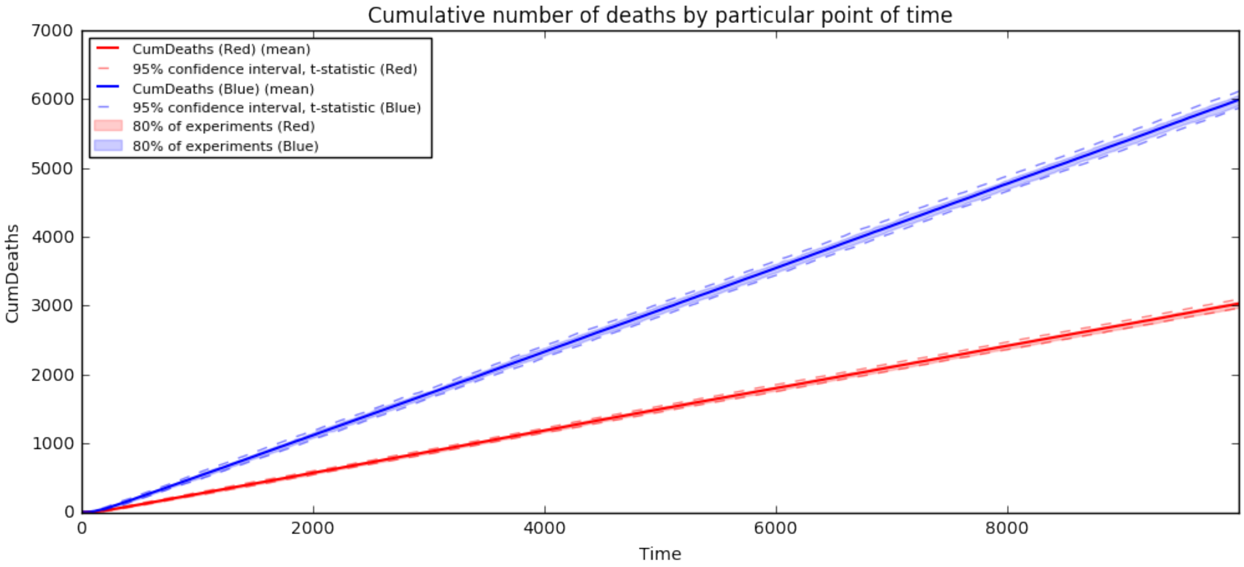
After understanding the outcomes of the same breed ants simulations new interesting questions come out. For example, which breed would be better suited for this world, the one that have double strength/power/health, or the one that have one additional possibility of birth each turn (with the same probability)? Some can expect them to behave almost similar to the previous examples (an argument is that the sum of powers of each ant breed will be almost the same). Some can expect more powerful to dominate the environment (an argument is that they will easily destroy weak opposite ants, which would be scattered randomly across the grid). Finally, some can expect that the doubled number of ants will find resources better and outbalance others by their number during fighting.

These are some valid points but it is difficult to determine a priori which outcome is more likely even in this simple model of the system. This is where simulation becomes a very powerful tool to answer those kind of questions. As usual, the number of experiments was set to 40 and the number of steps in each simulation was set to 10000. Red ants have twice maximum power (200) and blue ants have 100 as the maximum as usual. Blue ants have additional pseudo-center one cell closer to the blue center, which will provide an additional new ant each turn according to the same rules as centers do.

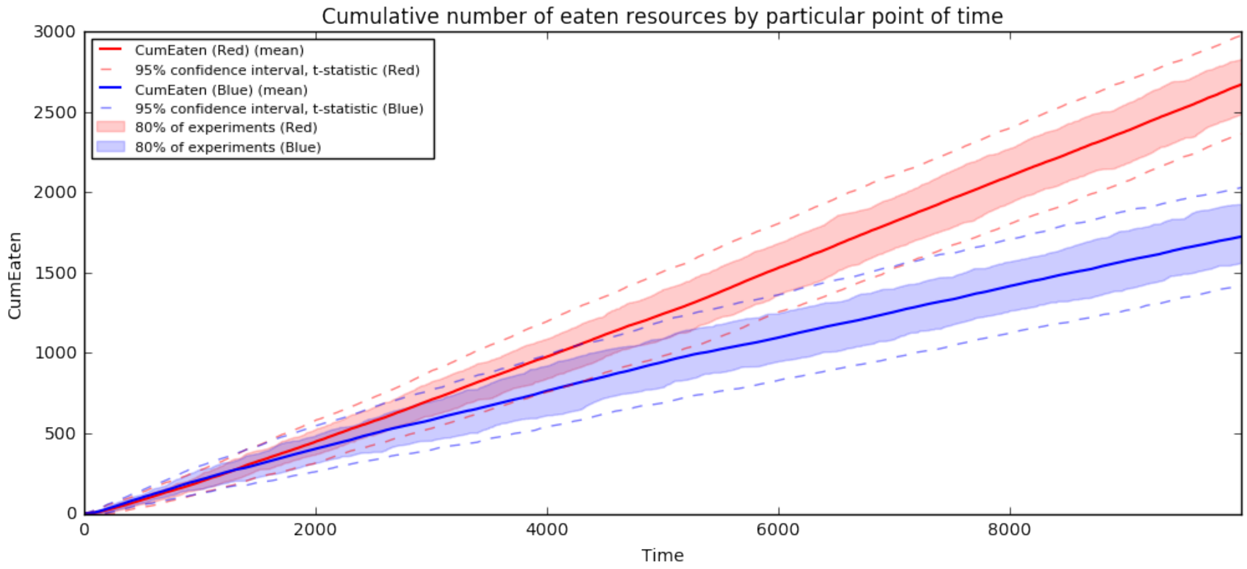
Below one can see the change in population sizes of each team. One can see the effect of two birth possibility in the first hundred of moves. During only those steps the difference in population between red and blue ants was significant. Next thousands of moves red slightly increased the average among all experiments, but by 10000 and using 40 experiments it is still not enough to say that the population size of red ants is significantly larger than blue ones. There is a possibility of red ants to dominate blue ones in the future, according to the slightly increasing trend of red population and slightly decreasing trend of blue population.



One can see below the next graph, which shows the cumulative number of deaths. It was somehow expected to be so, because of the difference in power between ants. The number of cumulative deaths among blue ants is twice larger than among red ones.



The last graph shows the most interesting statistic, which can help us to answer the question of who is better suited for resource collection. One can see below that starting from approximately 6000 moves red ants gain significant advantage over blue ones in terms of collecting resources. More rigorous analysis shows that among 40 experiments red ants always collect more in the long run (although, they might not succeed during first 7000-8000 steps). By 10000 steps the worst case scenario for red ants (among 40 experiments) was to gain only 300 resources more than blue did.



Putting all together, with the ability of double strength red ants are going to dominate most of resources, local battles and their population is expected to slightly grow further after 10000 steps.

Many factors caused this long run effect of red ants' slightly dominance (including the world's structure, probability rules, the ant model, etc.). In fact, one can easily show up with bunch of possible explanations (which were presented above), but it is necessary to be rigorous in making causal statements and, for example, simply quoting "Life is not about quantity, it's about quality" is not enough (the quote belongs to Malorie Blackman, a British writer).

Conclusion

In this project multi-agent based simulations were studied using an example of a simple model of ants. Development was accompanied by other tools such as a debugger, a profiler, and a documentation generation tool.

Simulations showed how using simple rules for autonomous decisions agents can be part of complex behavior of the whole system. The behavior of the whole system changed in order to fit to current locations of resources.

Simulations involving two identical breeds showed constant struggle between two colonies in a long run. The ant model designed to prioritize resource collection over fighting played a role in the results.

Simulations involving two different breeds showed that in a long run those who had twice as strong power were slightly taking control over those who had twice many births.

Bibliography

1: Luck, M., McBurney, P., Shehory, O., & Willmott, S., Agent technology: computing as interaction (a roadmap for agent based computing), 2005, Liverpool, UK: AgentLink, url=http://www.agentlink.org/roadmap/al3rm.pdf
2: Peer-Olaf Siebers and Uwe Aickelin , Introduction to multi-agent simulation, 2008, PO Siebers, url=https://arxiv.org/pdf/0803.3905.pdf
3: Emily Singer, The Remarkable Self-Organization of Ants, 2014, QuantaMagazine, url=https://www.quantamagazine.org/20140409-the-remarkable-self-organization-of-ants/
4: GitHub repository Multi-Agent based Simulation: Ants, url=https://github.com/shayakhmetov/multiagent_simulation
5: Python Call Graph, create call graph visualizations for Python applications, url=http://pycallgraph.slowchop.com
6: Doxygen, Generate documentation from source code, url=http://www.stack.nl/~dimitri/doxygen/