Simulation of Swarm Intelligence by Nanobots in a Blood Vessel

Guido Linders
Thomas Stein
Joost van der Molen
Winand Renkema
Jelmer Alphenaar

July 4, 2014

Abstract

Recreating swarm intelligence has become very popular during the last few years, especially in the field of artificial intelligence. Such swarms are characterized by a decentralized way of agents working together to accomplish goals, just like a lot of animals, such as social insects, bees, flocks of birds and schools of fish. The advantage of this is their flexibility, even when making a lot of mistakes, the swarm can still be very successful. In this paper we are going to focus on making an artificial implementation of the flow of blood in which nanobots are placed and should neutralize threats such as bacteria. There is still a lot that can be learned from swarm intelligence and the optimization of it.

Introduction

Swarm intelligence is the collective behaviour of a large group of agents that individually perform simple and logical actions. This collective behaviour is characterized as being very complex even though the individual behaviour of each organism is very simple and logical. An excellent example is a colony of ants: while an individual ant only performs simple tasks like following or leaving scent trails, the colony as a whole is able to accomplish more complicated goals, such as finding the shortest path to food sources or architecturing the structure of a nest. The behaviour of such a group is self-organizing. (Blum & Xiaodong, 2008)

Swarm intelligence doesn't have an (intelligent) leader. All individuals act in a coordinated way without a specific leader and show intelligent behaviour as a collective. Because of the simple rules and large amount of individuals neccessary for this, swarm intelligence can easily artificially be designed to be scalable, parallel and fault tolerant. (Dorigo & Birattari, 2007)

1. Scalability means that the system can easily increase in size without needing to add additional rules for the agents, because exchange of information happens through the environment. Simply increasing the amount of agents in a larger system will provide a sufficient result.

2. Parallel action can occur, because each agent acts individually and can thus perform a different action at the same time at a different location. This is desirable since a small group within the swarm can accomplish different parts of a larger complex task.

3. It really doesn't matter much if one of the agents doesn't function correctly, due to the decentralization. In the swarm this agent can easily be replaced by another. The behaviour is part of the whole swarm which makes the system fault tolerant.

Creating swarm intelligence for robots in the medical world is becoming increasingly popular. There has been a lot of research in this area already. Most of this research primarily focuses on the communication between the nanobots. In a research by Ghada Al-Hudhud (Al-Hudhud, 2012) a model for communication between nanorobots is created. The communication is both decentralized and centralized in order for the nanorobots to perform their task which is searching lipoprotein cholesterol threshold concentration in a specified location in a blood vessel.

Our research will focus on the (swarm) intelligence and not so much on what the best way is for the nanobots to communicate. We will assume that each nanobot can send pulses and receive pulses from other nanobots. We will use this communication method to simulate swarm intelligence.

Experiments

For our research, we aim to create a two-dimensional simulation of nanobots floating around in a bloodstream. We will attempt to have our nanobots display swarm intelligence as a whole by defining simple individual behaviour. We will document how these individual changes affect the behaviour of the mass of bots. The goal for these nanobots is to neutralize a pathogen in the virtual bloodstream. We will work with nanobots who are able to move left and right inside the arteries, but not change their velocity in the direction of the bloodstream.

Furthermore each nanobot can send and receive pulses as mentioned earlier. A nanobot will send pulses when it collides with an antibody or pathogen. This is to warn other nanobots that there is a pathogen nearby. These pulses have a small range and duration in which they can reach other nanobots. If a nanobot will receive a pulse it will also send a pulse to warn other nanobots.

Next to this, the nanobots are able to cling to the artery walls and other particles. When attached to other particles they can check if the guest cell is a pathogen or a 'good' cell made by the body. When a cell is determined as malign it will be neutralized.

This simulation of the bloodstream will be implemented in Java. The stream contains white and red bloodcells, platelets, pathogens and nanobots. The elements have slightly different velocities, based on their mass. We've built in the option that pathogens can divide. Furthermore the white bloods cells will produce antibodies when they collide with the pathogens. The research will be structured as a series of experiments aimed at finding the correct individual behaviours for the nanobots to collectively neutralize the bacteria threat.

Our first experiments focus on how to increase the effectivity of fighting the pathogens on individual basis. We look at the effect of zigzagging on the efficiency. In our first experiment we've simulated 1000 cycles of a bloodstream. In this bloodstream 100 bacteria and 41 nanobots were present. The bacteria in this simulation cannot divide. This is done to get more accurate results. We looked at how much bacteria are left after 1000 cycli. Every try we've simulated the same cycle by making all random variables similar for the try with zigzagging and without zigzagging.

In the first phase we let the nanobot zigzag when it collided with an antibody. The idea behind this was that a bigger range was reached so that a pathogen would be found faster. The reason for this was that when an antibody is nearby a nanobot there is also a pathogen nearby. This is because white cells produce antibodies when colliding with a pathogen. As zigzagging costs energy we first implemented that a nanobot is going to zigzag when they collide with an antibody.

In the second phase of our first experiment we let our nanobots zigzag continuously from the beginning. The reason to do this was because no improvements were observed when the nanobots started to zigzag when colliding with an antibody.

Another strategy was implemented for the nanobots to handle situations in which there are multiple kinds of pathogens, each with their own threat level. When a nanobot would detect a pathogen, it would check its threat level and send a pulse to notify the other bots, which would then send a pulse too. Soon every bot would be aware of the highest threat occupying the vein. They would leave the lower-level threats alone and go after the higher-ranked threat first.

In our second experiment we combined the zigzag functionality with the pulses in such a way that, when one nanobot started zigzagging as a result of a collision with an antibody, it would send a pulse and would cause nearby bots to start zigzagging as well when they receive the pulse. These nanobots would then too send a signal, causing every bot in the area to move in that same zigzag pattern.

We hypothesize that several rules of individual behaviour can be defined that will cause the combined mass of nanobots to behave in a way that could be described as intelligent. However, it will be challenging to precisely define individual behaviour in such a way that the group of nanobots will behave in the way that we want them to behave, that is, locating and exterminating bacteria. We will therefore follow a trial and error process of defining individual behaviours.

For our first experiments we thought that zigzagging in both phases would increase the efficiency of finding and destroying antibodies. We thought this would happen because a bigger range is reached in which is searched for pathogens.

We expect an increase in efficiency for fighting the pathogens for our second experiment where we use the pulses as well. We think that using pulses to trigger other nanobots to zigzag will help to find the pathogens faster than when only the nanobots which collide with an antibody will start zigzagging. On the other hand we don't think that using pulses to let other nanobots zigzag will be a faster method to fight pathogens than when the nanobots start to zigzag. We think that this has all to do with the fact that a bigger range is reached in which is searched for pathogens when the nanobots start zigzagging. However zigzagging costs energy because the nanobots should steer.

Results

We made a program that simulates the flow of blood with a lot of its component. In the stream of blood there are also a few bots which show a simple behaviour relevant to the situation. For example they can attach to other entities and check if they are malign or benign. In the case that the entity is malign, the entity should be destroyed. Else it should just let go of the entity and search for other malign entities.

We only included bacteria as malign threats, but this could be extended to other entities, such as viruses, as well The different speeds of the entities along with the random movement made the blood very dynamic as well. The situation of the blood can change very rapidly.

We made the representation of the flow of blood using a two-dimensional matrix filled with references to entities used in the system. Detecting collisions this way is a lot easier than checking our list of all objects in the world. Colliding with an entity will make the one in the back flow around the leading object, since otherwise it can cause a kind of "traffic jam". When initializing our blood vein all entities will be added percentage-wise (for example 45% blood cells) in random places.

In the graphical representation of our implementation we have chosen to make the vein donut-shaped. This makes the blood flow around naturally. We have also chosen to make the walls of the vein impossible to bounce against.

When colliding with the wall, the entity will simply end up on the other side. This makes it seem a lot more realistic, since this will also happen in reality (except three-dimensional). It also fixed a problem where entities would stack on the walls, since movement is random and colliding with the wall didn't change position.

The reason we have used two-dimensional instead of three-dimensional is that it won't be as easy to see what is going on and computationally it's a lot more intensive. There is a counter which keeps track of the entities alive in the blood. Every turn every entity is flowed along with the stream of blood. Every entity is also moved to the left or right. Only bots are able to move by themselves, which they would in reality do with some kind of rudder to steer. When bots destroy an entity the entity is removed from the list of entities.

The results of the first experiments in which the nanobots will start zigzagging when colliding with an antibody and in which the nanobots start zigzagging from the beginning can be found in the tables below.

The first table shows the results of the first phase of the first experiment where the nanobots started zigzagging when colliding with an antibody. Counted are the amount of pathogens left after 1000 cycles. Pulses were not taken into account. The second column is meant to give a kind of base values for the experiments. During these measures no change in behaviour will occur when a nanobot collides with an antibody. In the third column are the amount of pathogens shown when the nanobots will zigzag if they collide with an antibody.

Try	Without zigzag	With zigzag	Amounts each try
1	31	29	cycles: 1000
2	38	34	bots: 41
3	25	36	white blood cells: 20
4	35	34	pathogens: 100
5	25	41	red blood cells: 900
6	35	41	platelets: 900

The table below shows the results of the second phase of the first experiments in which the nanobots start zigzagging from the beginning. Again the amount of pathogens left are counted and pulses are not taken into account.

Try	Without zigzag	With zigzag	Amounts each try
1	30	20	cycles: 1000
2	26	21	bots: 41
3	23	27	white blood cells: 20
4	37	24	pathogens: 100
5	21	11	red blood cells: 900
6	30	25	platelets: 900

This table displays the results of the tests where the nanobots first start zigzagging when colliding with an antibody, and then send pulses to notify other nanobots to do the same.

Try	Without zigzag	With zigzag	Amounts each try
1	30	19	cycles: 1000
2	28	22	bots: 41
3	22	17	white blood cells: 20
4	26	27	pathogens: 100
5	31	24	red blood cells: 900
6	27	13	platelets: 900

Conclusion

From our results shown in the first table above where the nanobots start zigzagging when colliding with an antibody not much can be concluded. There is no improvement when the zigzagging is turned on which would be the case if overall there would be a decrease in the amount of pathogens left after 1000 cycles. Strangely there does not seem to be any connection between the base results and the results when zigzagging is turned on. In some tries there is a big decrease in efficiency of fighting the pathogens and in other tries there is a slight increase in efficiency. From this we can only conclude that zigzagging when colliding with an antibody does not increase the efficiency.

This is quite strange as we expected an increase in eficiency. A possible explanation for this could be that the range in which the nanobot zigzags is too small or too big. Further research is needed in order to check this hypothesis.

The results from the second phase of this experiment (which are shown in the second table) does show an increase in efficiency in fighting the pathogens when comparing to the base results. In every try there is an increase in efficiency. On average there is a decrease in amount of pathogens left of approximately 23 % $(((30+26+23+37+21+30=167)-(20+21+27+24+11+25=128)/167)*100 \approx 23)$.

The final type of experiment tested the pulse funtion. The use of pulses resulted in a rapid and continuous signal being spread throughout the vein, causing every nanobot to start zigzagging soon after the first bot's collision with an antibody, and continuing to do so for the entire simulation, yielding very similar results to the test where the bots were programmed to zigzag non-stop.

The disadvantage of zigzagging continuously is that it costs more energy as the nanobots should steer a lot. This is not very efficient and was the reason that we only wanted to zigzag in specific situations. The pulses did not help to lessen the energy spent on zigzagging, as they caused the command to start zigzagging throughout the entire vein. From this, it can be concluded that the pulses are an effective way to spread a message to all bots, but not only to bots in a specific area.

Discussion

A week is a very short amount of time to make a complex system with a simple artificial intelligence. Nonetheless we think the simulation we have made is a great start. Every so often we discussed what is still left to do and noted some problems. Because of a lot of creative ideas in the group we have also come up with many ways to extend this program (some examples include a battle with two teams of bots with slightly different AI, attaching the robots to the artery walls to slow down movement, and using information provided by antibodies to stimulate a certain behaviour by the bots.)

Acknowledgement

We want to thank Anthony (Toto) van Inge for giving us the idea to work with swarm intelligence.

Appendix

Used software

1. Java compiler 1.6
2. Eclipse Standard 4.4

References

Al-Hudhud, Ghada. (2012). *On Swarming Medical Nanorobots.* International Journal of Bio-Science and Bio-Technology, 4 (1), 75-90.
Blum, Christian & Li, Xiaodong. (2008). *Swarm intelligence in optimization* (pp. 43-44). Springer Berlin Heidelberg.
Dorigo, Marco & Birattari, Mauro. (2007). *Swarm intelligence.* Scholarpedia, 2 (9), 1462.