Practical session: Didabots

Many insect and animal communities solve problems through apparently cooperative behavior. During this practical session you will initially replicate and then extend Maris and te Boekhorst's experiments on robots, called Didabots collective heap building process. The key idea behind the experiments is that seemingly complex patterns of behavior (e.g. heap building) can result from a limited set of simple rules that steer the interactions between different robots and their environment. By varying the sensor placement and movement behavior of the vehicles, try to replicate this emergent behavior (see Figure 1). Note, to replicate this behavior you may also need to change some other parameters of the simulation, such as the mass of the robot, the size of the blocks, etc. After you replicate the paper's results with 12 vs. 25 blocks, and with 1, 3, and 5 robots, you will then do a second experiment to investigate some necessary and sufficient conditions for heap formation. Think carefully about why you think heaps form, and choose a factor to vary (e.g., box mass, box size, sensor placement or sensitivity, etc.). conduct your own experiment by observing the distribution of heaps formed at three levels of that factor. Finally, you will write a scientific report presenting your experiment and findings. The original Maris & te Boekhorst paper can serve as a good example of what to aim for, in terms of style and content.

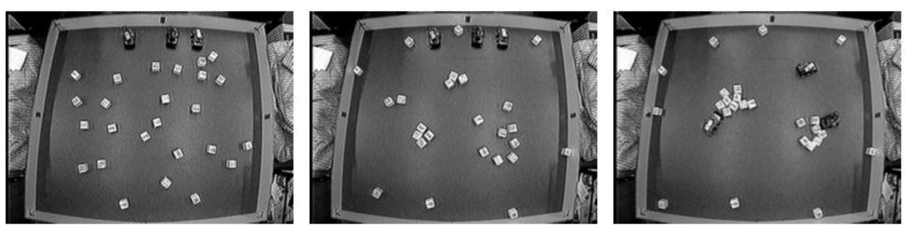

Figure 1: Example of heap building by Didabots (taken from Hafner, 2014). Initially the cubes are randomly distributed. Over time, a number of clusters form. By the end, only two clusters remain along with a number of cubes along the walls of the arena.

Assignment

First replicate the experiment in Maris and te Boekhorst (1996), finding the sensor position and movement behavior needed to produce heaping behavior similar to theirs with varying numbers of blocks (12 vs. 25) and different numbers of agents (1 vs. 3 vs. 5). Write this replication up, as well as the way you set up the vehicle to accomplish this (specify any parameters you changed). You are allowed to modify most of the parameters available in the simulation, this includes the block mass, box size, friction, etc. But do not modify the number of simulation steps or the size of the arena.

For your second experiment, think carefully about how different factors create sufficient/necessary conditions for heap formation, and choose one factor you believe (and argue in your writing) is important. Test that factor (e.g., block mass, friction, sensor position, etc.) at three levels, with at least 10 simulations per level, to see whether you reproduce the heaping behavior. Remember to keep the other simulation factors constant. Take screenshots of the final state of each simulation, and record and graph the distribution of heaps formed (see Data Collection).

Data Collection

Information on: You can run multiple simulations simultaneously with multiple browser windows (next to each other; not only in a tab!) For each simulation allow 20,000 steps to be completed. Screenshot the final state of simulation, and tally clusters from these: proportion of blocks in heaps and number of clusters of each size. Run at least 10 simulations per condition. Graph the distributions, and provide means per condition. No need for significance testing, although you could run a simple Kolmogorov-Smirnov test (in R: ks.test(); http://www.physics.csbsju.edu/stats/KS-test.html).

Report (Deadline: October 26th, 23:59)

- Write a scientific paper reporting the results of your replication (Experiment 1), and of your investigation of your chosen factor (Experiment 2).
- Reasoning of your vehicles's design (include a figure showing sensor placement and sensitivity).
- Report the replication results: Did you exactly reproduce the distributions they found? If not, any hypothesis why not?
- Report the factor you chose to examine further for its role in heap formation, and why you chose it (i.e., why you think it's important: what's your theory?).
- From your measurements, make similar graphs like Maris and te Boekhorst's (1996) Figures 5 and 6. Discuss similarities and differences their results.
    1. Frequency distributions of the number of clusters (a), the percentage of cubes in heaps (b) and cluster size (c). Helpful R code: http://www.cookbook-r.com/Graphs/Plotting_distributions_(ggplot2)/ (Similar can be done in python with matplotlib.)
    2. Dependency of final number of clusters on the number of robots for experiments with 12 (a) and 25 (b) obstacles.
- General tips on writing a good report can be found here: http://kachergis.com/projects/final_project.html

The overall goal is to figure out to the best of your ability (and time) why heap formation happens. The report should convince us that you thought about this, came up with a good idea to test, tested it thoroughly, and understand the implications of your results.

Deliverables

With your report please include the following in a single zip file. * Screenshots of one exemplary simulation result from each of your conditions. * All code files, with one clearly-labeled HTML file per condition that can simply be run to see one simulation from that condition.

Notes:

1. A "heap" (i.e. cluster of blocks) is understood to be a set of at least 3 blocks where the distance between them is smaller than the length of one block. Blocks at the walls are not part of a heap.
2. Only use (realistic) information that is provided by (noisy) sensors. If you create a sensor, make sure it is somewhat physically realistic, and randomly will give noisy information.
3. For movement, only use functions that apply force or torque to the robot. Except for initialization, never directly set its position or velocity.
4. For convenience the simulation code provided has a "robotMove" function, feel free to modify this function to move your robot. We have also included a "debugSensors" flag to allow visualization of sensor placement, when true the current sensor positions and their range is displayed.

Further explanation: While simulations offer a lot of flexibility, we must ensure that we don't fool ourselves by programming in ways that are physically impossible, or use knowledge that cannot be known. In the simulations, everything must be known and computed, so you have much more information available than if you program a real robot. Importantly, we can't just use any information that is available within the simulation, at least not directly. For example, a real robot cannot know all velocities, positions, masses, friction coefficients, etc. from every object around it. In the real world, all information comes from sensors, and thus we stick to this principle in simulation as well. Similarly, all a motor can do is to create force or torque, thus we cannot simply set the position or velocity of the simulated robot either.

References

1. Verena Hafner, "An example for (reactive) cooperative behavior: the swiss robots." (didabots.pdf)
2. Maris, M., & te Boekhorst, R., Exploiting physical constraints: heap formation through behavior error in a group of robots," in ed. M. Asada,

Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996, 1655-1660.
3. Michael R. W. Dawson, Brian Dupuis, and Michael Wilson, “Embodiment, Stigmergy, and Swarm Intelligence,” From Bricks to Brains (Edmonton, Alberta: Athabasca University Press, 2010), 226-237.