Swarm Intelligence Course (INFO-H-414)
Exams

July 9, 2015

1 Modalities

The exam is divided in two parts:

Project You are asked to provide the required deliverables and to present your project in a 7-minute talk, followed by 5 minutes of questions. This will account for up to 10 points of your final grade.

Questions You will be asked a number of questions concerning *the entire course material*. This will account for up to 10 points of your final grade.

To pass the exam one must collect at least 5 points for each part of the exam.

1.1 Timeline

- The date of the exam is September 4th, 2015.
- The project submission deadline is August 21st, 2015 at 23.59.
- Each day of delay on the submission will entail a penalty of 2 points on the final evaluation of the project.

1.2 Project Deliverables

For the project, you will have to provide:

- Your code in digital format (i.e., text files), so we can test it. Send it by e-mail to the people responsible for the project (see contacts at the end of the document);
- A short document (6-8 pages) written in English that describes your work. You have to explain both the idea and the implementation of your solution, and present the results you obtained.
- On the day of the oral presentation, you are expected to show slides and come with your own laptop. If you happen to not have a laptop, send us a PDF version of your slides **before** the day of the exam.

2 Projects

2.1 General Remarks

Apply what you learnt. It is mostly important that you stick to the principles of swarm intelligence: simple, local interactions, no global communication, no global information.

Avoid complexity. If your solution gets too complex, it is because it is the wrong one.

Honesty pays off. If you find the solution in a paper or in a website, cite it and say why you used that idea and how (it is not a negative thing implementing solution existing in the literature).

Cooperation is forbidden. Always remember that this is an **individual** work.

Analyse your solution. Provide a good analysis of the results of your solution (generate data, plot graphs, report statistical measures). Do not limit your project to the implementation of a nice solution. The result analysis is as important as the code implementation.

The project counts for 50% of your final grade. The basic precondition for you to succeed in the project is that it must work. If it does not, the project won't be considered sufficient. In addition, code must be understandable — comments are mandatory.

The document is very important too. We will evaluate the quality of your text, its clarity, its completeness and its soundness. References to existing methods are considered a plus — honesty *does* pay off! More specifically, the document is good if it contains all the information needed to reproduce your work without having seen the code and a good and complete analysis of the results.

The oral presentation is also very important. In contrast to the document, a good talk deals with *ideas* and leaves the technical details out. Be sure that it fits in the 7-minute slot.

2.2 Swarm Robotics Project: *Collective Decision Making with Simple Agents*

Decision making is a cognitive capability which allows individuals to select the best option among a set of possible alternatives. When a decision is taken by a group of agents, the process is defined as collective decision making.

Collective decision making is a fundamental cognitive capability that can be found in several natural systems. For instance, each spring, honeybee colonies collectively tackle as a decision problem the nest site selection. A different example is the collective decision of bird flocks, which collectively move in one direction without need of any leader.

In this project, the student is asked to provide a robot swarm with the capability of discriminating between four options and collectively select the best one. The robot swarm is composed of simple robots capable to individually estimate each room quality, to communicate with neighbours through light signals and to memorise the quality of only the last visited room. Despite the individual robot simplicity, the robot swarm, as a whole, must decide for the best option. Therefore, robots must collaborate to agree on the selection of the best alternative.

2.2.1 Problem definition

The robot swarm operates in a scenario composed of 4 *target* rooms r_i (with $i \in [0,3]$) connected by a central room. Each target room has a quality value v_i (with $i \in [0,3]$), which is computed as the average of two metrics:

$$v = (v_G + v_O)/2 \tag{1}$$

The two metrics are computed as function of the following environmental characteristics:

- **The ground color (v_G)**: the ground color varies in the full gray scale from white to black and determines the value $v_G \in [0,1]$. We assume that $v_G = 1$ for white grounds, $v_G = 0$ for black grounds and in the between v_G scales linearly. The robots can perceive the ground color through the ground sensor.

- **The number of objects (v_O)**: each room stores a number of objects $\mathcal{O}_i \in [2,12]$ which determines $v_O \in [0,1]$. We assume that v_O is linearly proportional to the number of objects $\mathcal{O}$, i.e., the more the better. Each object is marked with a green LED, which can be perceived by the robots through the camera.

Goal The robot swarm has to select the room r_i with best quality v_i:

$$r_i,\ i \in \underset{i \in [0,3]}{\arg\max}\, v_i \tag{2}$$

To select a room a robot must physically move into the selected room. We assume that the swarm has done a collective decision when the quorum $Q = 0.9$ of the total swarm moved into a room r. For instance, for a swarm of 20 robots, the decision is considered as taken when at least 18 robots have moved into the same target room.

Simplicity of the robots The swarm is composed of 20 simple robots with a minimalistic design. Each robot is equipped with:

- **wheels**: to move in the environment;

- **LEDs**: to communicate with the neighbour robots;

- **Omnidirectional camera**: to detect LEDs, that may be useful to (i) identify the rooms, (ii) count the objects in a room, and (iii) read the LED color of the neighbour robots.
- **ground sensor**: to read the color of the ground.
- **limited memory**: to memorize only the quality of the last visited room. Note that this constraint must not be violated in order to obtain a sufficient solution. Each robot that estimates the quality of a room must overwrite the previous quality estimates (if any) that it stored.

Additional Remarks

- The environment is designed with a central room connecting the 4 target rooms, so that the robots can encounter in the central room and interact with each other.
- Robots are able to differentiate the four target rooms by mean of LED lights of different colours that are placed on the door of each target room. The colours are assigned as follows: magenta, blue, orange, and red for rooms 0,1,2,3 respectively. The robots can also use these lights to localise themselves in the environment.

Figure 1 shows a top view of the environment composed of four target rooms and a central room. The central room has no light and black ground. Each room has a different quality which may vary in each experiment and is determined by the composition of the two metrics v_G and v_O (see Equation (1)). The quality of the four target rooms of the example illustrated in Figure 1 are given in the Table 1. The best is room is room number 0 with the evaluation: $v_0 = 0.635158$.

Room i	v_G	v_O (O)	**Room quality** v_i
Room 0	0.970316	0.3 (5)	0.635158
Room 1	0.183674	0.6 (8)	0.391837
Room 2	0.621518	0.0 (2)	0.310759
Room 3	0.205702	0.0 (2)	0.102851

Table 1: Target-room's quality values of the example depicted in Fig 1.

2.2.2 Goals

The goal of this project is to design, implement and test a robot controller that allows the swarm to collectively select the best room r_i between 4 possible target rooms. The best room has the the highest quality v_i.

Different solutions are possible. Bonus points are awarded for simple and reactive controllers accompanied by an exhaustive result analysis. Remember to follow the principles of swarm robotics and apply what you learned during the

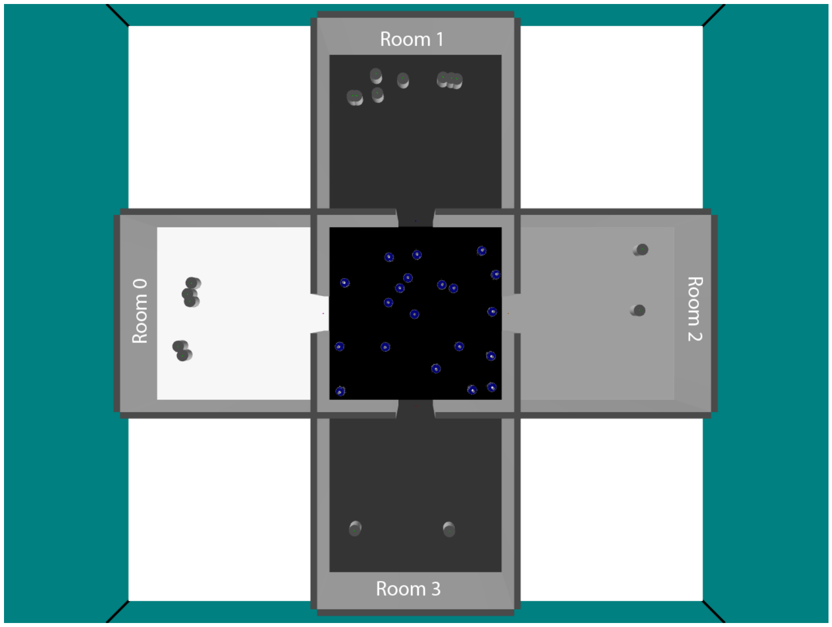

Figure 1: A top view of the environment.

course. Solutions in which the robots aggregate in a database global information about every room will be considered poor.

The student is required to analyse the solution at first with the provided configuration, and after by varying the swarm size N in the range of values $\{30, 40, 50\}$ robots. We are interested to know how the performance of your systems vary as a function of N. A complete analysis must be performed to evaluate the quality of the behaviour. In particular, quantitative numerical measures of the performance of the system must be produced. ARGoS automatically dumps data on a file whose name must be set by you in the XML experiment configuration. This file is composed of several lines, each line containing five elements:

- The current step
- The number of robots in room 0
- The number of robots in room 1
- The number of robots in room 2
- The number of robots in room 3

To present statistically meaningful results we suggest that you repeat your experiments at least 30 times.

Be aware that, for the project evaluation, **the analysis is as important as the implementation**. A project presenting a behaviour that is capable of selecting the optimal room wonderfully will be evaluated poorly if the analysis part is missed or limited.

Technical information on ARGoS, Lua, and the experiment configuration file is reported in the swarm robotics page of the course: http://iridia.ulb.ac.be/~lgarattoni/extra/h-414/.

Setting up the code

- Download the experiment files: SR_Project_H414.tar from http://iridia.ulb.ac.be/~lgarattoni/extra/h-414/SR_Project_H414.tar.
- Unpack the archive and compile the code:
  - $ tar xvf SR_Project_H414.tar # Unpacking
  - $ cd SR_Project_H414 # Enter the directory
  - $ mkdir build # Creating build dir
  - $ cd build # Entering build dir
  - $ cmake -DCMAKE_BUILD_TYPE=Release ../src # Configuring the build dir
  - $ make # Compiling the code
- Set the environment variable ARGOS_PLUGIN_PATH to the full path in which the build/ directory is located:

  $ export ARGOS_PLUGIN_PATH=/path/to/SR_Project_H414/build/
- You can also put this line into your $HOME/.bashrc file, so it will be automatically executed every time you open a console. Run the experiment to check that everything is OK:
  - $ cd /path/to/SR_Project_H414 # Make sure you are in the right directory
  - $ argos3 -c decision-making.xml # Run the experiment

If the usual ARGoS GUI appears, you're ready to go.

2.2.3 Deliverables

The final deliverables must include source code and documentation:

Code: The Lua scripts that you developed, well-commented and well-structured.

Documentation: A report of 6-8 pages structured as follows:

– Main idea of your approach.

– Structure of your solution (the state machine).

– Analysis of the results.

3 Contacts

Marco Dorigo	mdorigo@ulb.ac.be	for general questions
Mauro Birattari	mbiro@ulb.ac.be	for general questions
Andreagiovanni Reina	areina@ulb.ac.be	for the project
Lorenzo Garattoni	lgaratto@ulb.ac.be	for the project