Experiments in swarm robotics

Andrea Roli
andrea.roli@unibo.it
Dept. of Computer Science and Engineering (DISI)
Alma Mater Studiorum Università di Bologna

A.Y. 2018/2019

Aggregation task

A typical task in SR is *aggregation*, i.e. the swarm behaviour consisting in gathering in one place. A typical method for aggregation in SR comes from the observation of social insects, in particular coackroaches. An informal model for aggregation is described as follows:

- Each robot performs random walk (possibly with collision avoidance[1]) and stops with probability P_s;
- P_s depends on the number of nearby robots already stopped;
- the higher this number, the higher P_s (positive feedback);
- when a robot has stopped in a group, it can leave it with a probability P_w and start again to wander (negative feedback);
- the higher the number of stopped nearby robots, the lower P_w.

Let $S \in [0, 1]$ be the spontaneous stopping probability, $W \in [0, 1]$ the spontaneous walking probability and N the number of nearby stopped robots. These probabilities can be updated as follows:

- $P_s = \min\{P_s^{max}, S + \alpha N\}$
- $P_w = \max\{P_w^{min}, W - \beta N\}$

[1] It is easier to start without a collision avoidance behaviour and then introduce it.

where S, W, α, β, P_s^{max} and P_w^{min} are parameters.[2]

Robots can perceive their nearby companions by means of a *range and bearing communication system*, which usually returns angle and distance—and a short message—of surrounding robots. In ARGoS you can use the RAB mounted on the footbots.

Exercise 1

Implement this behavioural scheme in ARGoS. Experiment with S, W, α and β. The behaviour can be modelled by means of a probabilistic automaton. You may want to use the following pieces of code.

```
-- send something to signal the presence to the other robots
robot.range_and_bearing.set_data(1,1)
```

meaning: on byte 1 (first argument) send value 1 (second argument).

```
-- send something to signal to the other robots that you are moving
robot.range_and_bearing.set_data(1,0)


-- Count the number of robots sensed close to the robot
function CountRAB()
  number_robot_sensed = 0
  for i = 1, #robot.range_and_bearing do -- for each robot seen
    -- see if they are close enough.
    -- What happens with different cutoff values?
    if robot.range_and_bearing[i].range < 30 and
                     robot.range_and_bearing[i].data[1]==1 then
      number_robot_sensed = number_robot_sensed + 1
    end
  end
end


 Ps = math.min(1,S+alpha*N)
 Pw = math.max(0,W-beta*N)
```

[2]These parameters in general depend upon the number of robots, the size of the arena and the maximal range used for the range-and-bearing. For a rectangular arena of 4×5 meters, 30 robots and a maximal range of 30cm, you may initially try with: $W = 0.1$, $S = 0.15$, $P_s^{max} = 0.99$, $P_w^{min} = 0.01$, $\alpha = 0.1$, $\beta = 0.05$

Exercise 2: Aggregation on a black spot

To build an aggregate on a dark spot we may change the probability update rules as follows:

- $P_s = \min\{1, S + \alpha N + D_s\}$
- $P_w = \max\{0, W - \beta N - D_w\}$

where D_s and D_w are the probabilities of stopping (resp. leaving) on the black spot.

Exercise 3: Aggregation on one of two black spots

In nature, if there are many places in which insects can aggregate, they collectively choose only one. Try the behaviour designed in the previous exercise in an arena with two black areas and observe the behaviour.

Food for thought

- Is the behaviour robust w.r.t. parameter values?
- What happens if robots have different probabilities P_s, P_w?
- What are the advantages and disadvantages of the collective choice?
- How would you bias the aggregation on a specific area of the arena?