Alma Mater Studiorum - Università di Bologna
Campus di Cesena

DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA E SCIENZE INFORMATICHE

An information theory analysis of critical Boolean networks as control software for robots

TESI IN
SISTEMI INTELLIGENTI ROBOTICI

Relatore:
Prof. ANDREA ROLI

Co-relatore:
Dott. MICHELE BRACCINI

Presentata da:
Dott. MATTEO MAGNINI

Anno accademico
2019 - 2020

In arduis servare mentem

(Aircraft carrier Cavour)

Abstract

This work is an analysis of critical random Boolean networks used as control software for robots. The main goal is to find if there are relations between information theory measures on robot's sensors and actuators and the capability of the robot to achieve a particular task. Secondary goals are to verify if just the number of nodes of the networks is significant to obtain better populations of controllers for a given task and if a Boolean network can perform well in more than one single task. Results show that for certain tasks there is a strongly positively correlation between some information theory measures and the objective function of the task. Moreover Boolean networks with an higher number of nodes tend to perform better. These results can be useful in the **automatic design** process of control software for robots. Finally some Boolean networks from a random generated population exhibit **phenotypic plasticity**, which is the ability to manifest more phenotypes from the same genotype in different environments. In this scenario it is the capability of the same Boolean network (same functions and connections) to successfully achieve different tasks.

Contents

List of Figures

List of Tables

Introduction

This work belongs to the field of Robotics, in particular Intelligent Robotic Systems. A robot is a complete physical autonomous agent able to perform a task in a real world environment without the human intervention. The work concerns also Complex Systems Science by using particular types of Boolean networks as control software for robot. Ultimately this work regards information theory by using metrics to measure information properties over the robot controller.

The reason for this work is not only to expand the scientific knowledge on information theory and Boolean networks but also to give more expertise in the **automatic design** of control software for robots. Automatic design of controllers holds a crucial role in robotics and Boolean networks can be successfully used in the process as control software [23]. The main objective of this work is to investigate if some measures of information theory could be used as a not handcrafted merit factor for automatic design in specific tasks.

This thesis analyzes the results of robots controlled by critical random Boolean networks in predefined tasks. Three common tasks in robotics are chosen: obstacle avoidance, path following and phototaxis. Several experiments are executed with different network configurations through a simulation software. Results are collected in textual files for analysis. The goal is to find if there are relations between information theory metrics over robot's sensors and actuators and handwritten task objective function. Another objective is to verify if there are differences in networks populations with respect to robot's ability in achieving a task. Finally controllers are switched in the other two tasks they did not see. The goal is to test if the same critical random Boolean network is capable of performing well more than one task.

The thesis is structured as follow. Chapter 1 introduces information theory measures used in the analysis. Chapter 2 describes Boolean networks with focus on their possible topologies and on critical Boolean network. Chapter 3 illustrates the simulation software use for the experiments along with the robot and its characteristics. Chapter 4 defines the tasks, in particular the arena, duration and objective function. Finally in chapter 5 the results of the experiments are reported along with their analysis.

Chapter 1

Information Theory

This chapter is an overview of the main concepts of information theory. In particular in section 1.1 some information theory measures are formally described. These measures are used in the experiments (see chapter 4) to evaluate Boolean networks, results are reported in chapter 5. An estimation of the computational cost needed to compute the metrics introduced is provided in section 1.2.

1.1 Measures

Information theory is a pervasive science whose subject is information related phenomena and information itself. The field of information theory spread in 1948, when C. Shannon published a paper [25] where he defined a mathematical notion by which information could be quantified. Information theory has found applications in several fields like cryptography [1], data compression [13], neuroscience [6], bioinformatics [26], robotics [3] [10], etc. In the literature many information measures have been proposed. Here we introduce the measures that we use in the experiments analysis 5.

1.1.1 Entropy

Entropy (H) is the measure that quantifies the amount of information carried by a source. A source can be considered as an aleatory variable related to some sort of event.

An event that is certain has entropy zero. An event whose outcomes are all equally probable, the maximum of uncertainty, has the highest entropy.

The discrete entropy's formula introduced by C. Shannon [25] is:

$$H(X) = -\sum_{x} p(x)\log_2 p(x) \tag{1.1}$$

where X is an aleatory variable, x is one of its finite possible value, $p(x)$ is the probability of X to be equal to x, an abbreviation for $p(X = x)$. The curve equation for a Boolean variable is plotted in figure 1.1.

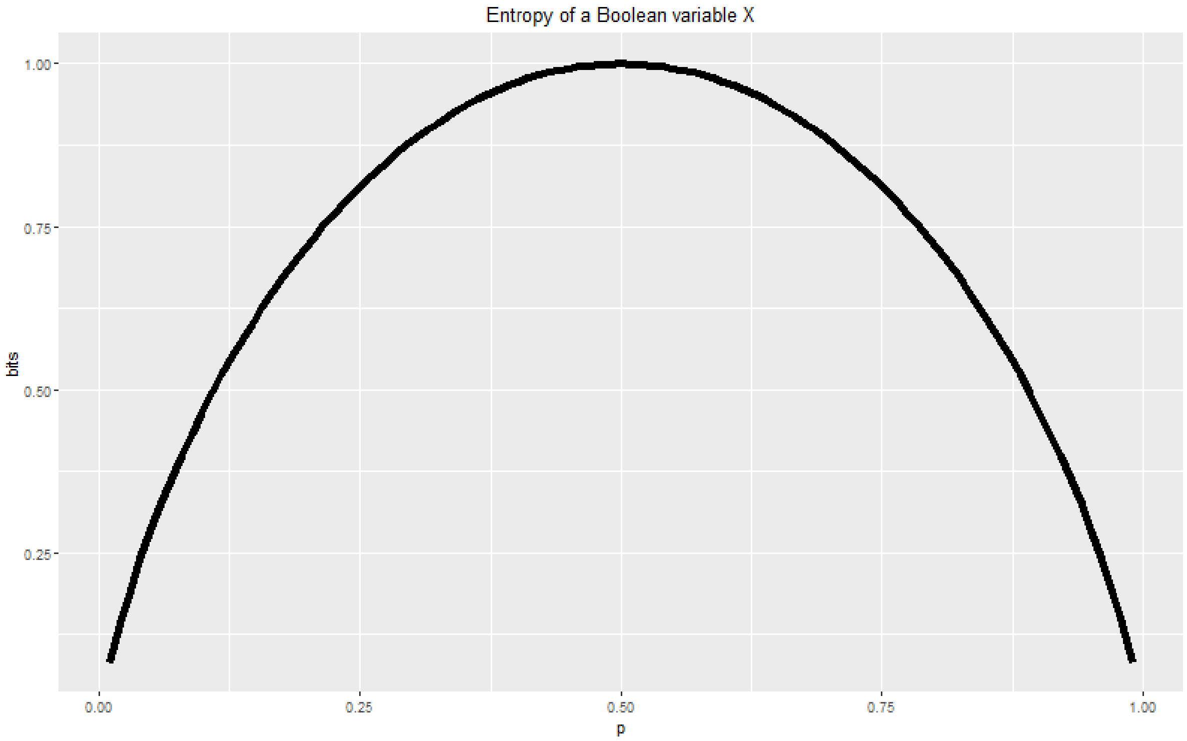


Figure 1.1: entropy 1.1 curve shape in the case of an aleatory Boolean variable, for example a coin toss where head has probability p, with $p \in [0, 1]$, and tail $1 - p$.

The choice of the base of the logarithm is arbitrary. For now on we always consider the base two. The unit of measure using the base two logarithm is called bit(s). The lower bound of H is zero, while the upper bound is:

$$Max(H(X)) = \log_2 |X| \tag{1.2}$$

1.1.2 Mutual Information

Mutual Information (MI) quantifies the amount of information shared by two aleatory variables. When the variables are **independent** MI is zero, while when they are fully **dependent** MI is maximized.

The discrete mutual information's formula [25] is:

$$MI(X,Y) = \sum_{x,y} p(x,y) * \log_2 \frac{p(x,y)}{p(x) * p(y)} \tag{1.3}$$

where X and Y are two aleatory variables, x and y are possible values of the variables and $p(x,y)$ is the joint probability of the two variables, an abbreviation for $p(X = x, Y = y)$. The lower bound of MI is zero, while the upper bound is the minimum entropy of the two variables:

$$Max(MI(X,Y)) = Min(H(X), H(Y)) \tag{1.4}$$

Proof:
Mutual Information can be rewritten as

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{1.5}$$

so, by contradiction

$$\begin{aligned} MI(X,Y) &> H(X) \\ H(X) + H(Y) - H(X,Y) &> H(X) \\ H(Y) &> H(X,Y) \\ \text{but } H(Y) &<= H(X,Y)\ \forall X,Y \\ \text{so } MI(X,Y) &<= H(X) \end{aligned} \tag{1.6}$$

and for symmetry $MI(X,Y) <= H(Y)$.

In figure 1.2 are summarized the main properties for functions H and MI (I in the figure) with different relations between the aleatory variables.

1.1.3 Predictive Information

“Observations on the past provide some hints about what will happen in the future, and this can be quantified using information theory’ ’[4]. W. Bialek and N. Tishby

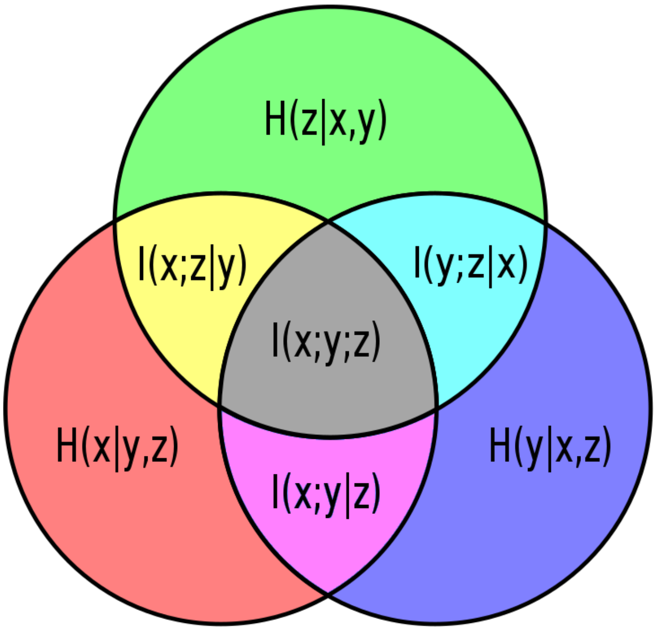


Figure 1.2: diagram showing the properties for functions entropy (H) and mutual information (I). Red circle represents $H(x)$, blue circle is $H(y)$, green circle is $H(z)$. Overlapping circles means partial dependency between the involved aleatory variables. Figure taken from [27].

introduced in 1999 a measure to quantify information sharing across time between two variables. We call predictive information (PI) the quantity of shared information between variable X and Y (time series) considering the latter at time $t + c$, c is a positive constant that we consider always 1 in the experiments. With $c = 1$ we fall into Markov processes which hypothesize that the past information contained in the previous step holds the entire information needed. The discrete predictive information's formula is MI considering the second variable one step ahead in the future:

$$PI(X,Y) = MI(X_t, Y_{t+1}) = \sum_{y_{t+1},x_t} p(x_t, y_{t+1}) * \log_2 \frac{p(x_t, y_{t+1})}{p(x_t) * p(y_{t+1})} \tag{1.7}$$

This function can be easily applied in many ways for measuring the complexity of a system. In [3] PI is computed on a robot's sensor values at time t and on sensor values at time $t+1$. Researchers have found that for the particular task and environment that they were analysing PI is high when the robot has a explorative behaviour and predictable future sensor values. PI is proposed as a candidate objective function for similar scenario. Instead, in [10], PI is computed on sensor values at time t and on actuator values at time $t+1$. PI is used, along with other measures of complexity, to evaluate robots during an evolution process in the achievement of a given task.

1.1.4 Transfer Entropy

Intuitively, transfer entropy (TE) is the amount of **directed**, time asymmetric, information exchanged over time from an aleatory variable X towards another one Y. Introduced by T. Schreiber[24], the most general discrete formula is:

$$TE_{X \to Y} = TE(X,Y) = \sum_{y_{t+1}, y_t^k, x_t^l} p(y_{t+1}, y_t^k, x_t^l) * \log_2 \frac{p(y_{t+1}|y_t^k, x_t^l)}{p(y_{t+1}|y_t^k)} \tag{1.8}$$

where $p(y_{t+1}|y_t^k, x_t^l)$ is the conditional probability of y_{t+1} given y_t^k, x_t^l, similarly for the other conditional probability. The apexes k and l indicates the amount of delay to consider: y_t^k is an abbreviation for $(y_t, ..., y_{t-k+1})$. For ease of computational power we always consider $k = l = 1$. Like for the choice of $c = 1$ in PI, choosing $k = l = 1$ is equivalent to consider the entire past and the entire future for Markov processes.

Transfer entropy can be rewritten in function of simpler measures:

$$TE(X,Y) = H(Y_{t+1}|Y_t) - H(Y_{t+1}|Y_t, X_t) \tag{1.9}$$

It is straightforward to see that the lower bound is zero, while the upper bound is the minimum of the entropies of X and Y.

$$Max(TE(X,Y)) = H(X) \tag{1.10}$$

Proof:

$$\begin{aligned} H(Y_{t+1}|Y_t) &= H(Y_{t+1}, Y_t) - H(Y_t) \\ H(Y_{t+1}|Y_t, X_t) &= H(Y_{t+1}, Y_t, X_t) - H(Y_t, X_t) \\ TE(X,Y) &= H(Y_{t+1}, Y_t) + H(Y_t, X_t) - H(Y_t) - H(Y_{t+1}, Y_t, X_t) \\ MI(Y_{y+1}, X_t|Y_t) &= H(Y_{t+1}, Y_t) + H(Y_t, X_t) - H(Y_t) - H(Y_{t+1}, Y_t, X_t) \end{aligned} \tag{1.11}$$

$$0 <= MI(Y_{y+1}, X_t|Y_t) <= Min(H(X_t), H(Y_{t+1}))$$

so $Max(TE(X,Y)) = Min(H(X), H(Y))$.

1.2 Computational cost

To calculate the previous measures one must known the required probabilities. It is worth to spend some words about the computational cost of this operation. Indeed, if we want to compute the exact value of a measure previously introduced this could result in a time expensive calculus.

Lets suppose to have a file with the values of two time series, named X and Y. A row contains the value of a time step for X and Y, L is the file length.

- $H(X)$. We have to compute all the probabilities for each value of X then the summation in X. So the cost is $L|X| + |X| = O(L|X|)$;
- $MI(X,Y)$ or $PI(X,Y)$. We have to compute all the probabilities for each value of X, Y and the joint probability. Then we have to compute the two summations in X and Y. Cost is $L(|X| + |Y| + |X||Y|) + |X||Y| = O(L|X||Y|)$;
- $TE(X,Y)$. It can be rewritten as

  $$TE(X,Y) = \sum_{y_{t+1},y_t,x_t} p(y_{t+1}, y_t, x_t) * \log_2 \frac{p(y_{t+1}, y_t, x_t) * p(y_t)}{p(y_t, x_t) * p(y_{t+1}, y_t)} \tag{1.12}$$

  We have to compute all the probabilities for each value of X_t, Y_{t+1} and Y_t, however the latter two have the same ones. Then the joint probabilities for $< Y_t, X_t >$, $< Y_{t+1}, Y_t >$, and $< Y_{t+1}, Y_t, X_t >$. In the end the three summations in X_t, Y_t and Y_{t+1}. Cost is $L(|X| + |Y| + |X||Y| + |Y|^2 + |X||Y|^2) + |X||Y|^2 = O(L|X||Y|^2)$.

Cost for TE is linear in the file length and in the number of results of variable X. It is polynomial in the number of results of variable Y. This turns out to be the most expensive measure to compute. If the time cost is too expensive or in the case of continuous aleatory variables, one can use approximation techniques to estimate the probability distributions like KDE [19].

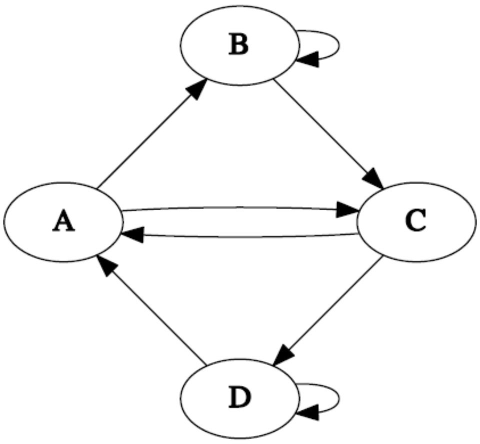
Chapter 2

Boolean Networks

In this chapter we introduce the concept of Boolean networks and the more common models, section 2.1, with a focus on critical Boolean networks. In section 2.2 a description is presented of how Boolean networks are used in experiments (4) as control software for robots.

2.1 Definition

A Boolean network is a discrete time and discrete state dynamical system. It can be instantiated as a directed graph where nodes hold both a Boolean value, 0 or 1, and a Boolean function, an example is represented in figure 2.1. The state of the network is the ordered set of its nodes' values. So, if the number of nodes is N, the number of all possible states of the Boolean network is 2^N. Vertices between nodes are oriented. A vertex carries the Boolean value of the starting node towards the ending node. A node computes its Boolean function based on the values of the incoming vertices and updates its Boolean value. The update scheme used in this work is always synchronous and deterministic. A Boolean function can be represented as a truth table and the number of possible functions is based on the number of incoming vertices. If k_i is the number of incoming vertices for node n_i, then n_i has a Boolean function in a set of $2^{2^{k_i}}$ functions. A random Boolean network (RBN) is a Boolean network where vertices and Boolean functions are chosen randomly.



C	D	A
0	0	0
0	1	1
1	0	1
1	1	1

A	B	B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

C	D	D
0	0	0
0	1	1
1	0	0
1	1	0

Figure 2.1: on the left a graph representing a Boolean network, on the rights the nodes' Boolean functions as truth tables. Image taken from [22].

A Boolean network could have **attractors**. An attractor is a subset of the state space towards which a dynamical system evolves over time. The attractor can be a fixed point, one state, or a cycle, more states. Once a Boolean network reaches an attractor, if its update scheme is deterministic, it cannot reach states that are not in the set of the attractor. Dynamical systems can be analyzed using attractors' characteristics like length, basin of attractor, etc. The basin of attractor is the set of states from which the system reach the attractor for $t \to +\infty$. The state space is divided into basins based on the amount of attractors.

Different strategies of connecting nodes rise to different network topologies, so different features:

- random graphs, this is the first topology and it is used as a benchmark for the other ones. Random graphs are generated through a random process, for example adding vertices to the graphs by randomly selecting two different nodes with an uniform distribution. The node degree distribution is Poissonian or Gaussian. The topology used for the experiments is a critical random Boolean network, also known as N-K model, described in section 2.1.1;

- grid-like networks, this is a very high ordered topology. Each node is connected with a predefined number of neighbour, generally few units. Real examples are computers grids and low-voltage networks;

- scale-free networks, qualitative this topology has few nodes with many connections and the majority of nodes with few connections. More formally the node degree distribution follows a negative power low. These networks are robust to accidental damage, but vulnerable to specific attacks. They are also related to small world phenomena. This topology can represent several real systems such as social network relations, web pages, Internet, etc;

- small world networks, these networks have low mean path length between nodes and high clustering. They are in balance between random networks and grid networks.

Scale-free topology Boolean networks have been proved to model well real systems such as genetic networks. Random Boolean networks, like the N-K model, can still reach discrete results but definitely they need to be fine-tuned[2]. Also Boolean networks with small-world topology have reached interesting results, they "have a propensity to combine comparably large information storage and transfer capacity"[16].

Besides the topology, Boolean networks can be distinguished by their updating scheme. Generally there are two main different updating scheme: **synchronous** and **asynchronous**. In the synchronous update scheme all nodes are updated at the same time. It is always deterministic, a Boolean network in state s_t updates towards the same state s_{t+1}. Instead, in the asynchronous update scheme nodes are updated one at a time. The single node update is still deterministic, however the whole network update can be random.

Updating schemes based on C. Gershenson's taxonomy[11]:

- CRBNs, classic random Boolean networks were proposed by S. Kauffman[14] in 1969. The updating is synchronous and deterministic. CRBNs have attractors. Due to the determinism, once a CRBN reaches an attractor it can only have states belonging to the attractor (except for external perturbations);

- ARBNs, asynchronous random Boolean networks have similar characteristics of

CRBNs but the updating is asynchronous and random. At each time step a node is randomly selected and then updated, so they are not deterministic;

- DARBNs, deterministic asynchronous random Boolean network are like ARBNs but the choice of the node to update is not random. A node is selected following a rule based on a parameter that specify its update period. If more than one node have to update, then they are updated asynchronously following their order;
- GARBNs, generalized asynchronous random Boolean networks are like ARBNs but can update any number of node at each time step;
- DGARBNs, generalized asynchronous random Boolean networks are like DARBNs but if more than one node have to update, then they are updated synchronously.

2.1.1 Critical Boolean Networks

The number of incoming vertices for the node n_i is expressed by parameter k_i. If all nodes have the same value for k_i, the network is homogeneous and it has a global parameter **K** equal to that value. The Boolean function for a node n_i is a truth table with 2^K rows where for each row the output is chosen randomly with probability p_i to be set to 1. If all nodes have the same value for p_i, the network has a global parameter **p** equal to that value.

Networks follow the N-K model introduced by S. Kauffman[15] when they have global parameter K, connections are randomly chosen with uniform probability and nodes have global probability p to be active (value 1). A system in a **critical** regime is said to be in between "order" and "chaos"[18]. If a Boolean network has a stable behaviour its dynamics are frozen, perturbations die out in short time. If it has chaotic behaviour attractors' length is long, almost 2^N, and it is heavily affected by small perturbations. A Boolean network is critical when the attractors' length is short, low degree polynomial of N. This kind of network is robust to small perturbations. Parameters K and p in critical Boolean networks follow the relation[9]:

$$K_c = \frac{1}{2 * p_c(1 - p_c)} \tag{2.1}$$

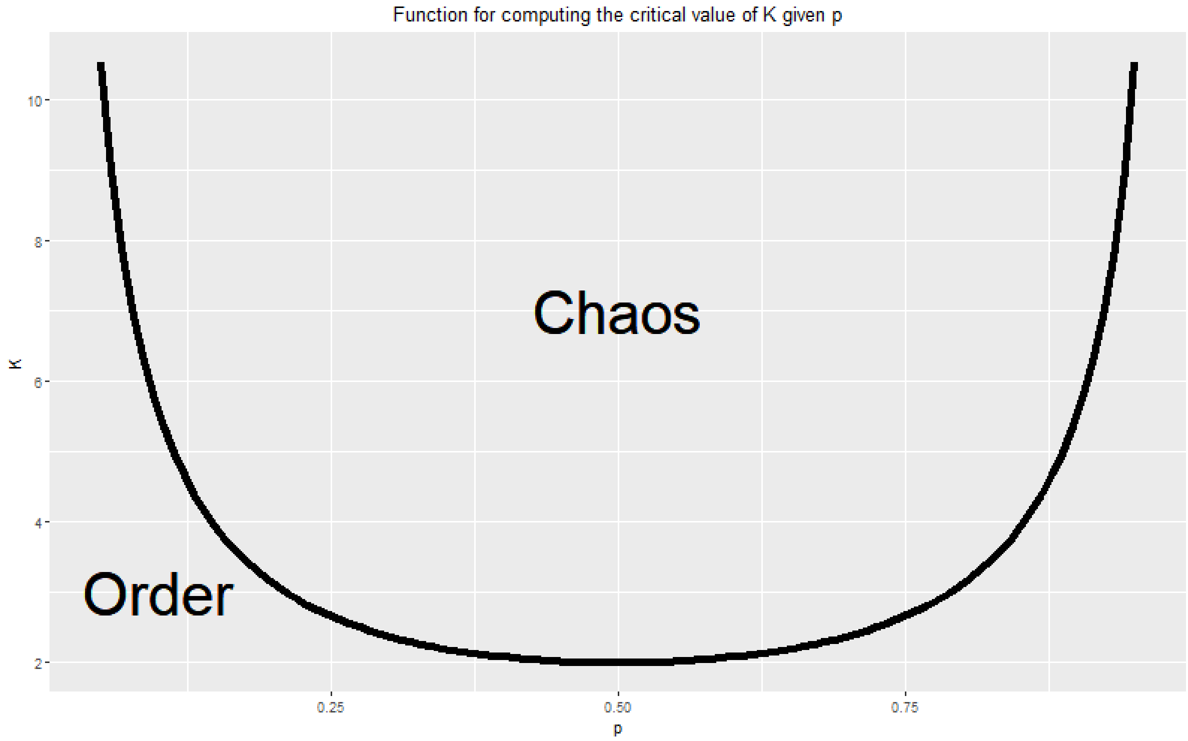


Figure 2.2: shape of function 2.1. Networks with parameters that are under the curve have frozen dynamics. Networks with parameters that are above the curve have long attractors. Networks whose parameters are on the curve are critical.

2.2 Boolean Networks as control software for robot

In the experiments we use **critical RBNs** without self loops as control software for robot. A moderate number of self loops makes RBNs more suitable to show differentiation phenomena[7]. Increasing the number of self loops raises the attractors but reduces robustness and stability[17]. The robot, detailed described in section 3.2, has a set of sensor Boolean values constantly updated over simulation time. The sensors are coupled with nodes of the network in a one-to-one relation. For each time step the nodes' value of the coupled ones are forced to be equal to the corresponding sensor values, then the network is updated. A subset of nodes, with different nodes from the ones already used, are coupled with the robot's actuators. The Boolean value of an actuator is forced to be a new value that is function of the corresponding node's value. Figure 2.3 represents the architectural scheme previously described.

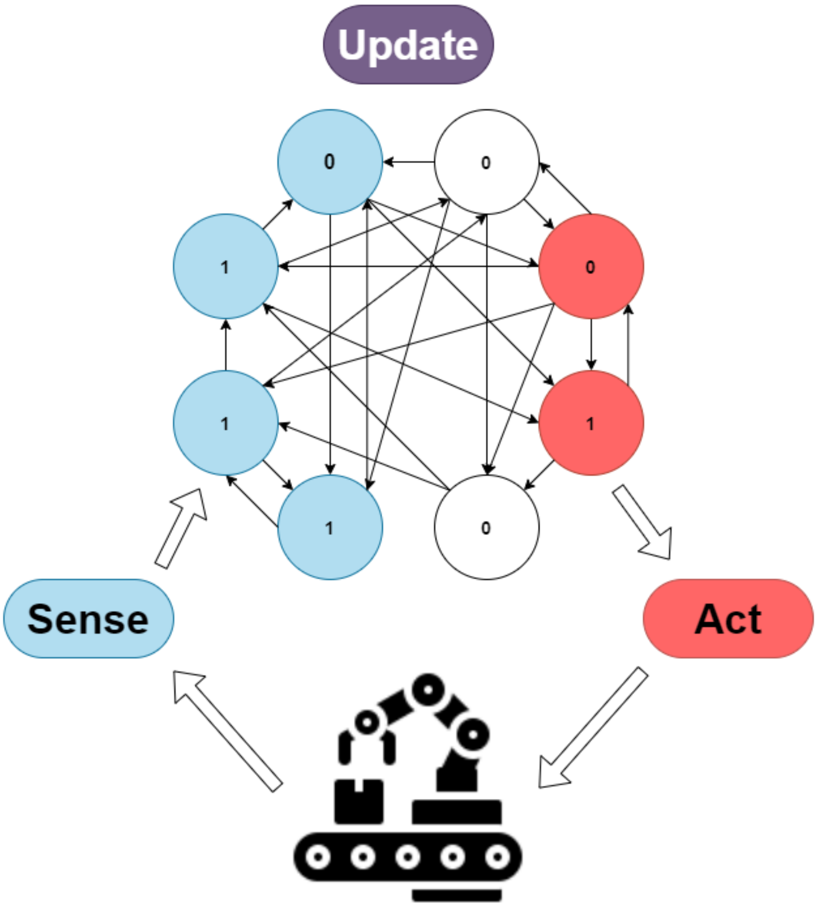


Figure 2.3: high level architectural scheme of the robot's controller. The robot acquires sensor information and forces sensor values into the network, blue nodes. Network updates its state and the Boolean values of the red nodes are used to set the actuator values. In the example of the image the Boolean network has $N = 8$ and $K = 3$.

We choose to use the following parameters for the networks:

- $N = 20$, $K = 3$, $p = 0.79$;
- $N = 50$, $K = 3$, $p = 0.79$;
- $N = 100$, $K = 3$, $p = 0.79$.

Self loops are not allowed and a node cannot output to the same node twice, furthermore nodes' function and vertices do not change over experiment time, there is no network evolution.

The space of all the possible networks is more than exponential in K. We have to consider all the possible Boolean functions, the number of nodes and the possible ways of connecting nodes. The functions are 2^{2^K} so, with $K = 3$, they are 256. The number of ways of connecting nodes, because we do not allow self loops and allow only one directed connection from a node n_i towards a node n_j, is $N - 1$ for the first vertex, $N - 2$ for the second one and $N - 3$ for the third. The number of all possible networks is:

$$Networks = 2^{2^K} * (N)_{K+1} \tag{2.2}$$

With $K = 3$ the numbers of unique networks in function of the number of nodes are:

- $N = 20$, $networks = 29,767,680$;
- $N = 50$, $networks = 1,414,963,200$;
- $N = 100$, $networks = 24,092,006,400$.

Some of them are equivalent. By properly applying some operations on the network, like changing Boolean functions or connections, it is possible to obtain an equivalent one (same state transitions).

Chapter 3

ARGoS simulation software

This chapter describes the simulation software used for running the experiments. The simulation model and environments are presented in section 3.1. In section 3.2 the robot *foot-bot* is shown in detail along with its sensors and actuators. In the last section 3.3 is briefly described the GUI to inspect a single experiment.

3.1 Environment

The simulation software used for the experiments is **ARGoS** [20] (Autonomous Robots Go Swarming). ARGoS is a fast multi-robot simulator developed at IRIDIA, Artificial Intelligence research laboratory of the Université Libre de Bruxelles. It is easy to use and versatile, it can handle different scenarios for several applications. ARGoS grants high accuracy, high flexibility and high efficiency. For accuracy it is intended the close gap between reality and simulation. Flexibility refers to let the users add new features such as new robot types or new sensors. Efficiency concerns the ability to provide satisfactory run-time performance.

An experiment configuration file must be written for each type of experiment. The file is in *xml* and the main things that can be specified are:

- *framework*, set up of internal parameters of the system (ex. threads) and of the experiment (ex. length);

- *controllers*, list of user defined controllers and configurations;

- *arena*, list of entities to add to the arena at the beginning of the experiment and their initial positions;
- *visualization*, optional configuration for visual inspection, see section 3.3.

For the experiments controllers are written as external files in *Lua* (version 5.4) [21]. Control software is written thanks to the previous work of A. Gnucci, who has developed a Lua module to create RBNs and to modify them online [12]. Further details on code are in section 4.1. In the same chapter 4.2 are also described the arenas used for each task.

3.2 Foot-bot

The robot chosen for the experiment is called *foot-bot* (FB), which is already available in ARGoS. The real FB[5] is a ground-based robot that moves through a combination of wheels and tracks called *treels*. During the simulations we treat the *treels* as a simple pair of wheels. In figure 3.1 the real FB is shown along with its main characteristics. What we use in experiments:

- *proximity*, there are 24 sensors equally deployed in a ring around the robot capable of perceiving near objects. A single sensor can detect object up to 10 cm, it returns a reading composed of an angle in radians and a value in the range $[0, 1]$. The angle corresponds to where the sensor is located, while the a value equal to 0 corresponds to no object being detected by a sensor, values greater than 0 mean that an object has been detected. The value increases as the robot gets closer to the object;
- *base_ground*, there are 8 sensors equally deployed in a ring around the robot capable of distinguish the ground color. Each sensor returns a Boolean value, 0 if ground is black or dark grey, 1 if ground is white or light grey, and a offset that corresponds to the position read on the ground by the sensor;
- *light*, there are 24 sensors equally deployed in a ring around the robot capable of perceiving light sources. Like in the case of *proximity*, each sensor returns an angle in radians, corresponding where the sensor is located, and a value in range of $[0, 1]$.

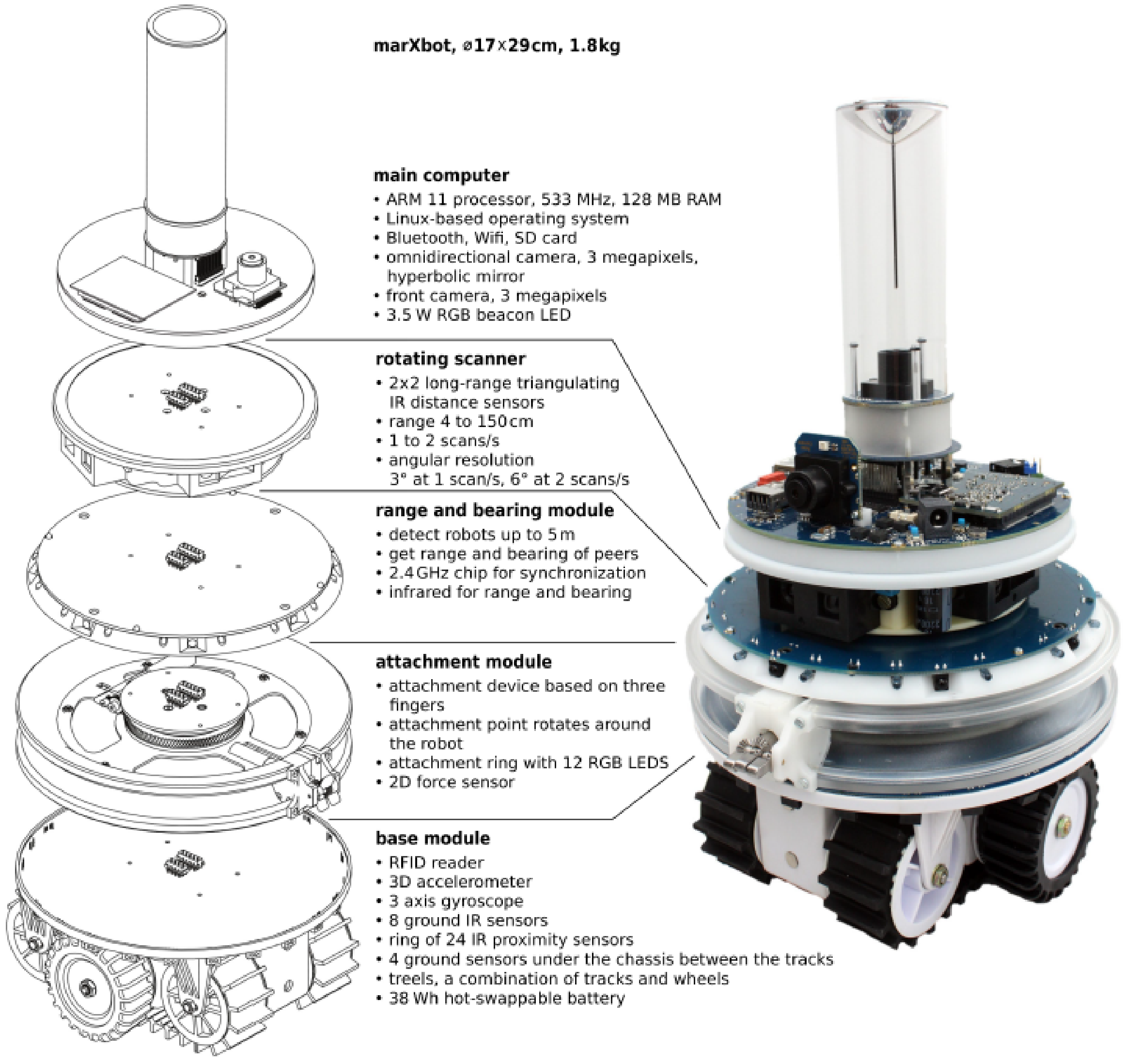
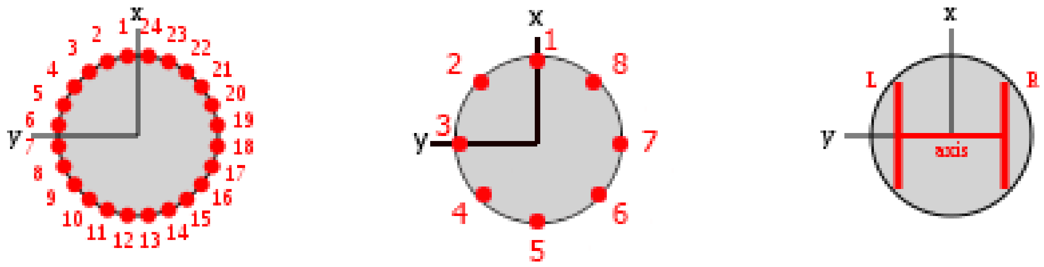

marXbot, ø17x29cm, 1.8kg
main computer
• ARM 11 processor, 533 MHz, 128 MB RAM
• Linux-based operating system
• Bluetooth, Wifi, SD card
• omnidirectional camera, 3 megapixels, hyperbolic mirror
• front camera, 3 megapixels
• 3.5 W RGB beacon LED
rotating scanner
• 2x2 long-range triangulating IR distance sensors
• range 4 to 150cm
• 1 to 2 scans/s
• angular resolution 3° at 1 scan/s, 6° at 2 scans/s
range and bearing module
• detect robots up to 5m
• get range and bearing of peers
• 2.4GHz chip for synchronization
• infrared for range and bearing
attachment module
• attachment device based on three fingers
• attachment point rotates around the robot
• attachment ring with 12 RGB LEDS
• 2D force sensor
base module
• RFID reader
• 3D accelerometer
• 3 axis gyroscope
• 8 ground IR sensors
• ring of 24 IR proximity sensors
• 4 ground sensors under the chassis between the tracks
• treels, a combination of tracks and wheels
• 38 Wh hot-swappable battery

Figure 3.1: on the left the exploded view of the robot with focus on sensors and actuators. On the right an image of the real *foot-bot*. Figure taken from [5].

A value of 0 corresponds to no light being detected by a sensor, while values grater than 0 mean that light has been detected. The value increases as the robot gets closer to a light source;

- *wheels*, the real robot moves through *treels*, two sets of wheels and tracks. For simplicity we consider *treels* as normal wheels. Wheels are deployed along the y

axis (see figure 3.2 (c)). One can specify the linear velocity in cm/s for each wheel.



(a) (b) (c)

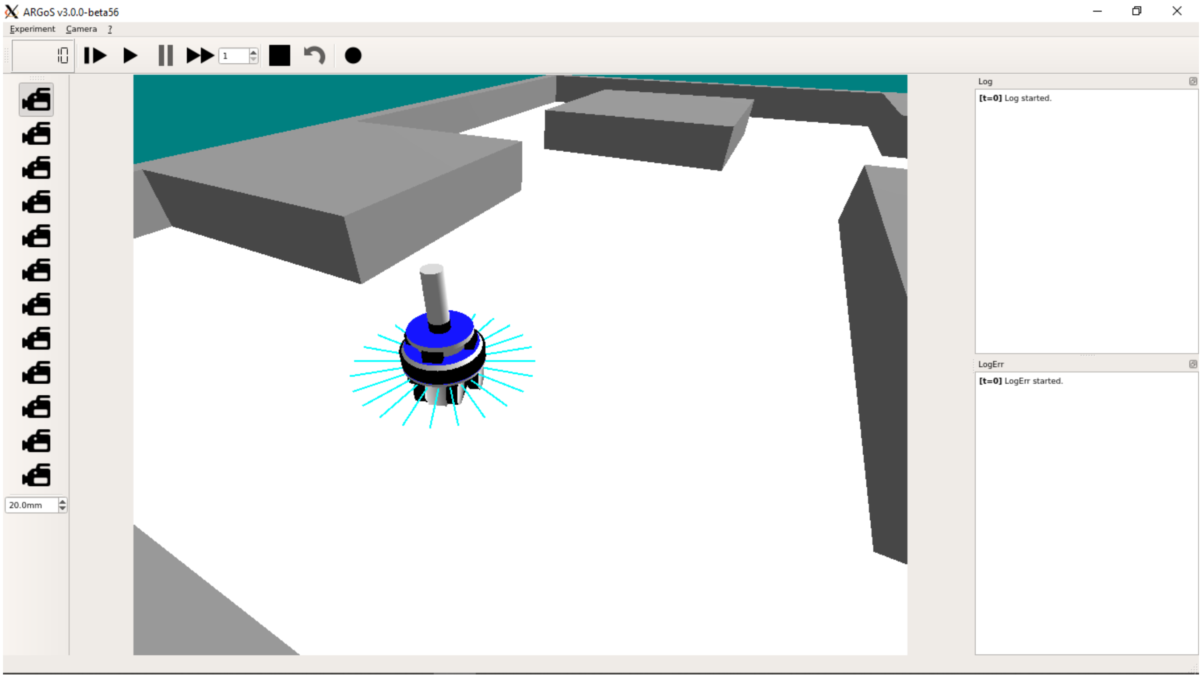
Figure 3.2: (a) placement of *proximity* and *light* sensors. (b) Placement of *base_ground* sensors. (c) Placement of the wheels.

Because the number of sensors for *proximity* and *light* is too high for BNs of small dimension and computational prohibitive in the calculus of information theory measures with standard resources, these sensors are logically grouped to reduce their number. Sensors are grouped in 8 new sensors, each of them containing 3 adjacent original sensors. A new sensor value is equivalent to the maximum of its 3 original sensor values. Moreover, with the exception of *base_ground* sensors, sensors values are in continuous domain. Due to the Boolean nature of the controller, these continous values are binarized with threshold $t = 0.1$.

3.3 Visual inspection

The simulation software can be executed in two mode: with a graphics interface and without it. By default, when experiment are executed for the first time, visualization is disable. Instead, when an experiment needs to be inspected to find out some particular behaviour of the robot visualization is enable. A visual inspection of the experiments gives qualitative information about the ones that have measures value of difficult interpretation.

The window offered by ARGoS is visible in figure 3.3. The main panel in the center shows the simulation and the user can move the camera or zoom in and out. The light



```
[t=0] Log started.
```

```
[t=0] LogErr started.
```

Figure 3.3: ARGoS GUI. In the center the simulation with the *foot-bot* in the arena. On the right two panels for standard output and errors. On the top control buttons of the simulation.

blue rays around the robot are a representation of what a particular type of sensors perceives. In the example rays refer to *proximity* sensors and a ray becomes red when the corresponding sensor value is greater than 0. On the top there is a control panel where the user can start, stop, restart or fast forward the simulation. Here there is also a logic time counter. On the right there are two panels for standard output and errors. They are quite handy during the debugging phase. On the left there are some cameras that the user can configure in the ARGoS file of the experiment with preassigned positions and zoom.

Chapter 4

Experiments

In this chapter is briefly presented the coding environment used for the experiments execution (4.1). Then in section 4.2 tasks are detailed described with particular focus on the chosen objective functions.

The main objective of the experiments is to investigate possible correlations between robots that are suitable for a given task and high information theory measures. Secondly we check if the controller's parameter N is statistically influential in the achievement of the goal task. Finally we test if the same controller can be used with satisfactory results in a different task.

4.1 Coding environment

For each different task there is an ARGoS configuration file. The user must specify in *controllers* tag which robot control software to use (reference to external file). ARGoS supports both C++ and Lua. We have chosen Lua 5.3 [21] mainly for two reasons. First of all is higher level than C++ and it has a garbage collector. Secondly, thanks to the work of A. Gnucci [12], we can use a module that allows to create RBNs and to perform on top of them almost all operations we need.

User defines three functions in Lua files:

- *init*, executed only once at the beginning of the experiment;
- *step*, it is executed sequentially for each logic time step of the experiment;

- *destroy*, executed only once at the end of the experiment.

The common structure and main functions of controllers are presented in listing 1. At the end of an experiment it is output the network architecture, connections and Boolean functions, and a triple for each time step. The triple contains the network's state, the sensors values and the robot's position. The experiment executing process has been automatized with bash scripts. Output is redirected into text files for later analysis.

```lua
function init()
  math.randomseed(os.clock())
  network = BooleanNetwork(network_options)
end

function step()
  n_step = n_step + 1
  if n_step <= NETWORK_TEST_STEPS then
    collect_current_state()
    collect_robot_position()
    local inputs = get_input()
    collect_current_sensor_values(inputs)
    local binary = get_binary_input(inputs)
    execute_network(binary)
  end
end

function destroy()
  if n_step > 0 then
    print("network: " .. one_line_serialize(network))
    print(get_data_per_step())
  end
end
```

Listing 1: control software main functions: *init* randomly initializes the network, *step* collects sensor values and robot position, then it executes the network and moves the robot, *destroy* prints the network and experiment data.

The module for RBNs has been expanded with a function that allow to reload stored networks. It is now possible to rerun an experiment to visually inspect it or to load a network firstly used in one task to test it in another one.

4.2 Tasks

Three well known tasks in robotics as been selected for these experiments: obstacle avoidance, path following and photo-taxis. For all of them the same kind of robot has been used, the *foot-bot*, described in section 3.2. The controller is always a critical RBN with parameters $K = 3$ and $p = 0.79$ (see 2.1.1). The number of nodes N of the network can be 20, 50 or 100. In all configurations nodes are indexed. The first 8 nodes are coupled with the specific set of 8 sensors used in the task. The 9th and 10th nodes are coupled with the two wheels. Robots perceive through their sensors and move in the arena. The wheels can be:

- both active, in this case the robot goes straight forward with a linear velocity per wheel of 15 cm/s;
- right active and left inactive, the robot turns left. Right wheel has a linear velocity of 15 cm/s, the left one has 0;
- left active and right inactive, the robot turns right. Left wheel has a linear velocity of 15 cm/s, the right one has 0;
- both inactive, the robot does not move.

Each task has a unique objective function to evaluate the performance of the robot. To have a fair meter of comparison we run experiments with a controller that does random walk. The random walk is performed as follow:

- goes straight forward for S step, where S is a random uniform integer between 20 and 60;
- turns of R radians, where R is a random uniform value between $-\pi/2$ and $\pi/2$;
- repeat until the end of the experiment.

The robot moves at the same speed of the robots controlled by RBNs. When turning, the robot is in the same spot for all the duration of the rotation.

In all experiments of the same task robots start at the same position and orientation. For each task has been executed 1000 experiments per network configuration (20, 50 and 100 nodes) and 1000 experiments with the handcrafted random walk controller.

4.2.1 Obstacle Avoidance

A robot is in an arena limited by walls with some fixed obstacles. The goal for the robot is to freely move exploring the arena as much as possible while avoiding obstacles. Robot can perceive nearby objects thanks to *proximity* sensors. On *foot-bot* there are 24 sensors, too many for small networks and for fast accurate calculus of information theory measures. So sensors are grouped in 8 compound sensors, 3 adjacent sensors each. The value of the new ones is equal to the maximum of the old ones. Because the sensors value is in continuous domain, it is binarized with threshold $t = 0.1$.

The objective function for obstacle avoidance is:

$$F_{oa} = max(\overrightarrow{p_0, p_t}) * \frac{1}{T} * \sum_t o_t * \overrightarrow{p_t, p_{t+1}} \tag{4.1}$$

where arrow means 2 dimensions euclidean distance between two points, p_t is the robot position at time t, o_t is equal 0 if at least one sensor is 1 at time t, it is 1 if all sensors are 0 at time t. Qualitative speaking the function rewards robots that move without any obstacles nearby and that reach a far position from the starting point. The lower bound is 0, a robot that does not move for the entire duration of the experiment, while the upper bound depends on the arena morphology, on the robot starting point and on its velocity.

Obstacle avoidance experiments last 200 seconds and for each second there are 10 time logic steps. Maximum wheels' linear velocity is 15 cm/s, so for a single time step a robot can run 1,5 cm. With this time and the current velocity a robot is able to explore almost the hole arena.

The arena is a 6 meters square with fixed solid borders. Walls are 5 cm re-entrant, so the actual surface is 5.9 x 5.9. In the arena there are 9 fixed squared obstacles displayed as in figure 4.1. The central obstacle is 1.5 x 1.5 with center in $\{0, 0\}$, the other

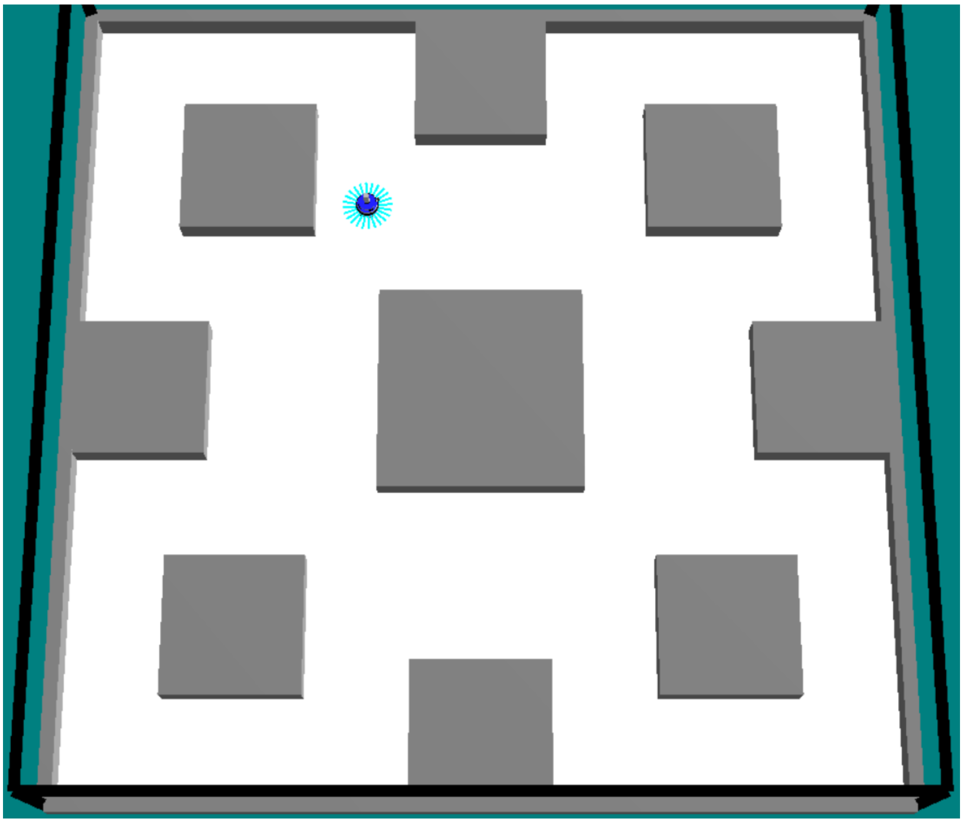


Figure 4.1: obstacle avoidance arena.

obstacles are 1 x 1 and they are placed in $\{0, 2.5\}$ (top), $\{1.75, 1.75\}$ (top-right), $\{2.5, 0\}$ (right), $\{1.75, -1.75\}$ (bottom-right), $\{0, -2.5\}$ (bottom), $\{-1.75, -1.75\}$ (bottom-left), $\{-2.5, 0\}$ (left), $\{-1.75, 1.75\}$ (top-left). A robot that straightforward hits an obstacle cannot move, however if the robot hits the object with a certain angle it can move with restriction and its trajectory is altered. The starting position it is always $\{-0.856, 1.474\}$ and it has been chosen randomly. Due to the robot radius, which is 8.5 cm, the maximum distance from the starting position that can be reached is the distance between $\{-0.856, 1.474\}$ and $\{2.865, -2.865\}$ (bottom right corner). The value of this distance is almost 5.1594 meters. If we suppose that the robot always moves forward, which is obviously an overestimate, the mean of the run distance per step is equal to 0.015 meters. Given the described arena and these robot's characteristics the objective function is limited by $0 \leq F_{oa} < 5.716 * 0.015 \approx 0.0857$.

4.2.2 Path Following

A robot is in an arena limited by walls with a black path on the floor. The goal for the robot is to move in the arena while staying on the circuit and possibly run all over it as much as it can. Robot can perceive floor color thanks to *base_ground* sensors. There are 8 sensors and sensors value is Boolean, 0 if color is black, 1 if it is white.

The objective function for path following is:

$$F_{pf} = max^{*}(\overrightarrow{p_0, p_t}) * \frac{1}{T} * \sum_{t} g_t * v_t \tag{4.2}$$

where g_t is 1 if at least one sensor is 0 (the robot is on the circuit), it is 0 if all sensors are 1 (robot away from the circuit) at time t. v_t is 1 if at least one wheel is active, it is 0 if the robot does not move at time t. The max^{*} function computes the maximum distance between the robot starting position and the furthest position reached by the robot during the experiment while it is on the circuit. Robots that keep moving on the path and completely run over it have the highest reward. The lower bound of the objective function is 0, robot that does not move at all, while the upper bound depends on the arena and on the path.

Path following experiments last 500 seconds and for each second there are 10 logic steps. Maximum wheels linear velocity is 15 cm/s, so for a single time step a robot can do 1,5 cm. With this time and the current velocity a robot is able to complete once the hole path.

The arena is a 6 meters square with fixed solid borders. Walls are 5 cm re-entrant, so the actual surface is 5.9 x 5.9. The black path is like in figure 4.2, it is very heterogeneous to make this task as difficult as possible and it covers almost the 12.24% of the surface. The starting position is always $\{-0.771, 1.327\}$, it has been chosen randomly and a robot in this position is on the path. The furthest point on the path from the starting position is approximately $\{2.355, -2.1\}$. Considering the radius of the robot we can overestimate the furthest point in $\{2.412, -2.163\}$. Function $max*$ cannot have greater value than 4.724, and this is the same maximum limit for F_{pf} in the case of a robot always moving on the path. So the objective function is limited by $0 \leq F_{pf} < max^{*}(p_0, p_t) \approx 4.759$.

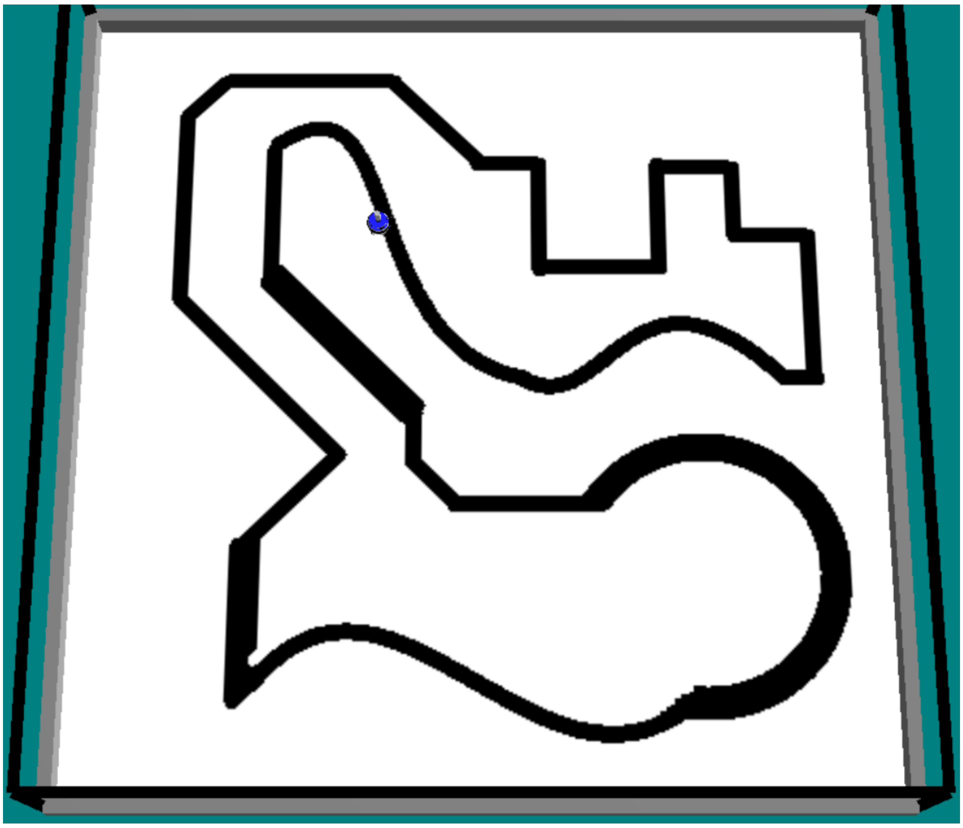


Figure 4.2: path following arena.

4.2.3 Phototaxis

A robot is in an arena limited by walls with a light source. The goal for the robot is to move as close as possible to the light source and in the shortest amount of time. Robot can perceive light intensity through *light* sensors. Like for *proximity* sensors in obstacle avoidance, *light* sensors are grouped with the same principle.

The objective function for photo-taxis is:

$$F_{pt} = \frac{T}{\sum_t \overrightarrow{p_t, L}} \tag{4.3}$$

where L is the light source position. Robots that immediately move towards the light source have the highest values while robots that performs anti-photo-taxis have the lowest values. Unlike in the previous tasks, the objective function is not limited by 0, a standing robot will receive the reciprocal of its distance from the light source.

Phototaxis experiments last 50 seconds and for each second there are 10 logic steps. Maximum wheels linear velocity is 15 cm/s, so for a single time step a robot can do 1,5 cm. With this time and the current velocity a robot is able reach the light source with some time margin.

The arena is a 8 meters square with fixed solid borders. Walls are 5 cm re-entrant, so the actual surface is 7.9 x 7.9. The light source is placed in position $\{1.5, 0, 0.5\}$ as in figure 4.3. The starting position is always $\{-1.857, 1.474\}$ and the distance between the robot and the light source is computed in the 2D euclidean distance. With the current settings, starting position, light position and velocity, ignoring the starting robot orientation which is also always the same for all the experiments, the objective function is limited by $0.167 < F_{pt} < 1.107$. Indeed, because the robot is oriented pointing to the closest wall, the upper bound is a bit overestimated. A standing robot has an objective function value of almost 0.2728.

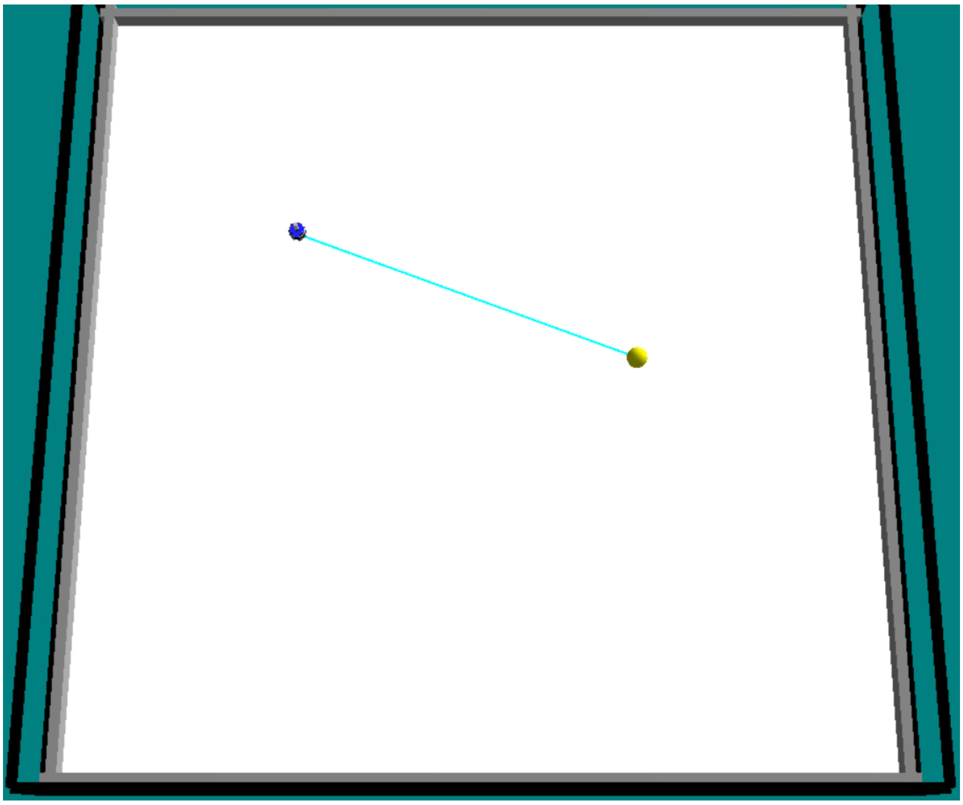


Figure 4.3: phototaxis arena.

Chapter 5

Results

This chapter summarizes the experiments that have been executed and all main results. Results are grouped by task and for each of them the following information are reported:

- distribution and study of the chosen objective function;
- computing and analysis of information theory measures;
- correlation analysis between high objective function and high measures.

Then we switch the best RBNs based on objective function and measures to use them in the other two tasks. We want to check if a RBN that is good in performing a task can be used with success in another one.

We have chosen to use four information theory measures:

- sensor entropy (SE), it is the entropy 1.1 computed on the sensors values;
- predictive information (PI), it is the mutual information 1.7 computed on sensors values at time t and on actuators values at time $t + 1$;
- transfer entropy (TE), it is the transfer entropy 1.8 computed from sensors value towards actuators values;
- reverse transfer entropy (RTE), it is the transfer entropy computed from actuators values towards sensors values.

Probabilities are directly estimated by frequencies from experiment files for each execution. Because for all tasks we use 8 sensors and 2 actuators, the theoretical measures' upper bound is 2 (bits) with the exception of SE which is limited by 8 (bits). These limits are far from being reached due to the arenas' configurations, indeed it is physically impossible for a robot to observe certain sensors values.

For each task have been executed 1000 experiments per RBN configuration (20, 50 and 100 nodes), plus 1000 experiments with the random walk controller. In total there are 4000 experiments per task.

5.1 Obstacle Avoidance

5.1.1 Objective function

F_{oa} score distributions are summarized in figure 5.1. RBNs score distributions are all wide spread with general low medians and means. The majority of robots has F_{oa} score near 0. This is due to robots that do not move at all or that after few steps stop moving. RBNs update towards an attractor that keep the values of the nodes linked to wheels equal to 0. Because sensor values are always the same along time if the robot does not move the RBN cannot escape the attractor. Another common way robots get stuck is they physically do it. Wheels may continue to move but the robot is facing a corner and cannot proceed. The robot does not go away from the corner, so its sensor values are the same and the RBN cannot escape the attractor.

F_{oa} score distribution for $N = 20$ has a lower mean than the ones with $N = 50$ and $N = 100$. All RBNs distributions are generally much worse than the random walk distribution. In fact for obstacle avoidance task random walk is a strategy often taken into account, at least as a benchmark for other options. However the fittest RBNs are capable of getting higher score than random walk. Through visual inspection robots that have a score equal to or higher than 0.05 do not get stuck and run almost a third of the arena.

Computing the Wilcoxon's test on fitness values between RBNs with different nodes number we obtain the result in table 5.1. Using a confidence level of 0.99 we have to reject the null hypothesis when we compare networks with $N = 20$ and $N = 50$ and networks

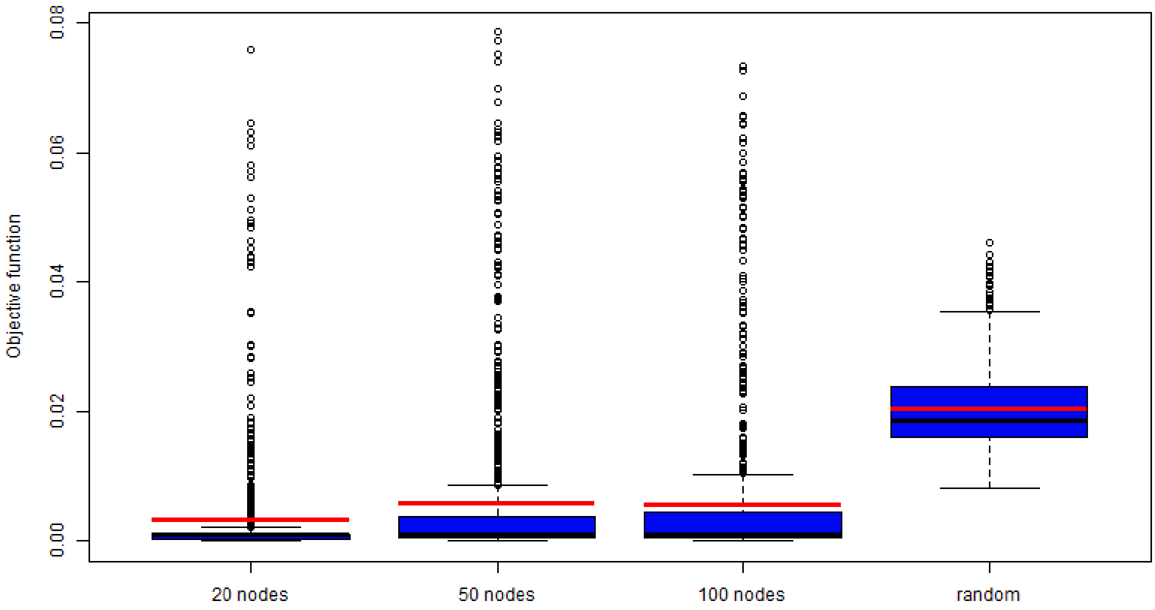


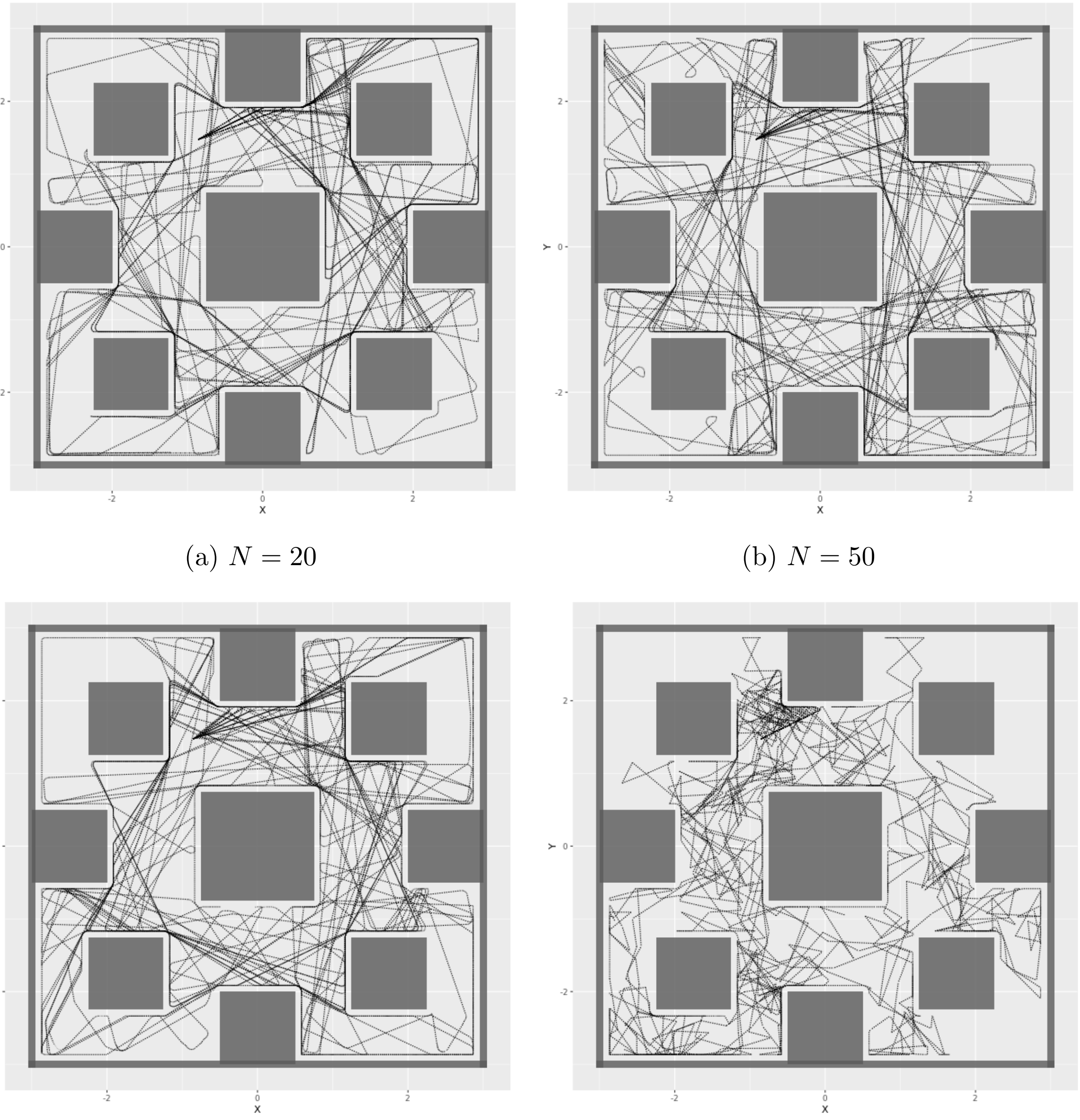
Figure 5.1: F_{oa} score distributions. From left to right distributions for: RBNs with $N = 20$, $N = 50$, $N = 100$ and random walk controller. Each distribution counts 1000 experiments.

N value	p-value	reject H_0
20 - 50	7.0785e-13	yes
20 - 100	7.0845e-16	yes
50 - 100	0.3806	no

Table 5.1: p-value results of Wilcoxon's test applied to F_{oa} score of networks grouped by nodes number in obstacle avoidance. Confidence is set equal to 0.99.

with $N = 20$ and $N = 100$. This means that the number of nodes is a parameter that has statistically influence on the probability of getting a better RBN for obstacle avoidance task. We cannot reject the null hypothesis in the case of $N = 50$ and $N = 100$, there is no statistically difference between the two distributions.

In figure 5.2 is reported a qualitative performance appearance of 20 robots that have obtained the highest scores per controller category. Top RBNs behave similar to each other, while top random walk controllers tend to explore less.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

(d) random walk

Figure 5.2: obstacle avoidance paths for top 20 robots with respect to F_{oa}.

5.1.2 Measures

Measures distributions are illustrated in figures 5.3 and 5.4. Qualitative speaking, besides SE, all measures have the majority of values near zero. We explain higher medians and means for SE because this measure concerns only sensors and has an higher full scale (8 bits instead of 2). Robots that perform the task sufficiently well see in the course of the experiment more distinct sensor values than the others. Seeing more sensor values means that the probabilities tend to be more evenly distributed across all possible values and this increases the entropy. The other measures instead consider also the actuators, in particular the information relation between sensors and actuators. However the nature of obstacle avoidance affects the bound between the two: robot's movement is altered by obstacles and perimeter walls and this leads to low metrics.

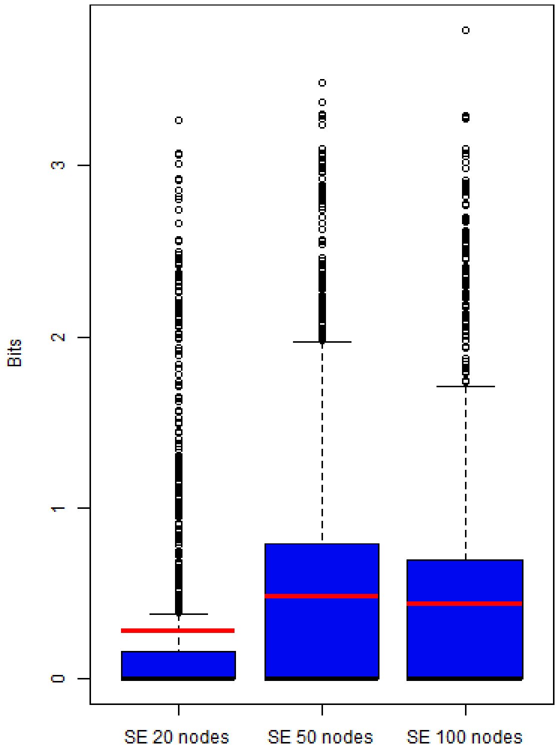


Figure 5.3: SE score distributions for obstacle avoidance. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

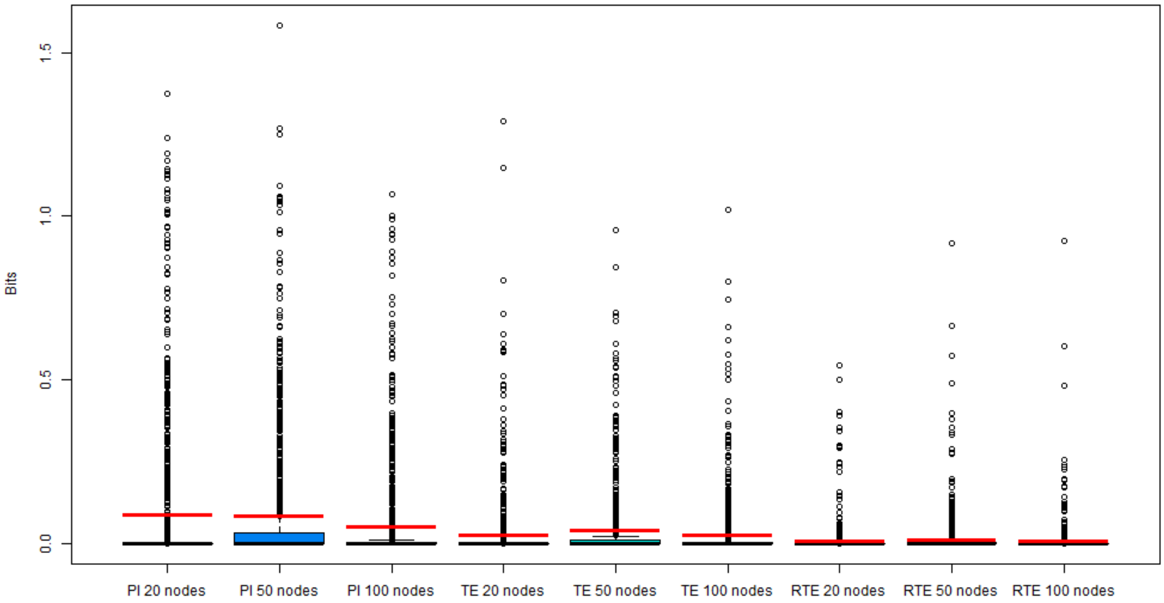


Figure 5.4: Measures score distributions without SE for obstacle avoidance. From left to right distributions for PI, TE and RTE. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

5.1.3 Correlation

Pearson's correlations values are reported in table 5.2.

Nodes	F_{oa} **- SE**		F_{oa} **- PI**		F_{oa} **- TE**		F_{oa} **- RTE**	
	correlation	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**
20	0.7675	5.1963e-195	0.5129	3.4324e-68	0.4616	6.1912e-54	0.3054	4.8483e-23
50	0.7913	1.608e-215	0.5193	3.7366e-70	0.4291	4.4539e-46	0.1745	2.7753e-08
100	0.8027	2.4937e-226	0.5725	3.6817e-88	0.3762	5.524e-35	0.1581	5.0006e-07

Table 5.2: Pearson's correlations between information theory measures and F_{oa}. Results are grouped by nodes number and each of them refers to 1000 experiments.

SE

First thing to notice is that SE shows a strikingly high positive correlation with the objective function. For all N values correlation is higher than 0.76. This is a confirmation of what anticipated in the previous section: robots suitable for obstacle avoidance see more different sensor values and therefore have higher SE.

SE - F_{oa} plots are shown in figures 5.5, 5.6 and 5.7. We observe similar plots for all network's configurations. When SE is between 0 and 1 F_{oa} rises in all RBNs. Also when SE grows until almost 2.5 bits F_{oa} still continues to increase. However there are several robots that have a medium or high SE value but a medium or low objective function. Thought visual inspection we have checked the behaviour of these robots and found that they do not move too far away from the starting position. Furthermore some of them loop around an obstacle or stop moving towards the end of the experiment. Robots that have the highest values of SE, greater than 2.5, do not have the highest objective function. A theoretical explanation is that to achieve obstacle avoidance a robot should be more often in a place where it does not perceive objects nearby. So the probability for the sensors value corresponding to this scenario to appear is much higher than the

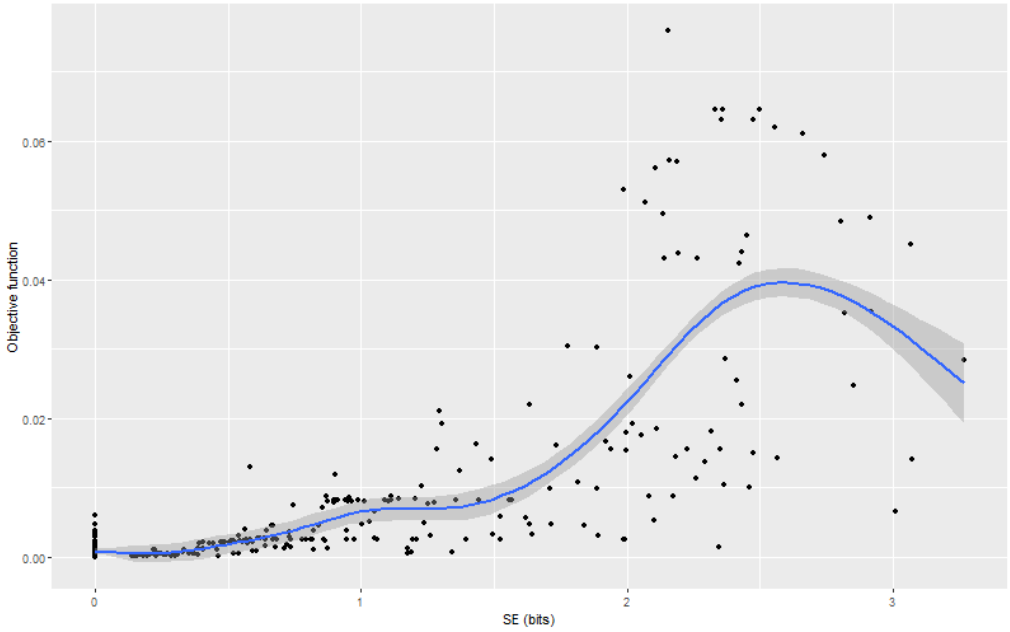


Figure 5.5: SE - F_{oa} plot for networks with $N = 20$.

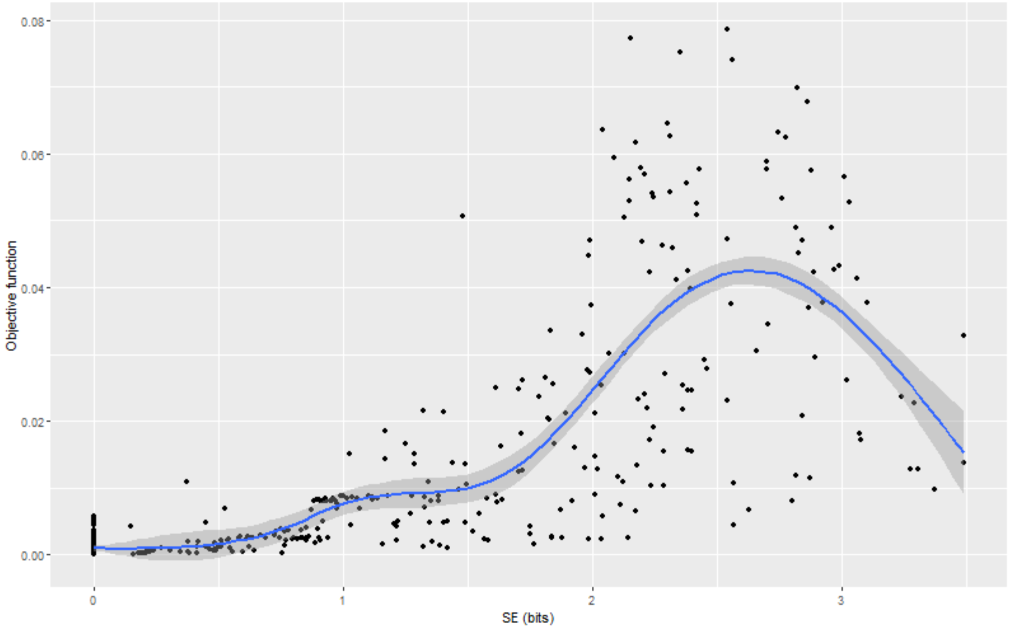


Figure 5.6: SE - F_{oa} plot for networks with $N = 50$.

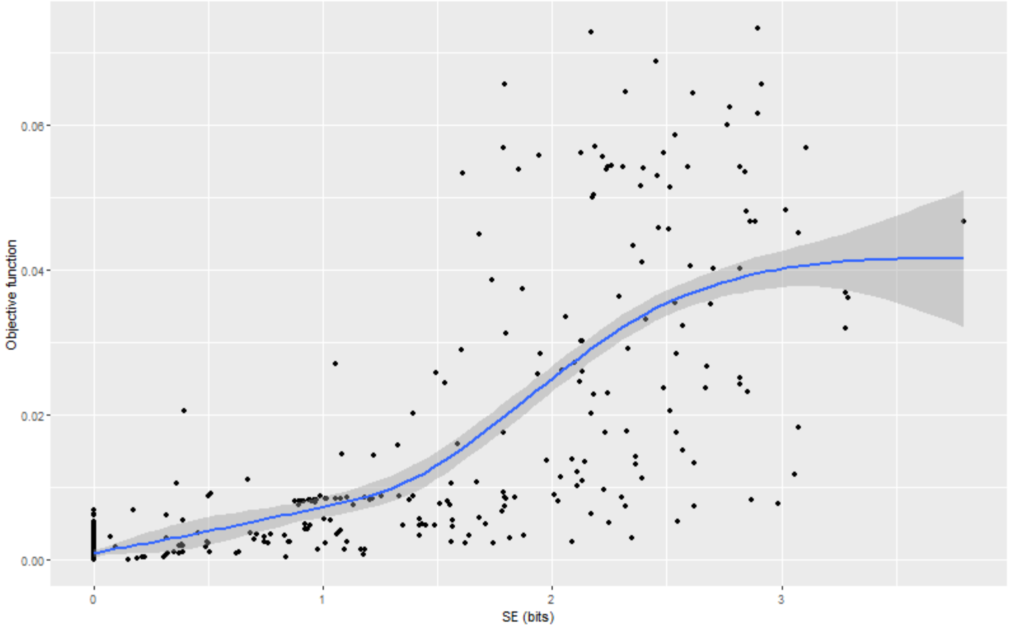
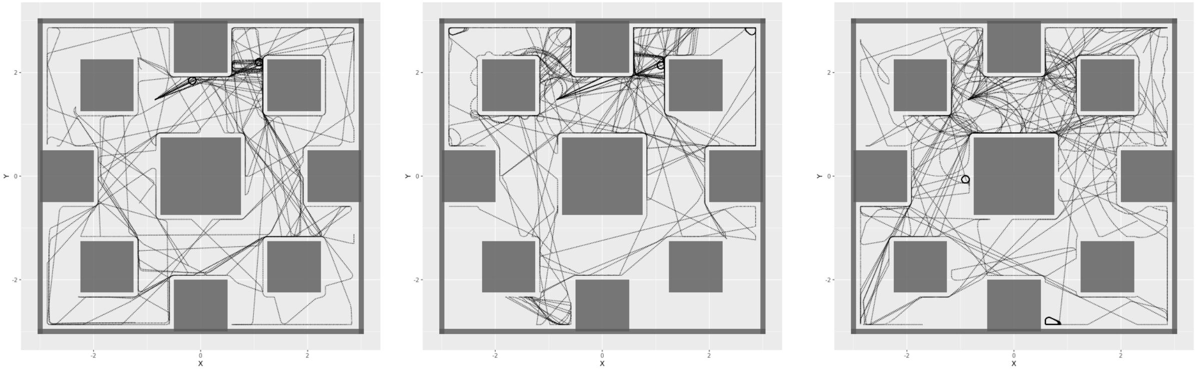


Figure 5.7: SE - F_{oa} plot for networks with $N = 100$.

values for obstacles near the robot. This unbalance in the probabilities prevents robots to get very high SE score. Instead robots that loop around obstacle have more equally balance probabilities and so higher SE despite being less capable of performing obstacle avoidance given the objective function we have chosen. The column in proximity of $SE = 0$ refers to robots that for the whole experiment perceive no obstacles, which corresponds to the starting state. Some of them have F_{oa} greater than 0 because they move in circle remaining close to the starting position.

Paths of the top 20 robots with respect to SE are reported in figure 5.8. Some robots explore almost the whole arena but also some of them loop around one or two obstacles. This last behaviour is visible at darker sections of the arena. Overall these figures are not too distant from the ones in 5.2.



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.8: obstacle avoidance paths for top 20 robots with respect to SE.

PI

PI - F_{oa} plots are shown in figures 5.9, 5.10 and 5.11. Predictive information is mildly positively correlated with F_{oa}, correlation is between 0.51 and 0.57. All plots are pretty similar to each other. In particular we can observe that in correspondence of PI ≈ 0.4 robots reach highest values for the objective function. Instead robots that have the highest PI have medium values of F_{oa}. These robots tend to run a circle with different radius or to bounce between two obstacles as reported in figure 5.12. Paths

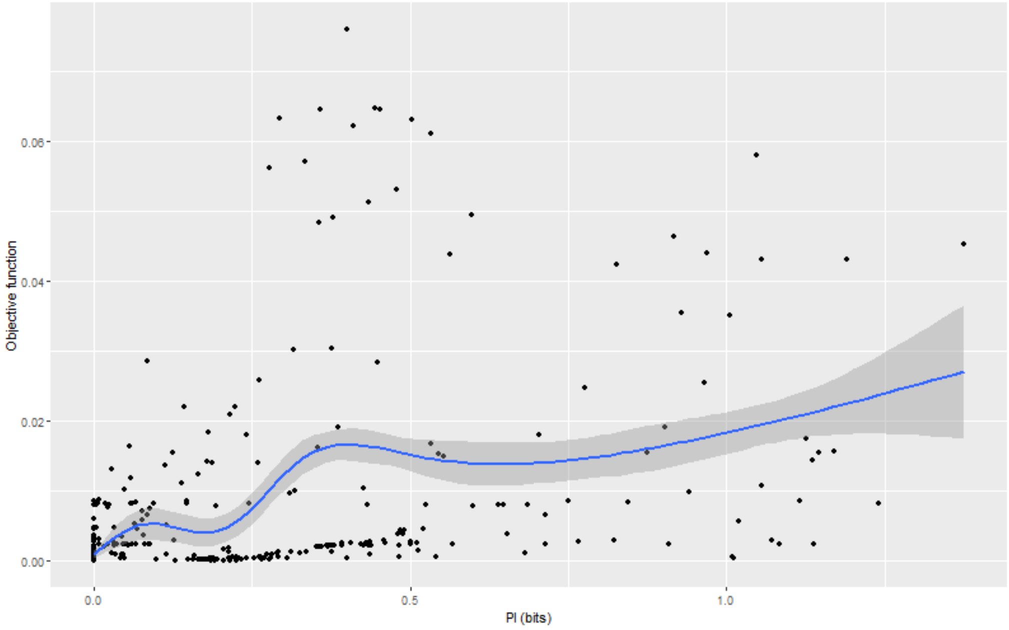


Figure 5.9: PI - F_{oa} plot for networks with $N = 20$.

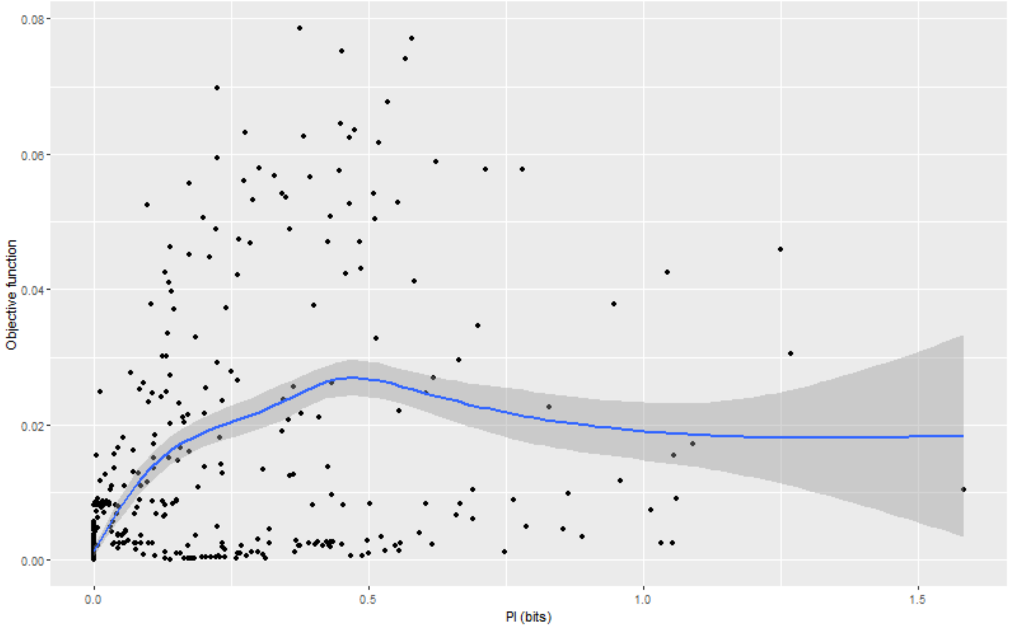


Figure 5.10: PI - F_{oa} plot for networks with $N = 50$.

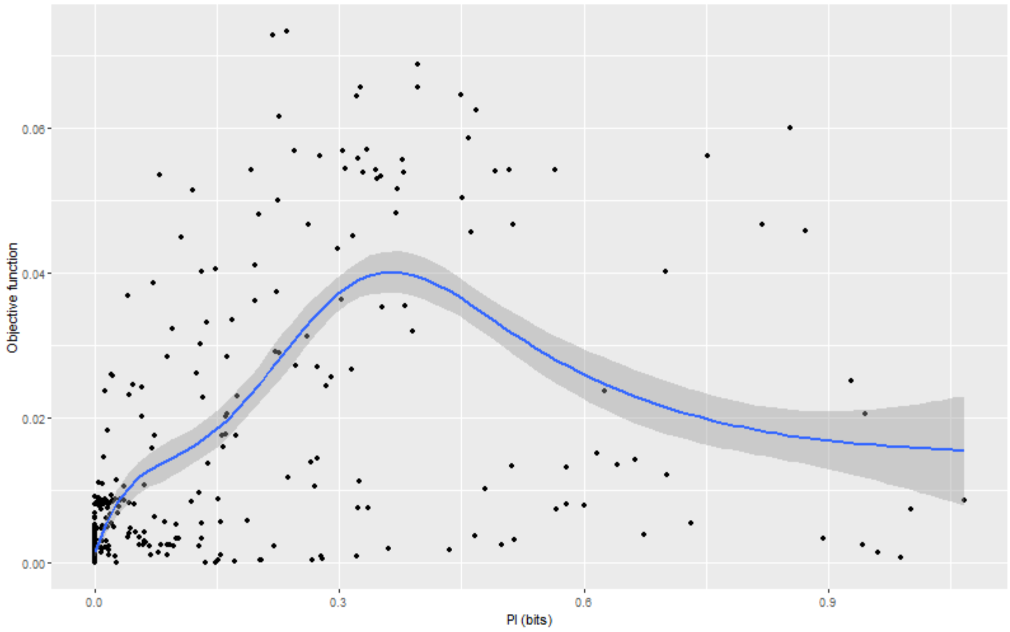
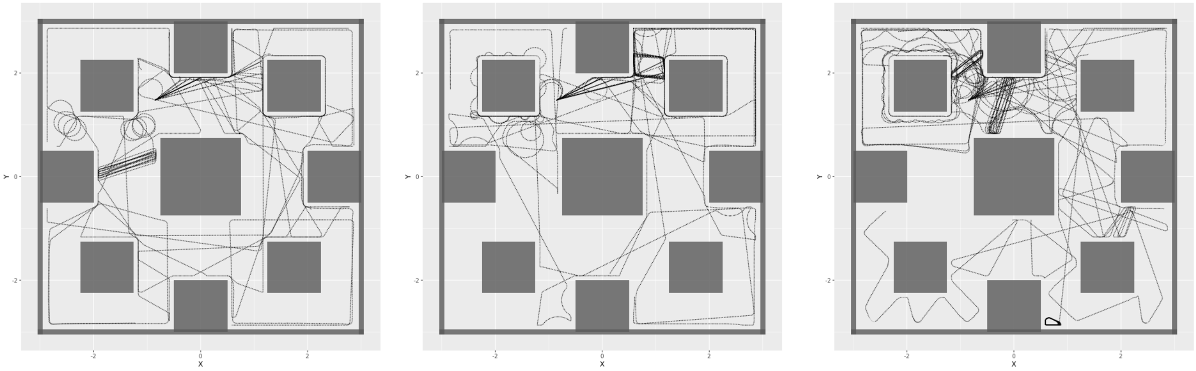


Figure 5.11: PI - F_{oa} plot for networks with $N = 100$.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

Figure 5.12: obstacle avoidance paths for top 20 robots with respect to PI.

are much different with respect to the ones in figure 5.2. We can assert that PI is only partially positively correlated with the chosen objective function for obstacle avoidance. In particular this is true for moderate values of predictive information.

TE

TE is weakly positively correlated with F_{oa}. Correlation values are between 0.37 and 0.46. TE - F_{oa} plots are shown in figures 5.13, 5.15 and 5.16. These plots resemble

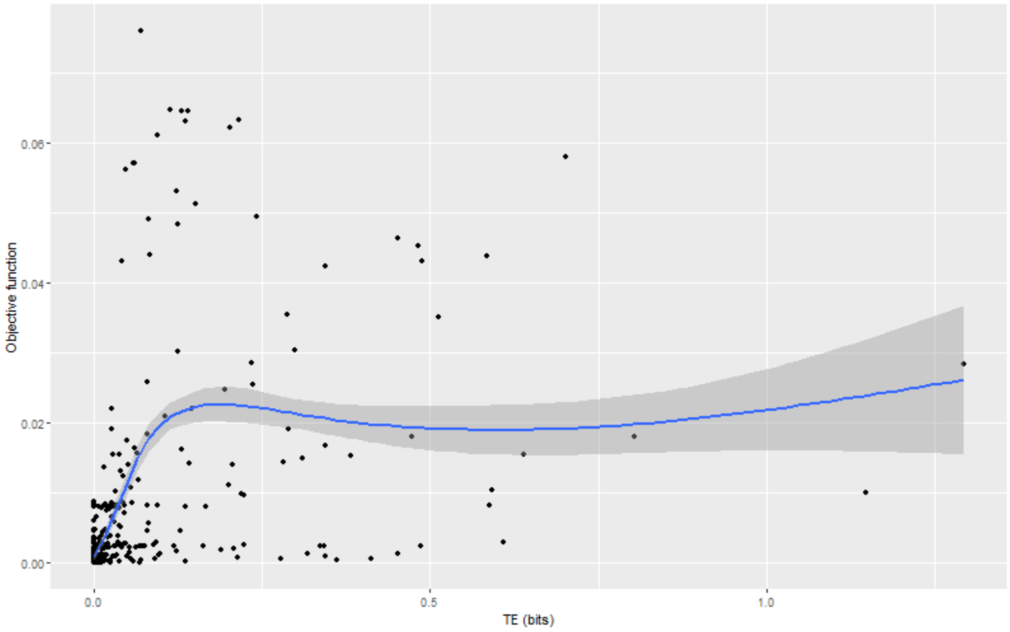
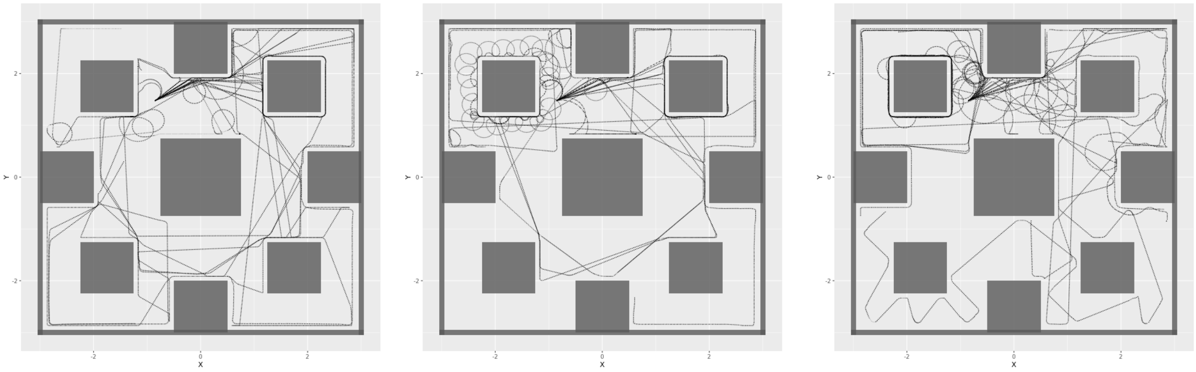


Figure 5.13: TE - F_{oa} plot for networks with $N = 20$.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

Figure 5.14: obstacle avoidance paths for top 20 robots with respect to TE.

the previous ones for PI. There is a concentration of robots near TE ≈ 0.12 that have

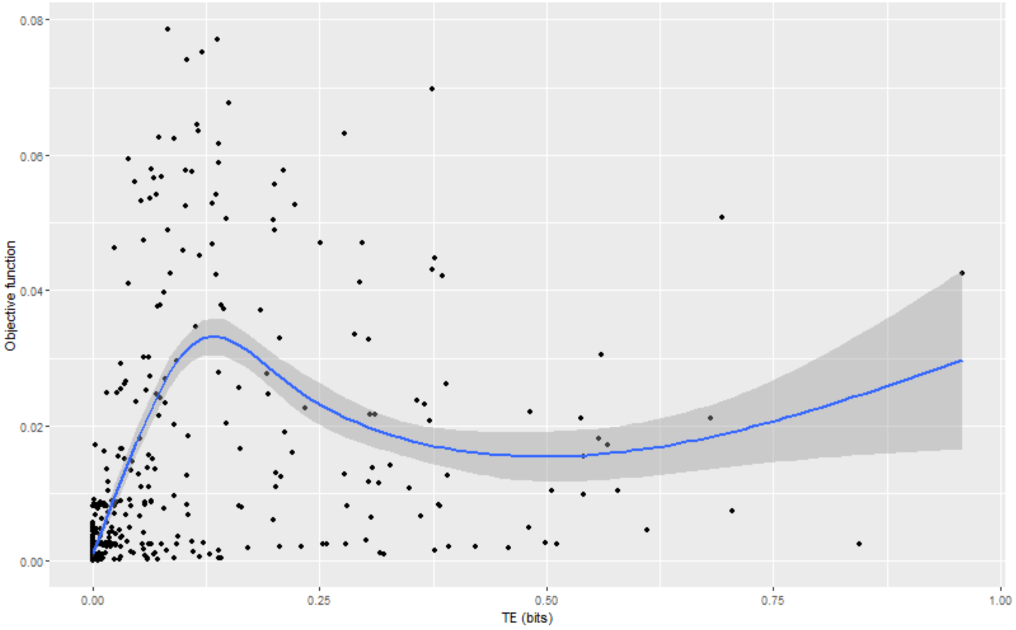


Figure 5.15: TE - F_{oa} plot for networks with $N = 50$.

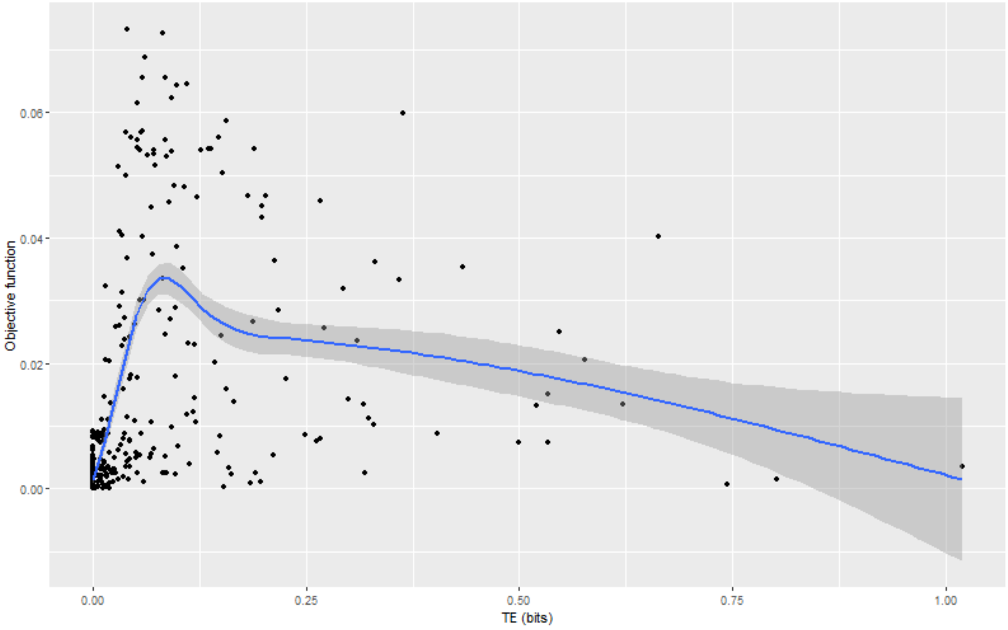


Figure 5.16: TE - F_{oa} plot for networks with $N = 100$.

high objective function. Above the threshold of 0.12 bits robots tend to perform worse. Robots that have highest values for TE have medium or low objective function. Paths of the top 20 robots that have highest transfer entropy are reported in figure 5.14. We observe more cycles and general less propensity to move far away from the starting point. We can assert that there is no evidence of notable correlation between transfer entropy and the chosen objective function for the task of obstacle avoidance.

RTE

RTE is not positively nor negatively correlated with F_{oa}. Correlation values are between 0.15 and 0.30. RTE - F_{oa} plots are shown in figures 5.17, 5.18 and 5.19. Both robots that are suited for obstacle avoidance and that are not with respect to the chosen objective function have very low value of RTE. Paths of the top 20 robots that have highest reverse transfer entropy are reported in figure 5.20. Like in TE, robots make cycles and they move less far away from the starting point. We can assert that there is no evidence of notable correlation between reverse transfer entropy and the chosen objective function for the task of obstacle avoidance.

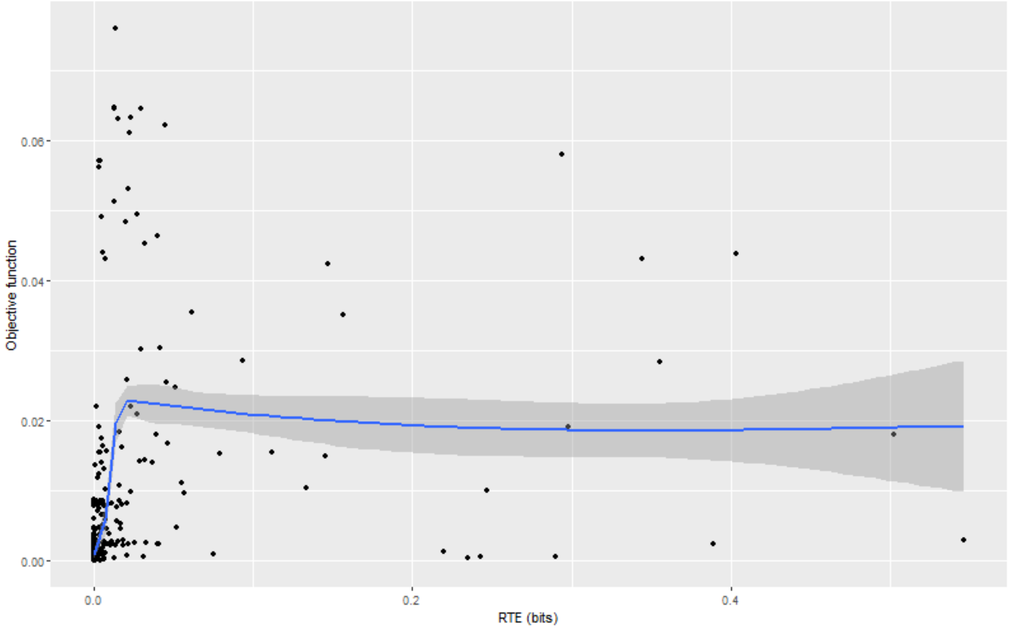


Figure 5.17: RTE - F_{oa} plot for networks with $N = 20$.

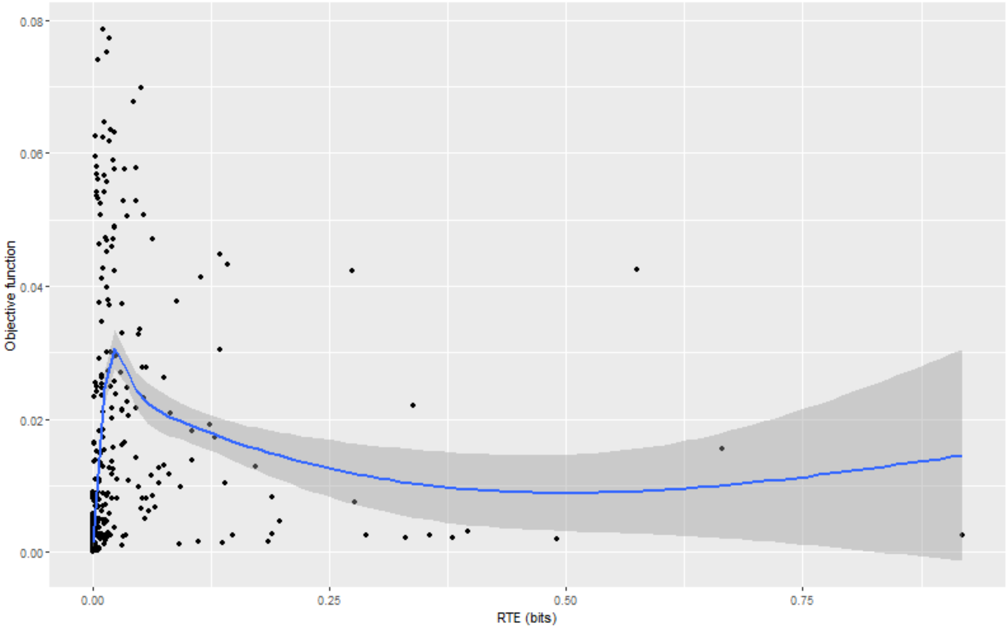


Figure 5.18: RTE - F_{oa} plot for networks with $N = 50$.

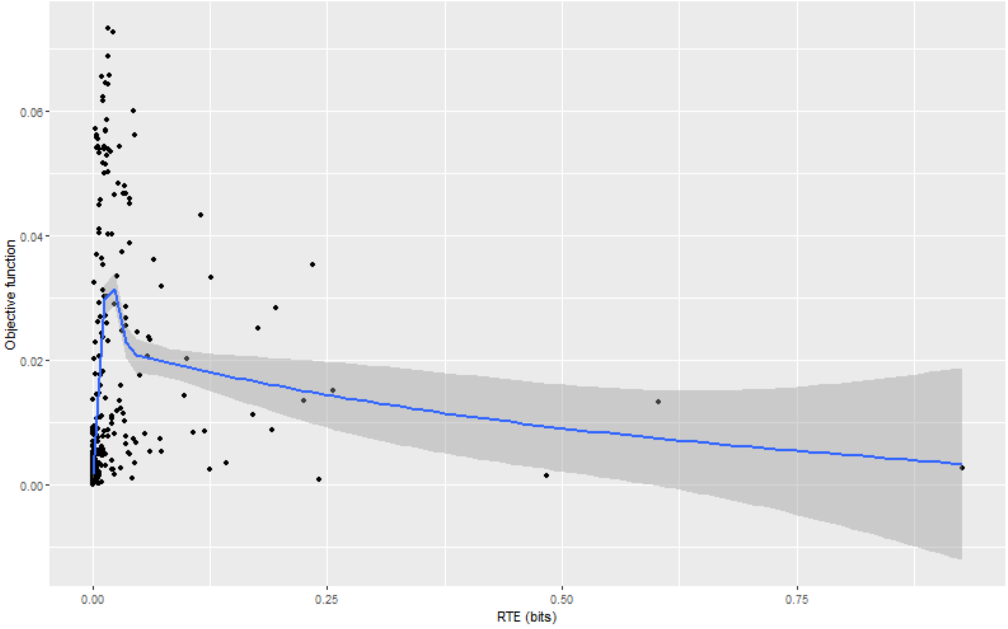
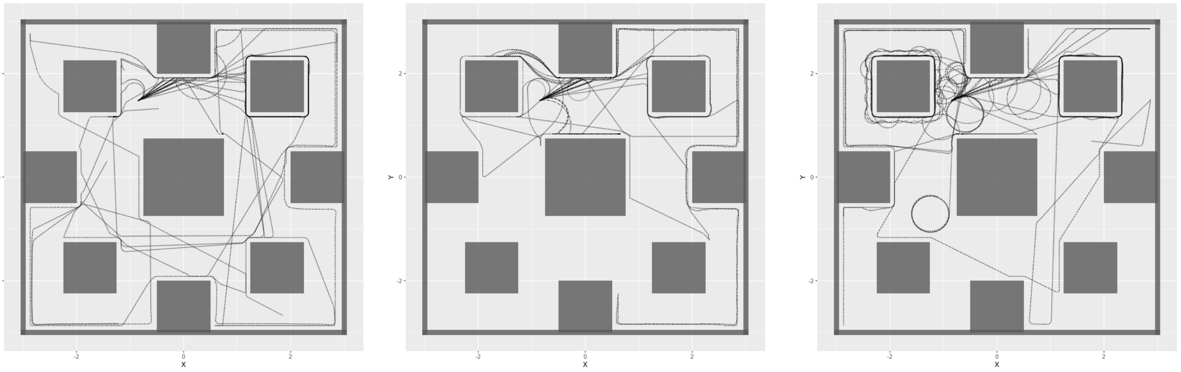


Figure 5.19: RTE



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.20: obstacle avoidance paths for top 20 robots with respect to RTE.

5.2 Path Following

5.2.1 Objective function

F_{pf} score distributions are summarized in figure 5.1. Like in the previous task RBNs generally perform worse than the random walk controller. This is due for the same reason as before: the majority of robots does not move or stop moving after few simulation steps. However we observe that the fittest robots according to the chosen objective function that use RBNs outperform the ones with the random walk controller. Top RBNs have F_{pf} higher than 4, while top random wolk robots have F_{pf} near 2. Through visual inspection robots that have a score equal or higher than 2.5 start to run across the path but with some inconsistency like loops. Robots with F_{pf} higher than 3 perform the task considerably well.

Computing the Wilcoxon's test on fitness values between RBNs with different nodes number we obtain the result in table 5.3. Accordingly to p-values using a confidence

Table 5.3: p-value results of Wilcoxon's test applied to F_{pf} score of networks grouped by nodes number in obstacle avoidance. Confidence is set equal to 0.99.

N value	**p-value**	**reject H_0**
20 - 50	1.4344e-17	yes
20 - 100	6.3114e-31	yes
50 - 100	0.0086	yes

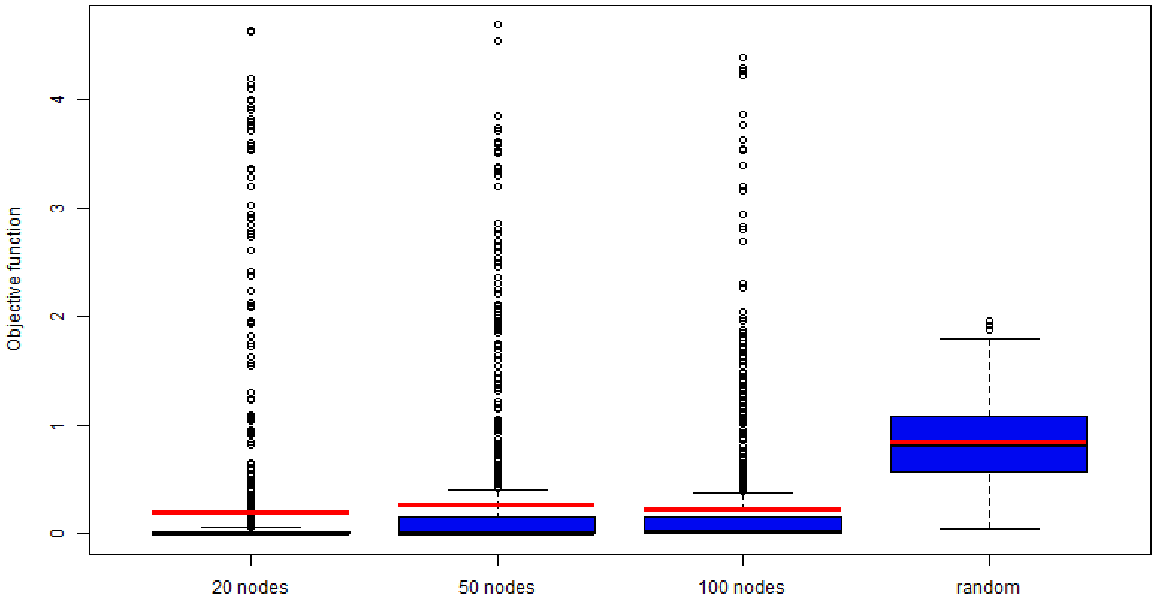
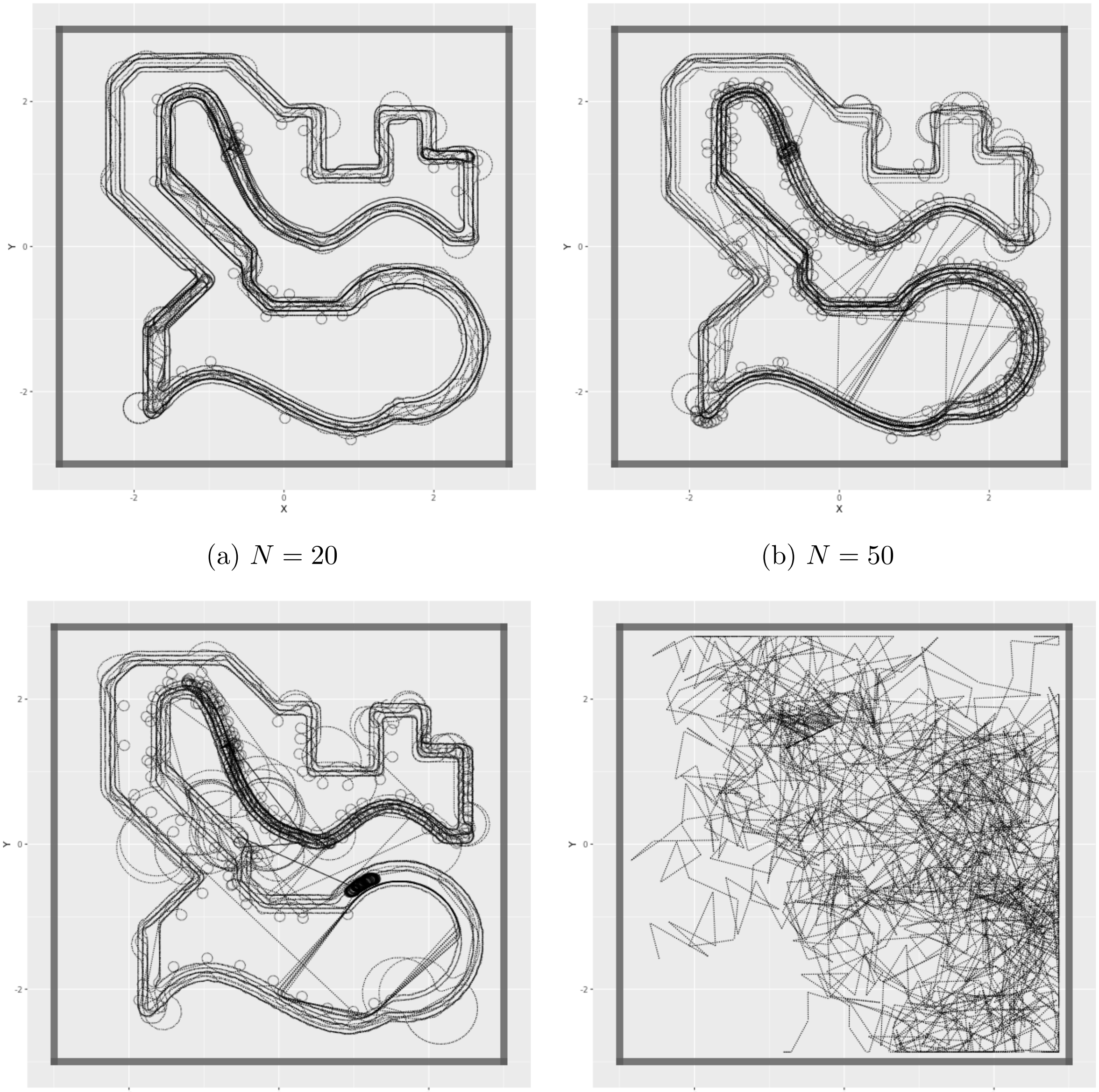


Figure 5.21: F_{pf} score distributions. From left to right distributions for: RBNs with $N = 20$, $N = 50$, $N = 100$ and random walk controller. Each distribution counts 1000 experiments.

level of 0.99 we have to reject the null hypothesis when we compare 20 and 50 nodes networks, 20 and 100 nodes networks and 50 and 100 nodes networks. This means that the number of nodes is a parameter that has statistically influence on the probability of getting a better RBNs for Path Following task, in this case with $N = 50$.

In figure 5.22 is reported a qualitative performance appearance of 20 robots that have obtained the highest scores per controller category. Top RBNs behave quite similar to each other (top RBNs with $N = 100$ are a bit worse), while top random walk robots do not achieve the objective. We observe that in RBNs paths there are some circles in correspondence of angles or wavy lines. After one or two circles robots settle and proceed over the path. Sometimes few robots cut through the circuit but ultimately when they reach it again continue to perform path following.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

(d) random walk

Figure 5.22: path following paths for top 20 robots with respect to F_{pf}.

5.2.2 Measures

Measures distribution are shown in figures 5.24 and 5.23. SE distributions have low medians but values reach almost 5 bits on a scale of 8. It is worth remember that it is

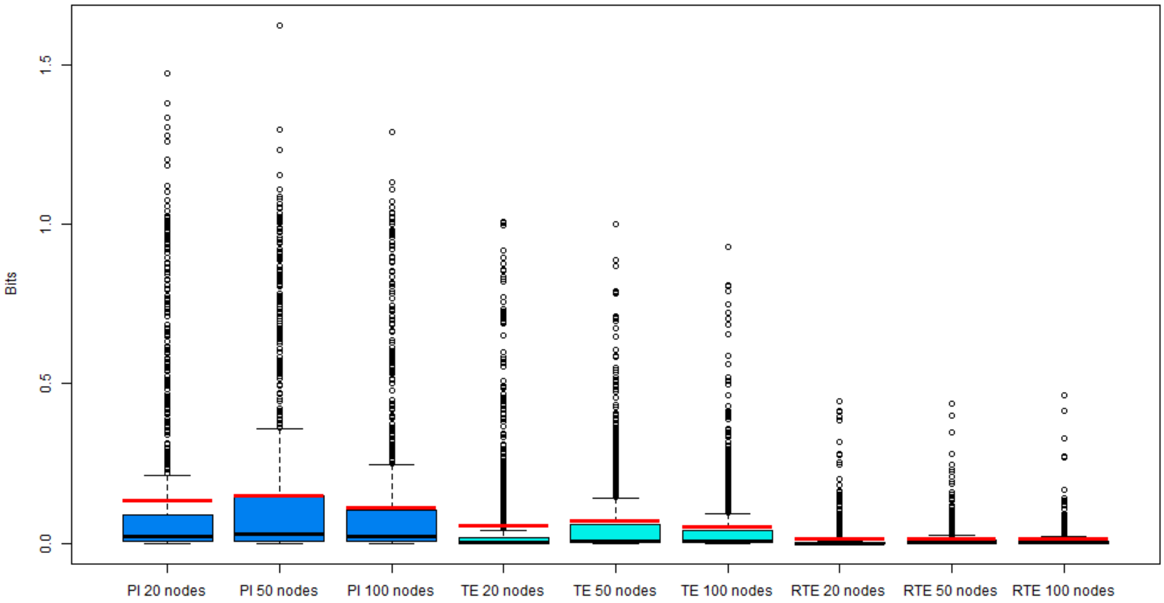


Figure 5.23: Measures score distributions without SE for path following. From left to right distributions for PI, TE and RTE. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

impossible to reach too high values due to the arena configuration. The low median is explained by robots that do not move or move only for few steps. Despite the task of obstacle avoidance the other measures distributions have higher means. This is due to the fact that robots are subjected to more different stimuli rather than in obstacle avoidance where a robot perceives only if it is close to an object and only in the object's direction. In path following a robot starts on the circuit and because the lines are generally not too wide and cover the nearby robot space there is an high variety of sensor values. Therefore robot's movement is not affected except from the external walls.

5.2.3 Correlation

Pearson's correlations values are reported in table 5.2.

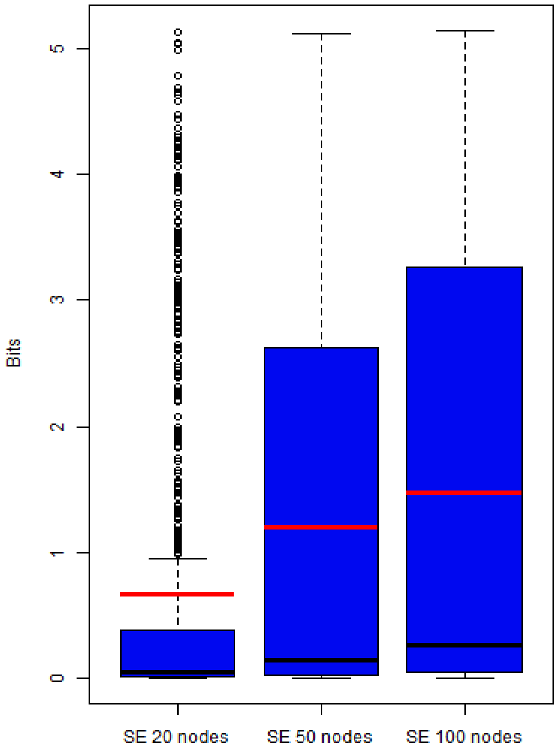


Figure 5.24: SE score distributions for path following. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

	F_{pf} - **SE**		F_{pf} - **PI**		F_{pf} - **TE**		F_{pf} - **RTE**	
	correlation	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**
20	0.5594	2.0326e-83	0.7148	3.3282e-157	0.8513	6.7828e-282	0.7717	2.078e-198
50	0.5089	5.4219e-67	0.735	1.2622e-170	0.7897	4.9862e-214	0.7749	4.5517e-201
100	0.4184	1.1752e-43	0.6807	4.2987e-137	0.7669	1.6546e-194	0.74	4.2484e-174

Table 5.4: Pearson's correlations between information theory measures and F_{pf}. Results are grouped by nodes number and each of them refers to 1000 experiments.

SE

SE unlike in the obstacle avoidance task is not the most correlated measure with F_{pf}. Indeed it is the less correlated, but correlation values are not too low, range is between 0.41 and 0.55. SE - F_{pf} plots are shown in figures 5.25, 5.26 and 5.27. From 0 to 2 bits F_{pf} increases, however there are several robots that have high SE and objective

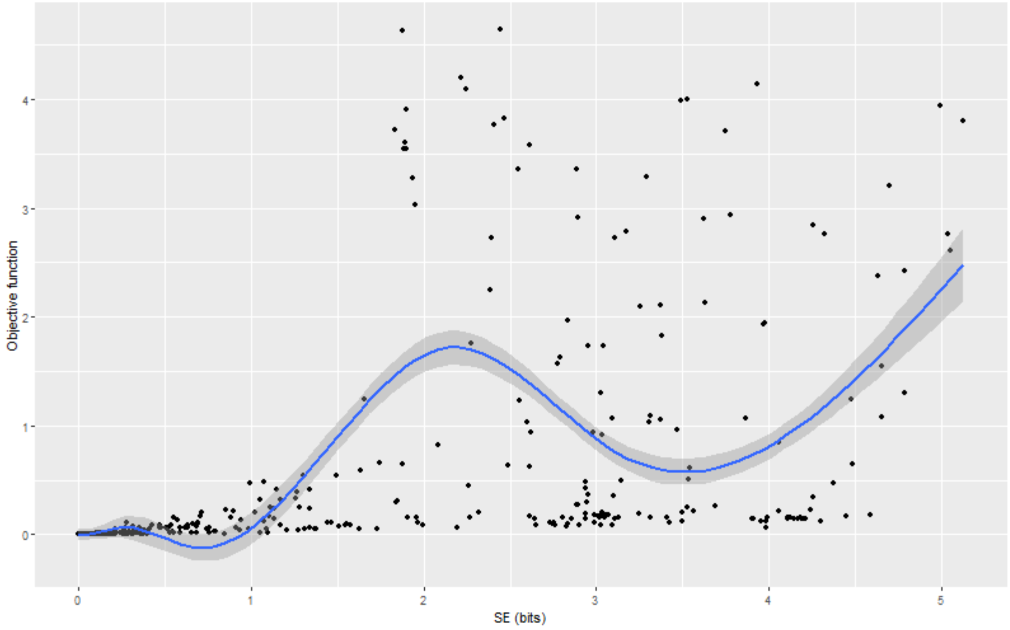


Figure 5.25: SE - F_{pf} plot for networks with $N = 20$.

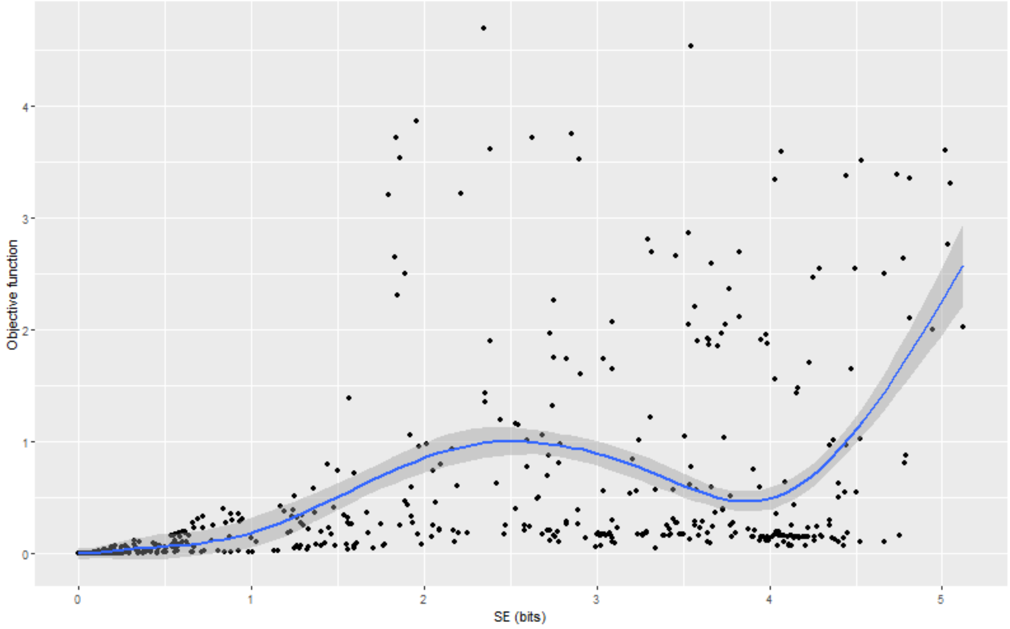


Figure 5.26: SE - F_{pf} plot for networks with $N = 50$.

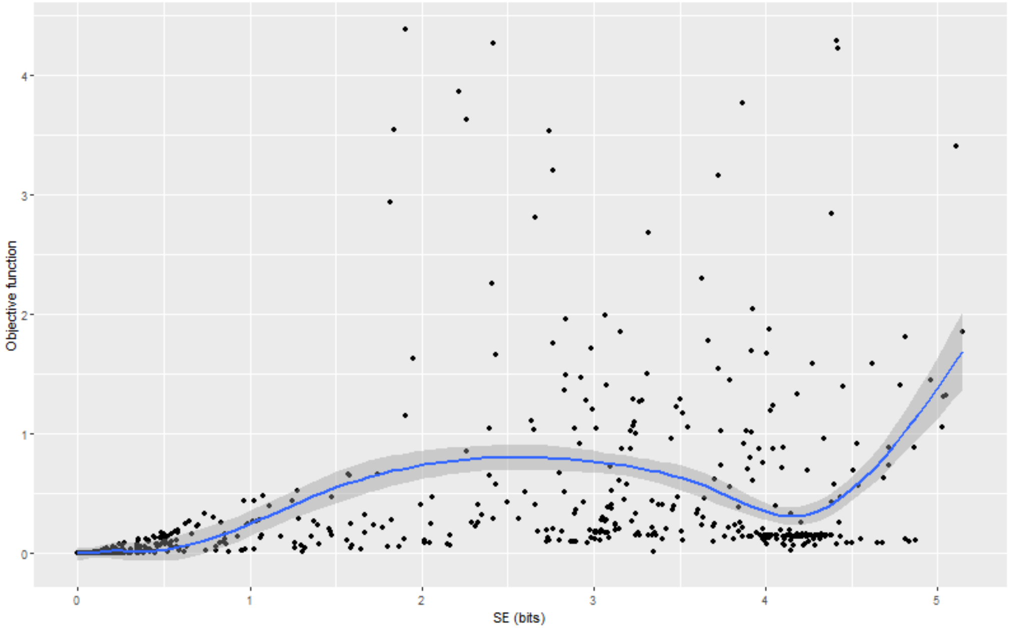
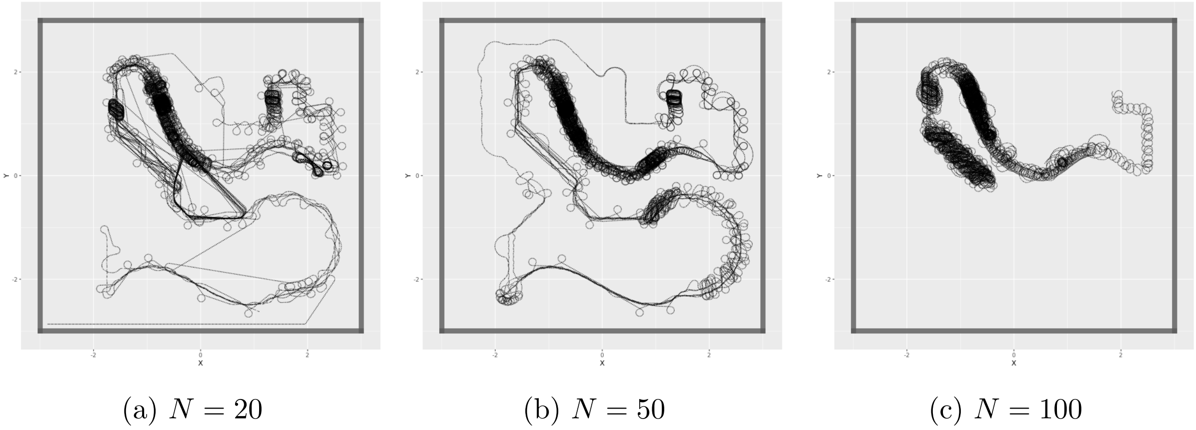


Figure 5.27: SE - F_{pf} plot for networks with $N = 100$.

function still near to 0. From a theoretical prospective to maximize SE a robot has to perceive different sensor values with equal probabilities. This behaviour is achieved by robots that move in circle around the path. Paths of the top 20 robots with respect to SE are reported in figure 5.28. We observe robots performing roto-translation over the



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

Figure 5.28: path following paths for top 20 robots with respect to SE.

path corresponding to darker areas or rings in the plots.

After this deeper investigation we assert that despite SE has a moderate positive value of correlation with the chosen objective function robots with the highest SE do not perform well in path following task.

PI

PI has correlation values between 0.68 and 0.73 with F_{pf}. This indicates that the metric is positively correlated with the chosen objective function. PI - F_{pf} plots are shown in figures 5.29, 5.30 and 5.31. We notice that generally robots with higher PI have higher score, however a notable number of robots with medium or high metric have medium or low score. Through visual inspection we have found that for these robots F_{pf} is not high because they run the same section path more time back and forth. So they do not reach far positions from the starting one and consequentially do not get high score. High value of correlation for predictive information is explainable by the fact that a robot, to stay on the path while moving, must have its actuators well coupled with its sensors. The RBN that controls the robot “recognizes” when to turn right or left based on the sensor values. Otherwise a robot cannot follow the path and gets a low score.

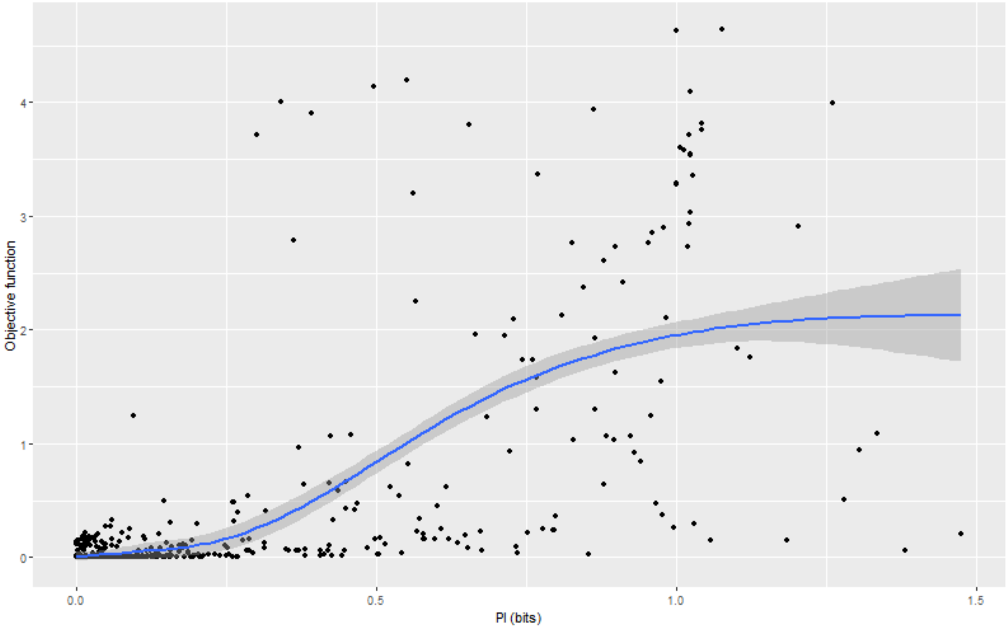


Figure 5.29: PI - F_{pf} plot for networks with $N = 20$.

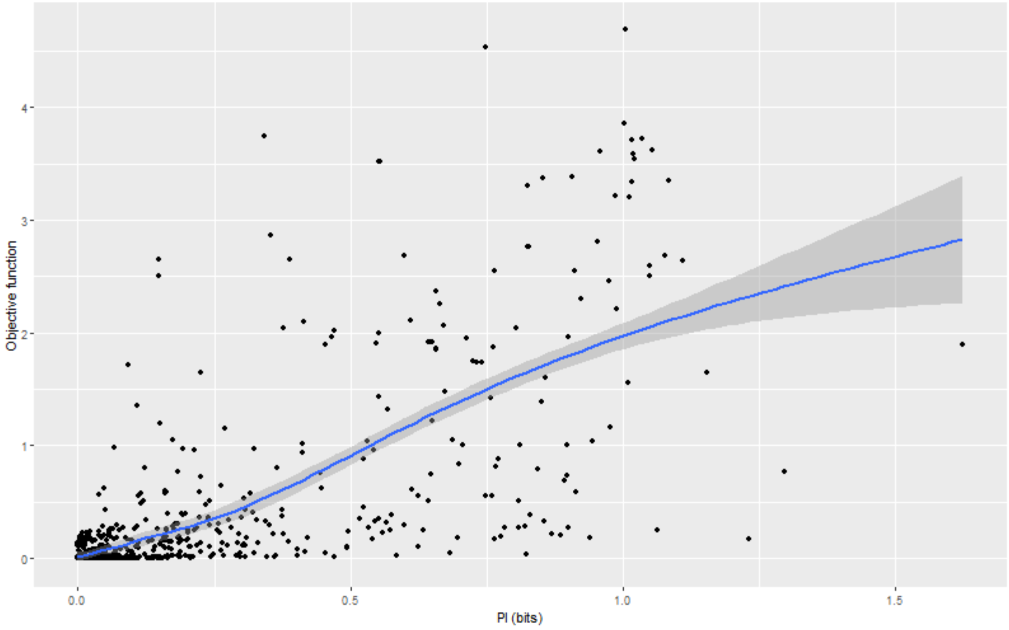


Figure 5.30: PI - F_{pf} plot for networks with $N = 50$.

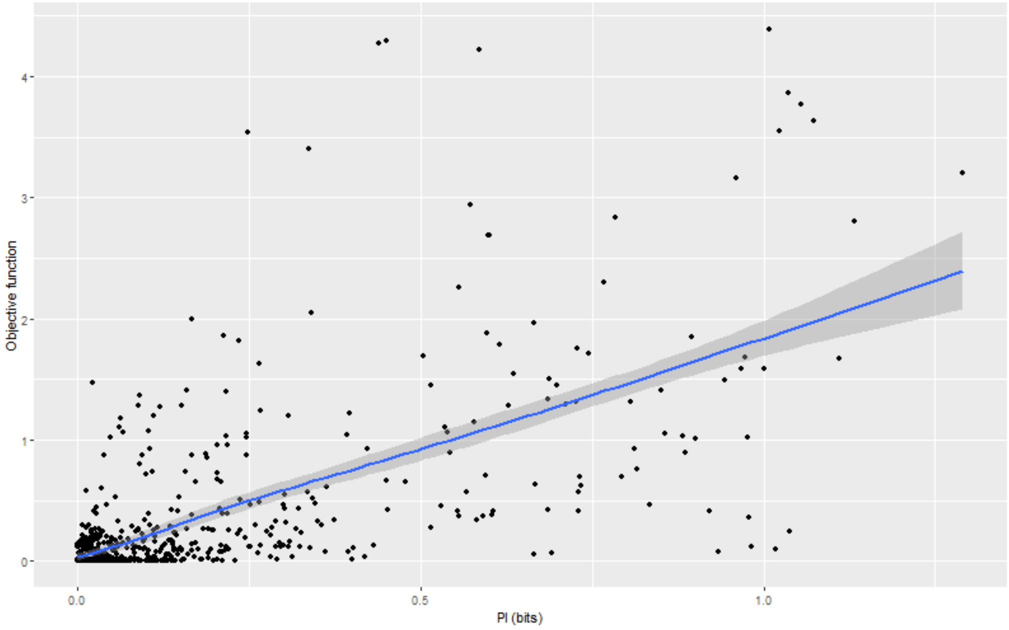
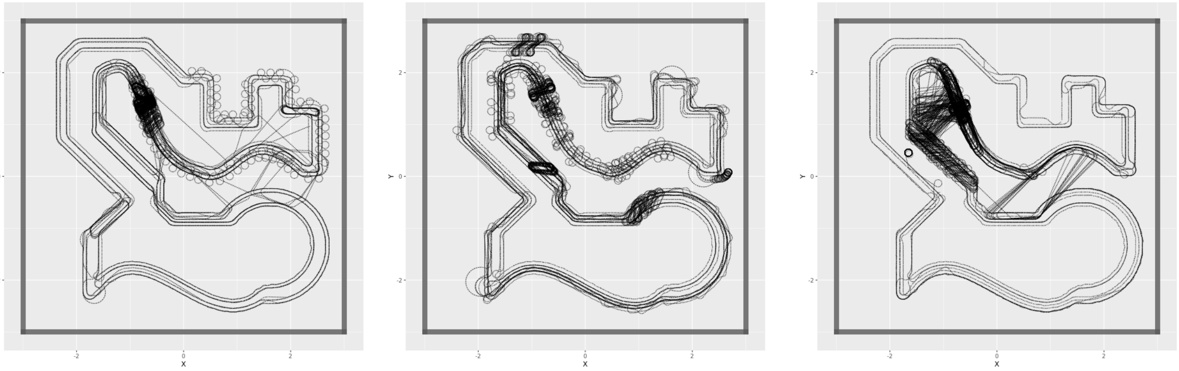


Figure 5.31: PI - F_{pf} plot for networks with $N = 100$.

Robots that instead have low PI and medium or high F_{pf} have less smooth movements with swings while running a straight line. Paths of the top 20 robots with respect to SE are reported in figure 5.32.

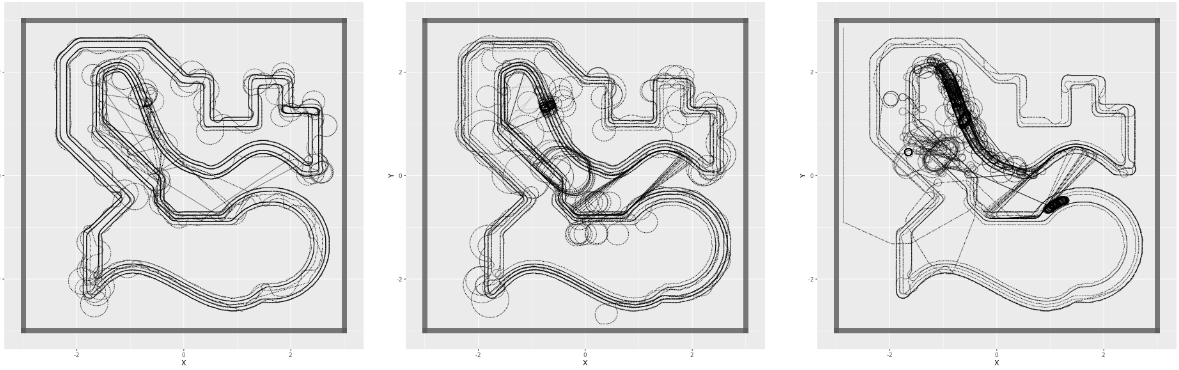


(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.32: path following paths for top 20 robots with respect to PI.

TE

TE is strongly positively correlated with F_{pf}. Correlation goes from 0.76 to 0.85 and it is the highest of the four. TE - F_{pf} plots are shown in figures 5.34, 5.35 and 5.36.



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.33: path following paths for top 20 robots with respect to TE.

Score tends to grow linearly with the metric. Compared to PI there are less robots

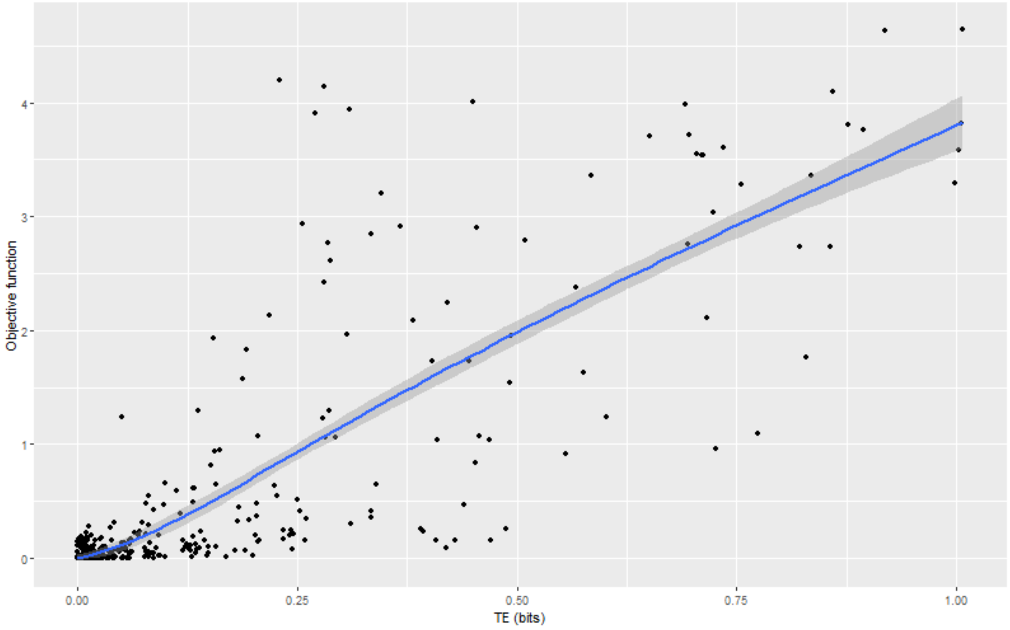


Figure 5.34: TE - F_{pf} plot for networks with $N = 20$.

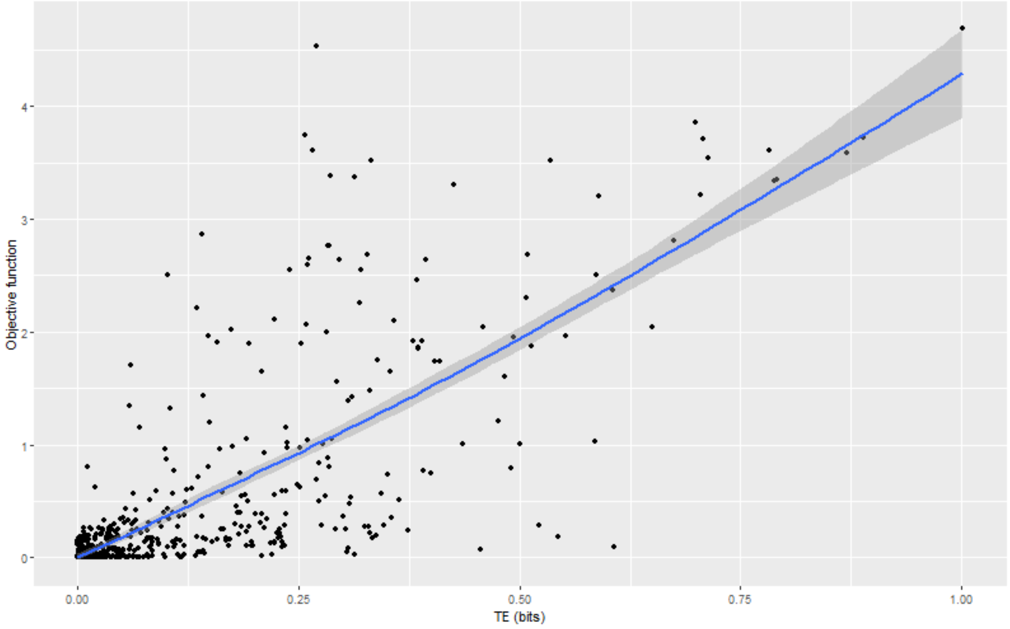


Figure 5.35: TE - F_{pf} plot for networks with $N = 50$.

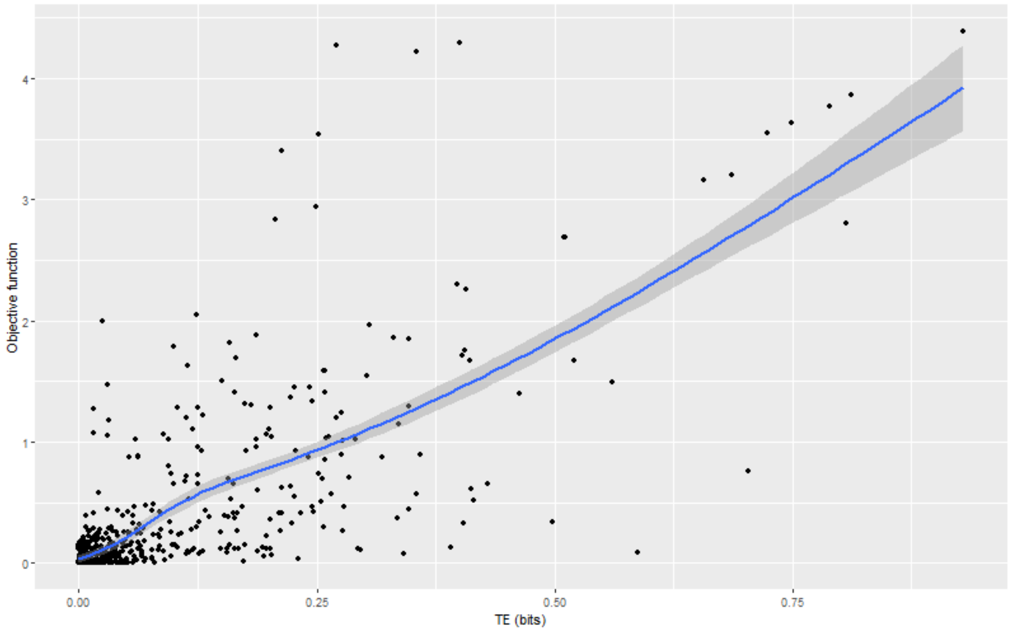


Figure 5.36: TE - F_{pf} plot for networks with $N = 100$.

with medium or high TE and medium or low score. The ones that still persist can be explained with the same reason: robots do not reach far positions. Also robots that have medium TE and high F_{pf} behave like the ones in PI. Paths of the top 20 robots with respect to TE are reported in figure 5.33. We notice a clear similarity with figure 5.22. In this case there are a bit more circles, especially in proximity of tight curves of the circuit.

After this deeper analysis we assert that transfer entropy from sensors to actuators is strongly positively correlated with the chosen objective function for the task of path following. TE can definitely be used as an alternative objective function that is not handcrafted by humans.

RTE

RTE is strongly positively correlated with F_{pf} and has similar values of TE, it goes from 0.74 to 0.77. RTE - F_{pf} plots are shown in figures 5.37, 5.38 and 5.39. We observe a solid logarithmic grow of the objective function. There are no robots with high RTE and low score, robots with more than 0.2 bits are all suitable for path following. Paths of

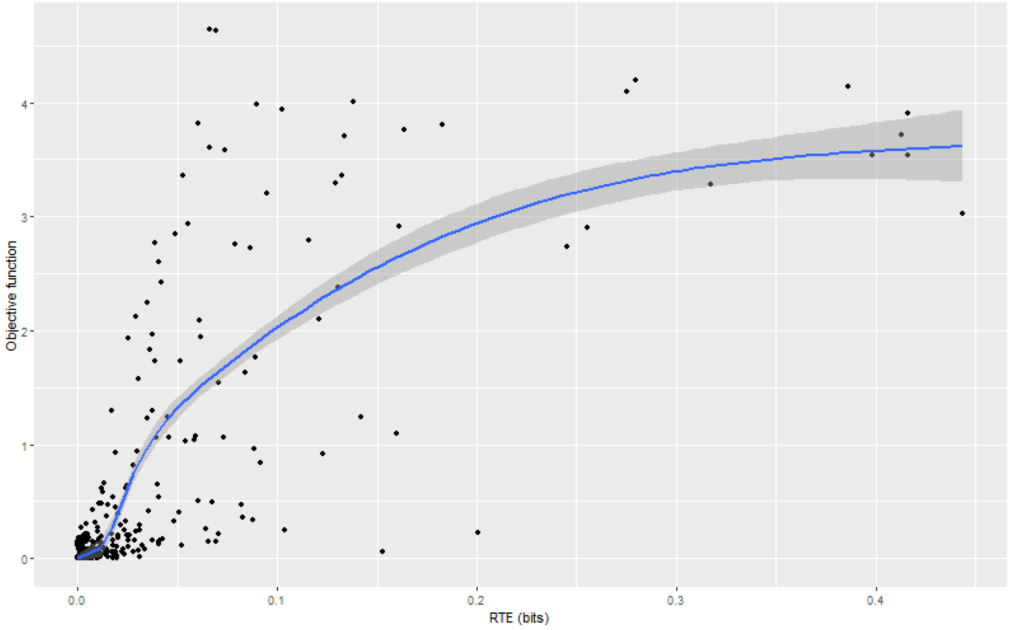


Figure 5.37: RTE - F_{pf} plot for networks with $N = 20$.

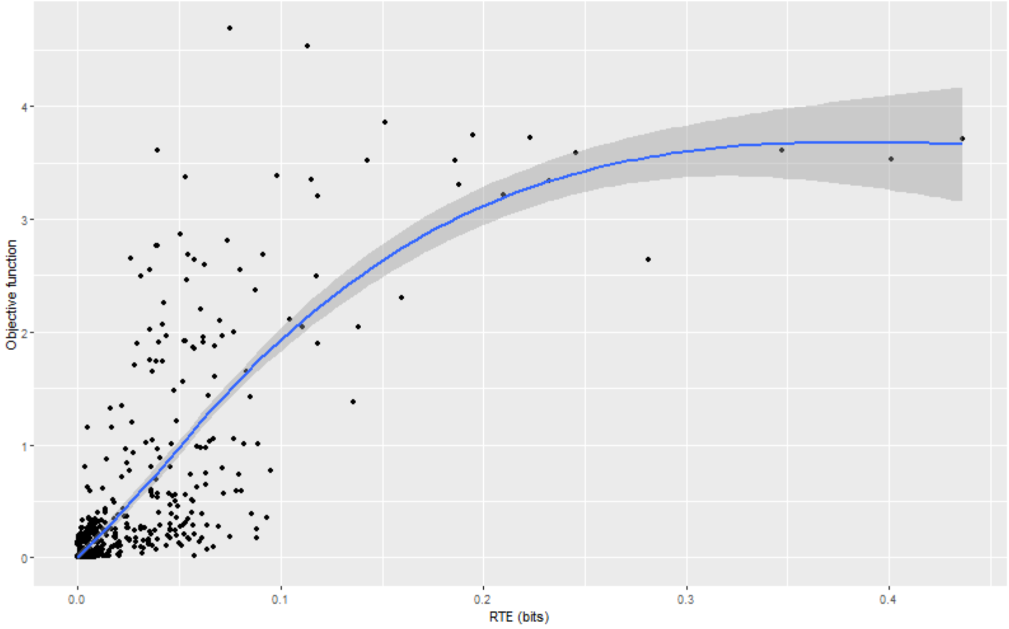


Figure 5.38: RTE - F_{pf} plot for networks with $N = 50$.

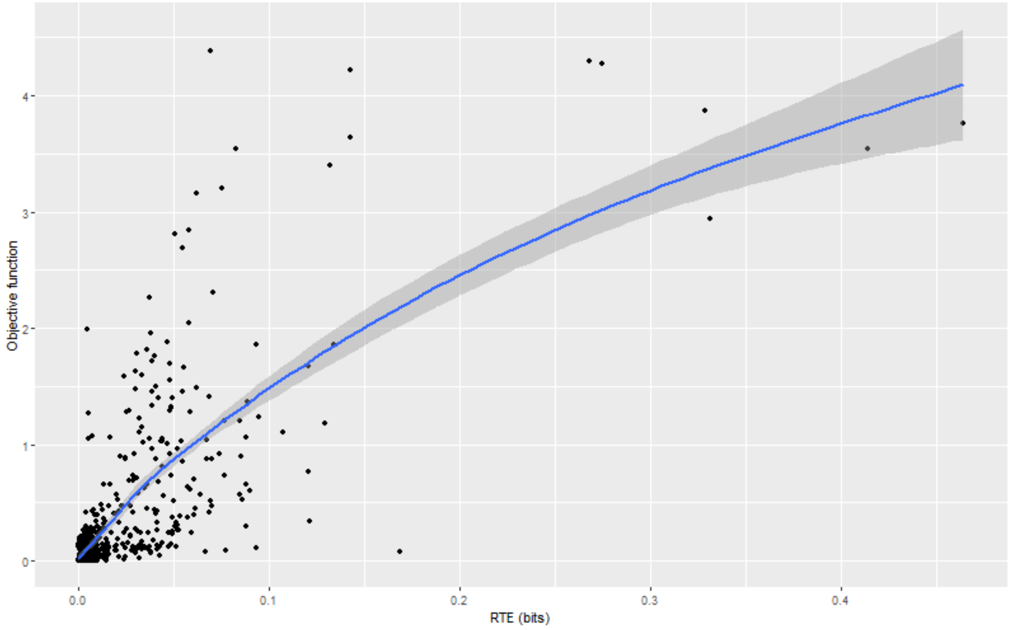
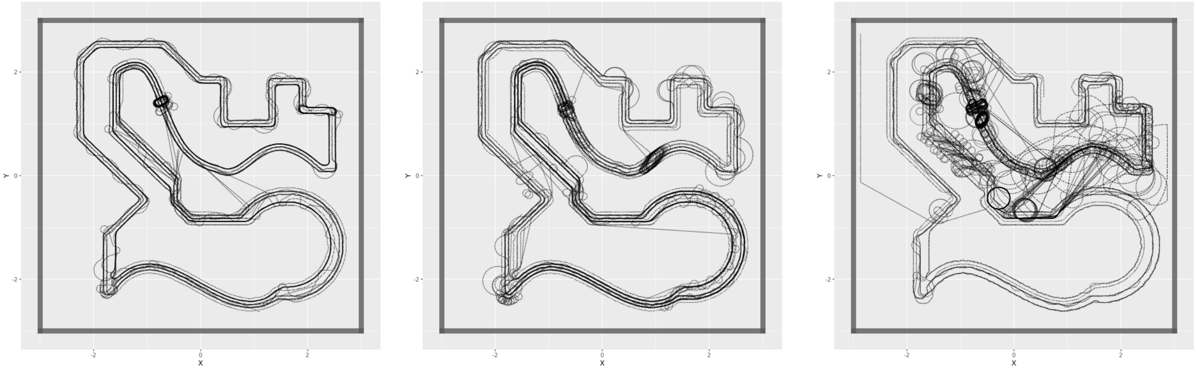


Figure 5.39: RTE - F_{pf} plot for networks with $N = 100$.

the top 20 robots with respect to RTE are reported in figure 5.40. Paths are extremely close to the ones in figure 5.22.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

Figure 5.40: path following paths for top 20 robots with respect to RTE.

Ultimately the transfer entropy from actuators to sensors is a metric strongly positively correlated with the chosen objective function. It can be used as an alternative not handcrafted objective function for the path following task.

5.3 Phototaxis

5.3.1 Objective function

Fpt score distributions are summarized in figure 5.41. Differently from the previous tasks, the chosen objective function for phototaxis is not limited by 0. A standing robot get almost a score of 0.2727. A robot that moves away from the light source gets a lower value. We notice that both RBNs and random walk controller have similar means and medians. However the most suitable RBNs outperform the best random walk robots. With F_{pt} greater than 0.4 robots move towards the light source but not in the fastest way. Not all robots with score close to 0.4 ends nearby the light source, but they definitely reduce the distance. Robots that get a score higher than 0.45 end in close proximity of

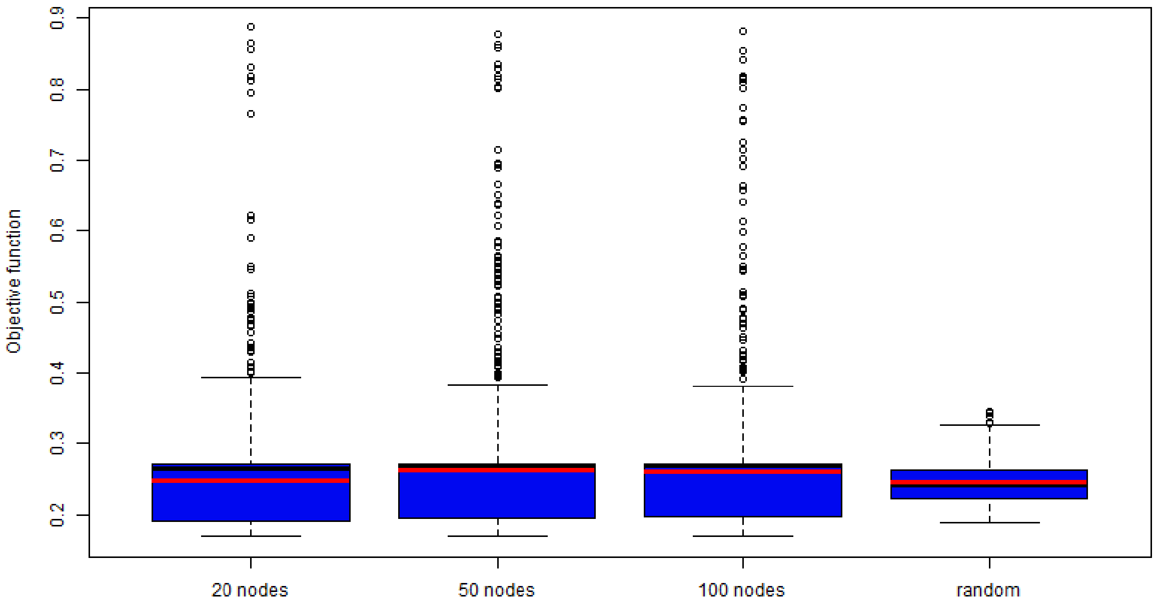
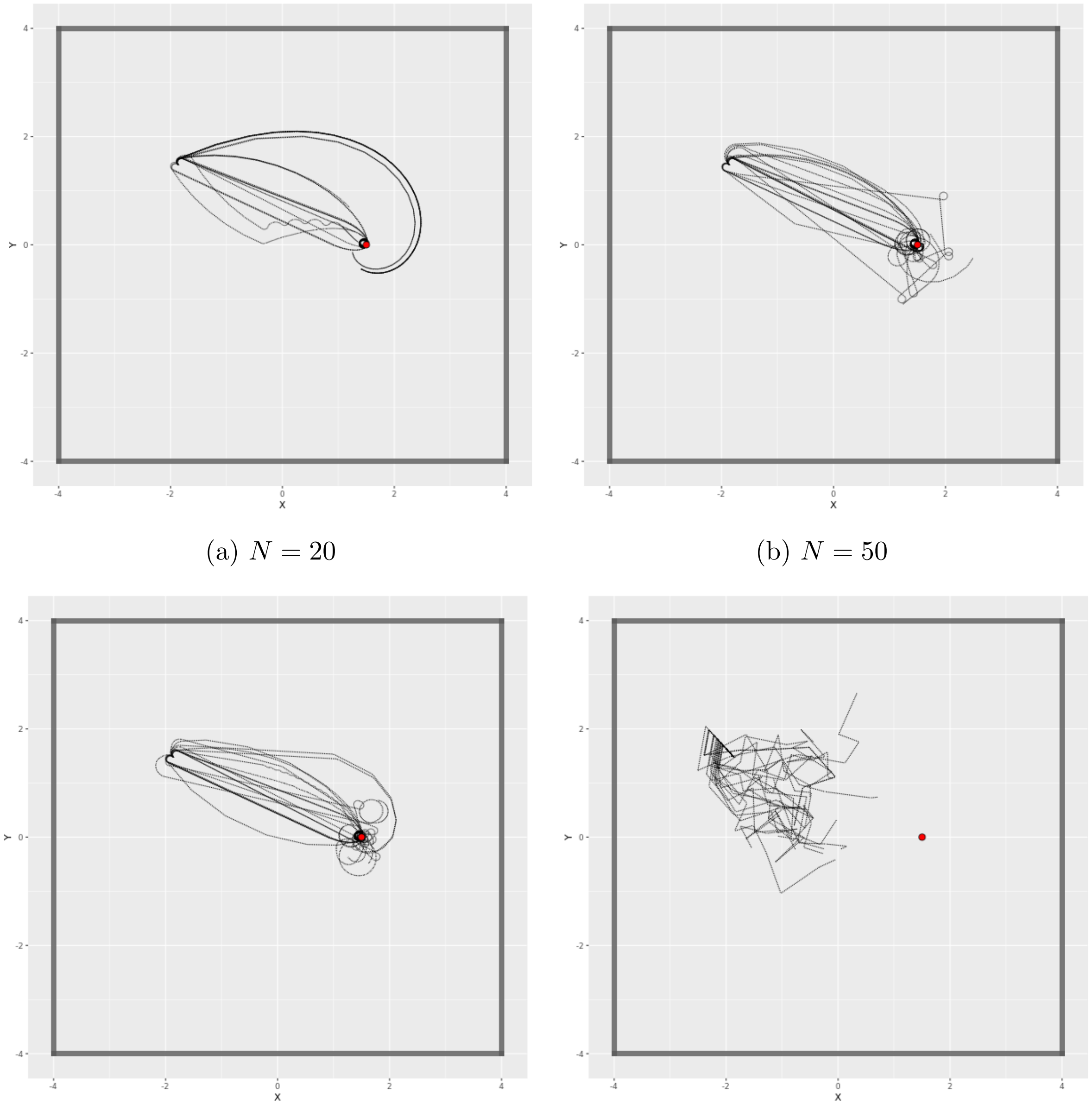


Figure 5.41: F_{pt} score distributions. From left to right distributions for: RBNs with $N = 20$, $N = 50$, $N = 100$ and random walk controller. Each distribution counts 1000 experiments.

the light source. With more than 0.5 robots go straight forward towards the light source.



(a) $N = 20$

(b) $N = 50$

(c) $N = 100$

(d) random walk

Figure 5.42: phototaxis paths for top 20 robots with respect to F_{pt}.

Computing the Wilcoxon's test on fitness values between RBNs with different nodes number we obtain the result in table 5.5. Distributions are quite similar to each other however RBNs with $N = 100$ are statistically different to RBNs with $N = 20$. It is more probable to get a RBN with higher score with $N = 100$ compared to robot a RBN $N = 20$ with a confidence of 0.99.

Table 5.5: p-value results of Wilcoxon's test applied to F_{pt} score of networks grouped by nodes number in obstacle avoidance. Confidence is set equal to 0.99.

N value	**p-value**	**reject H_0**
20 - 50	0.0546	no
20 - 100	6.5049e-5	yes
50 - 100	0.0328	no

In figure 5.42 is reported a qualitative performance appearance of 20 robots that have obtained the highest scores per controller category. The red dot corresponds to the light source, position 1.5,0,0. We observe that the fittest robots according to the chosen objective function reach the light source or loop around it in its proximity. Clearly we do not expect such good behaviour from the best random walk controller. These top RBN behaviours are divided into at least three different types. The first one is turning immediately towards the light source and reach it (see plots a, b and c). The second is to approach the light source more slowly in a spiral (see plots a and c). The third one is similar to the first but in proximity of the light source the robot start moving in circles (see plots b and c).

5.3.2 Measures

Measures distribution are shown in figures 5.43 and 5.44. SE distributions have means and medians between 1 and 2 bits. Te highest values reach almost 4 bits on a scale of 8. It is worth reminding that even for this task it is impossible to reach too high values due to the arena configuration. The other measures distributions have higher means compared with the previous two tasks. This happens because of the high sensor values variety. When a robot moves tangentially to the light cone, turns around or moves in circle its sensors perceive lots of different values. Therefore robot's movement is not affected except from the external walls.

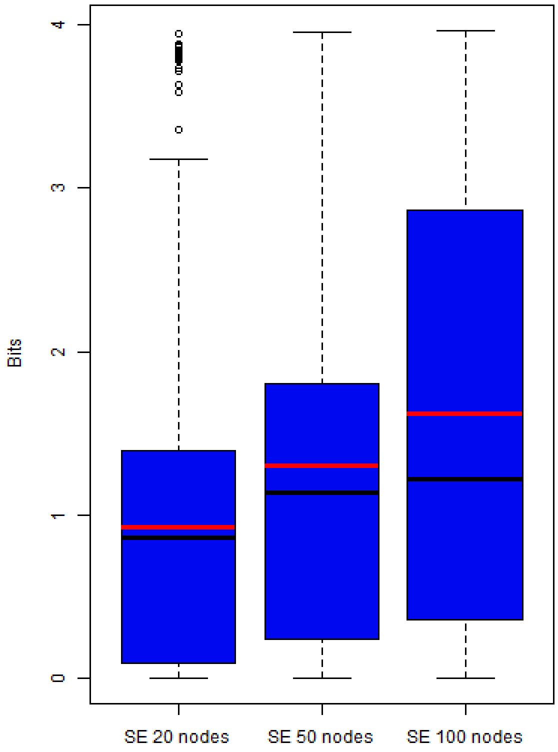


Figure 5.43: SE score distributions for phototaxis. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

5.3.3 Correlation

Phototaxis is a much different task compared to obstacle avoidance and path following. In the previous two there is a clear subspace of the arena where robot receives stimuli (near obstacles and on the path). In this subspace the robot has to behave in a proper manner to perform well the assigned task. In phototaxis instead the arena is almost entirely lighted by a light source and a robot does not recognize its intensity but only if there is light or not. Nobody has told the robot to prefer moving towards the source of these stimuli or moving away or around it. We expect much lower values of correlations between information theory measures and the chosen objective function. Pearson's correlations values are reported in table 5.6.

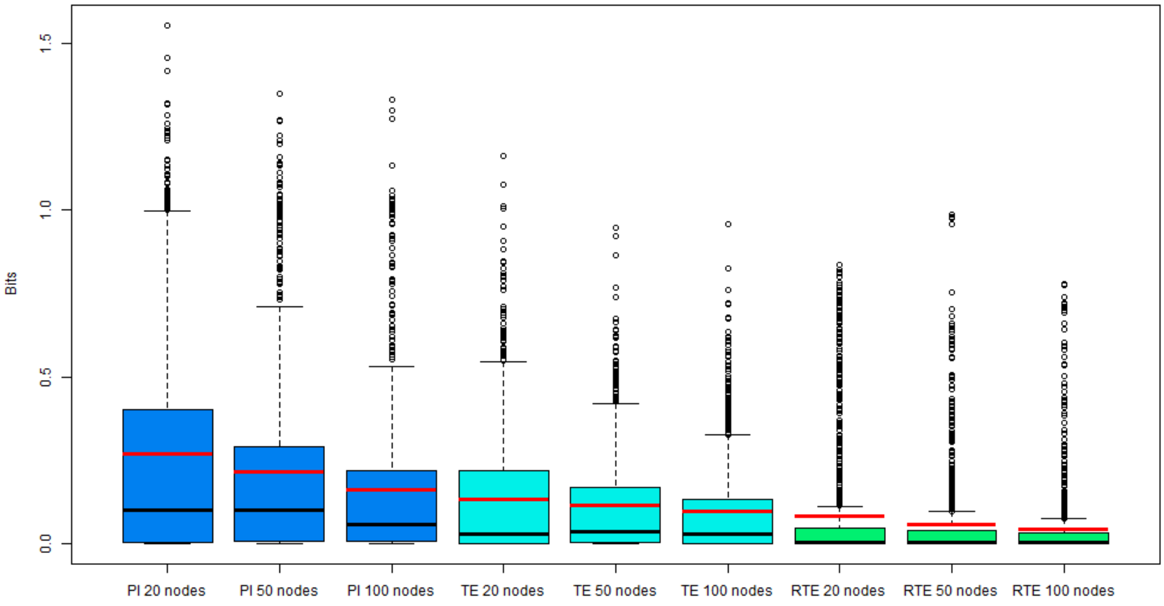


Figure 5.44: Measures score distributions without SE for phototaxis. From left to right distributions for PI, TE and RTE. Distributions are grouped by node number: $N = 20$, $N = 50$, $N = 100$. Each distribution counts 1000 experiments.

	F_{pt} - **SE**		F_{pt} - **PI**		F_{pt} - **TE**		F_{pt} - **RTE**	
	correlation	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**	**correlation**	**p-value**
20	0.0394	0.2122	0.2223	1.15e-12	0.2023	1.0552e-10	0.0434	0.1701
50	0.1949	5.0346e-10	0.3532	9.3298e-31	0.2722	1.8886e-18	0.0674	0.0329
100	0.1914	1.044e-09	0.4321	9.1552e-47	0.2781	3.1472e-19	0.1287	4.4642e-05

Table 5.6: Pearson's correlations between information theory measures and F_{pt}. Results are grouped by nodes number and each of them refers to 1000 experiments.

SE

SE is not positively nor negatively correlated with F_{pt}. Correlation values are between 0.03 and 0.19. SE - F_{pt} plots are shown in figures 5.45, 5.46 and 5.47. From 0 to 2 bits of SE, F_{pt} is quite inconsistent, while from 2 to 3 bits the objective tends to increase. Robots with SE greater than 3 bits have F_{pt} of almost 0.27. Robots that stay on the

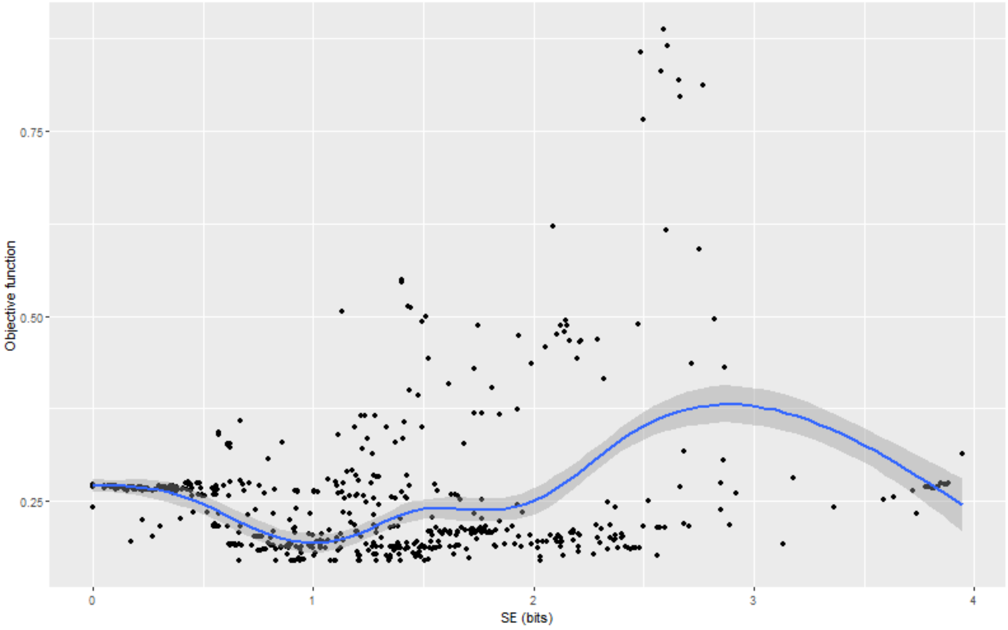


Figure 5.45: SE - F_{pt} plot for networks with $N = 20$.

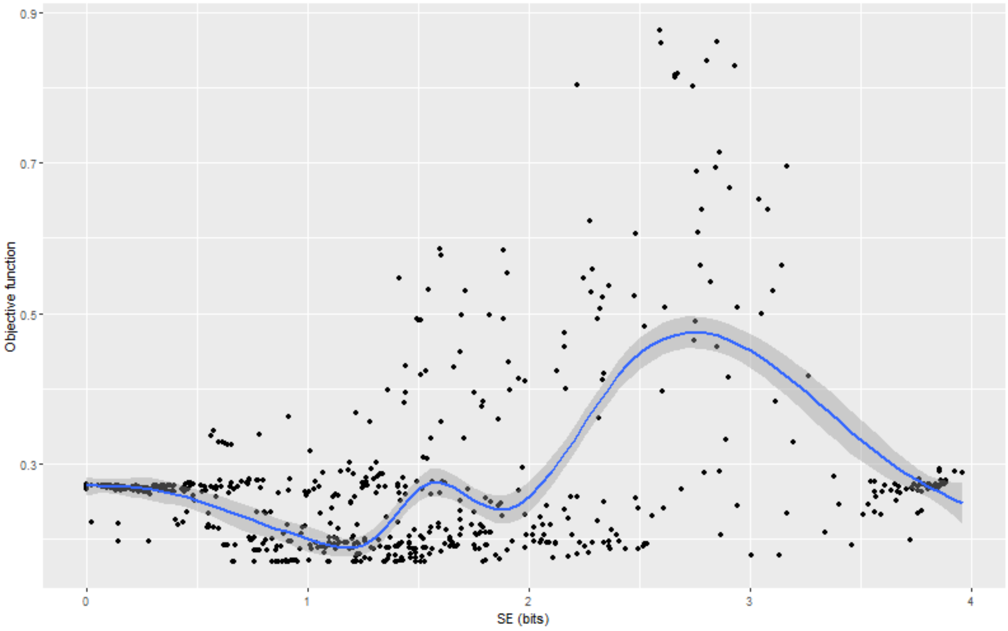


Figure 5.46: SE - F_{pt} plot for networks with $N = 50$.

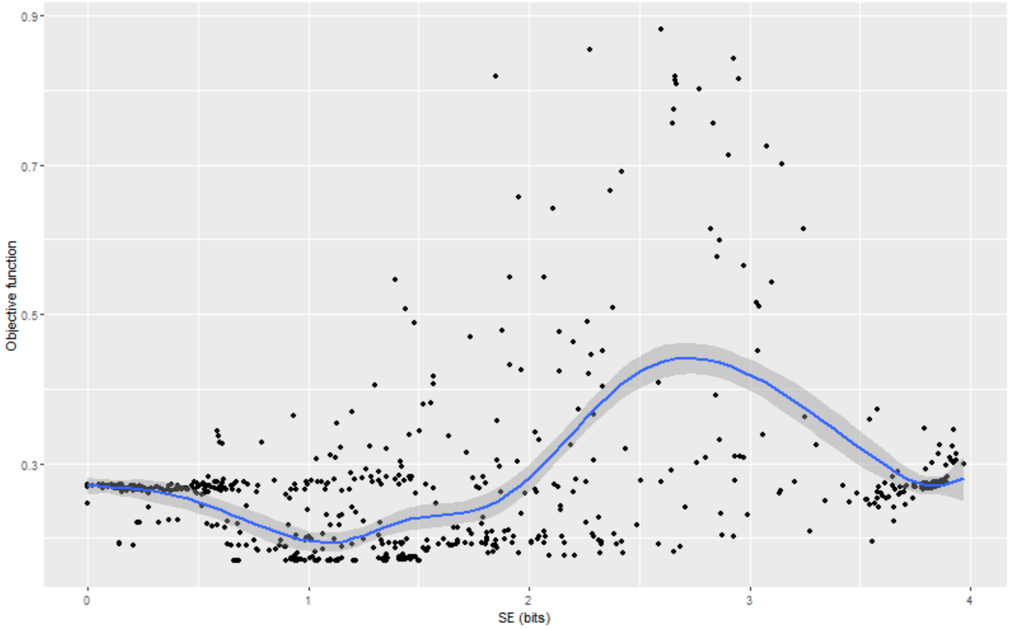
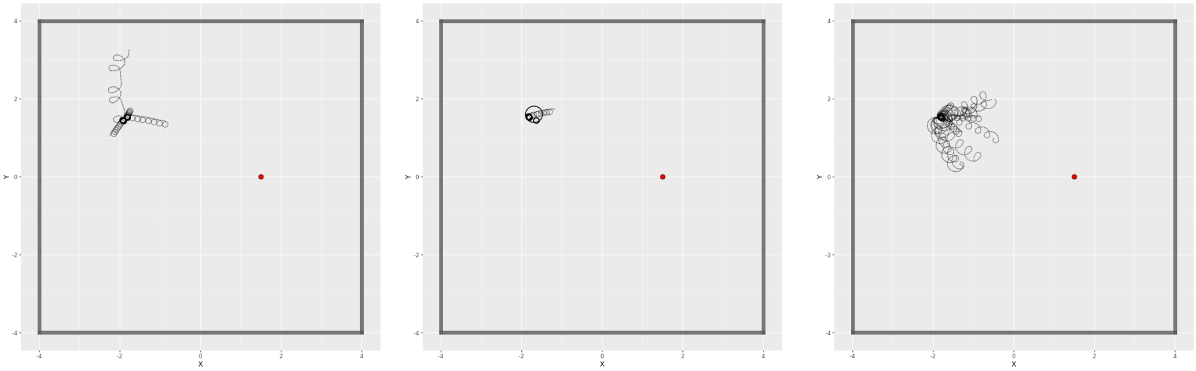


Figure 5.47: SE - F_{pt} plot for networks with $N = 100$.

line of $F_{pt} \approx 0.27$ do not move too away from the starting position. Dots under that thresholds correspond to robots that perform anti phototaxis. These results are expected because to maximize SE a robot should constantly move in circle and to do so it cannot run far away from the starting position and approach the light source.

Paths of the top 20 robots with respect to SE are reported in figure 5.48. From



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.48: phototaxis paths for top 20 robots with respect to SE.

the figure we have confirmation of the previous reasoning. Robots move in circle or roto-translate a bit away from the starting position.

PI

PI has correlation values with F_{pt} between 0.22 and 0.43. So only in the case of $N = 100$ there is a moderate positive correlation. PI - F_{pt} plots are shown in figures 5.49, 5.50 and 5.51. From plots we observe that there is no clear relation between predictive information and the chosen objective function. Only with $N = 100$ when robots get very high PI then they also get high score but could likely be a chance given the low number of these robots. Predictive information is high when the information shared between sensors and actuators is high, qualitative speaking when in the presence of the same sensor stimuli the robot behave always in the same way. The issue is that this happens both when the robot approaches the light source and when it run away.

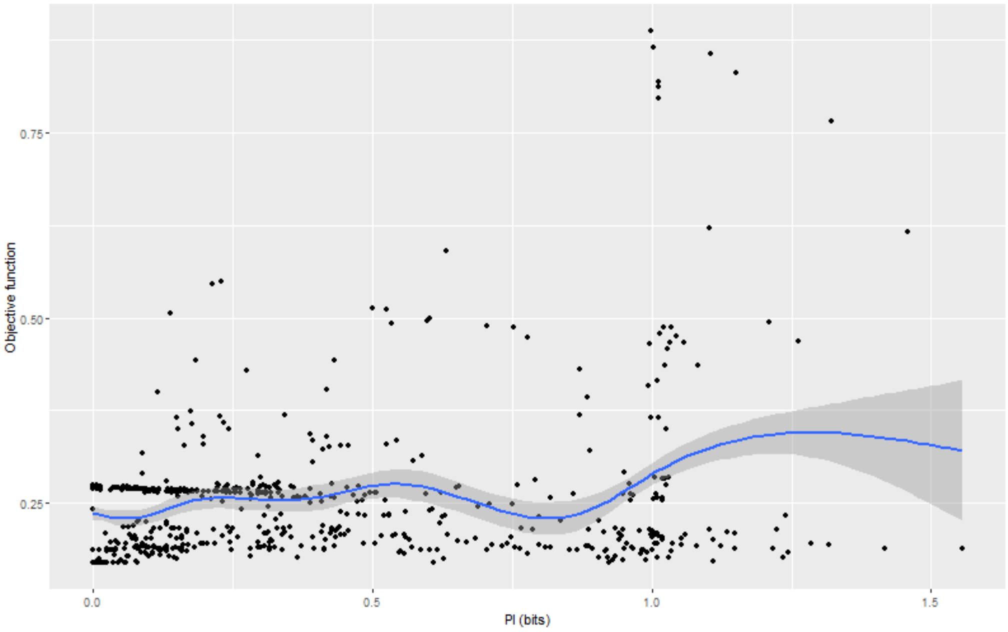


Figure 5.49: PI - F_{pt} plot for networks with $N = 20$.

Paths of the top 20 robots with respect to PI are reported in figure 5.52. We notice that robots are divided in two types: robots that actually perform phototaxis and

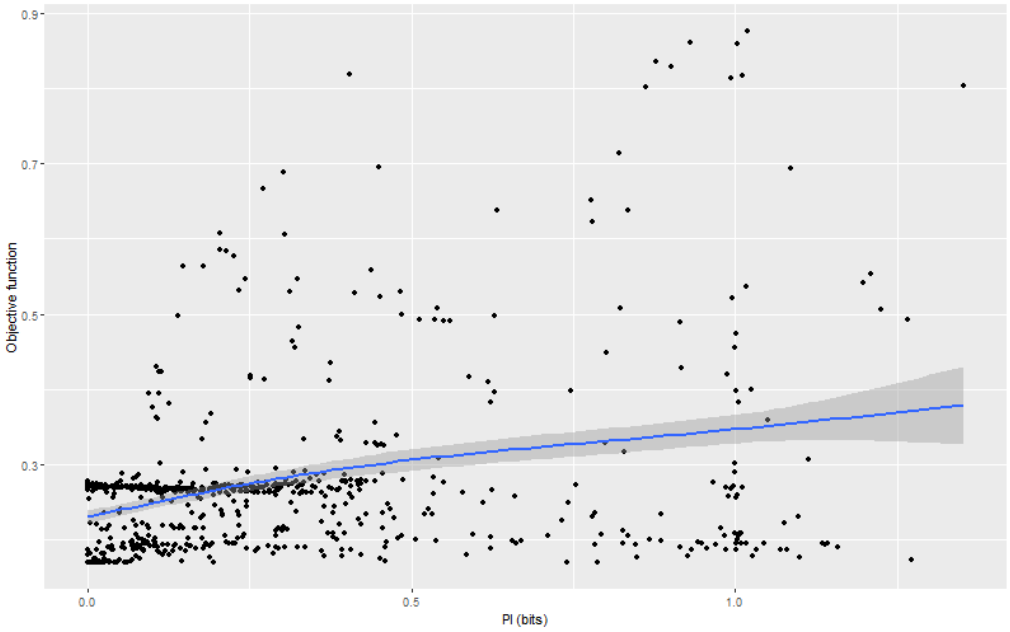


Figure 5.50: PI - F_{pt} plot for networks with $N = 50$.

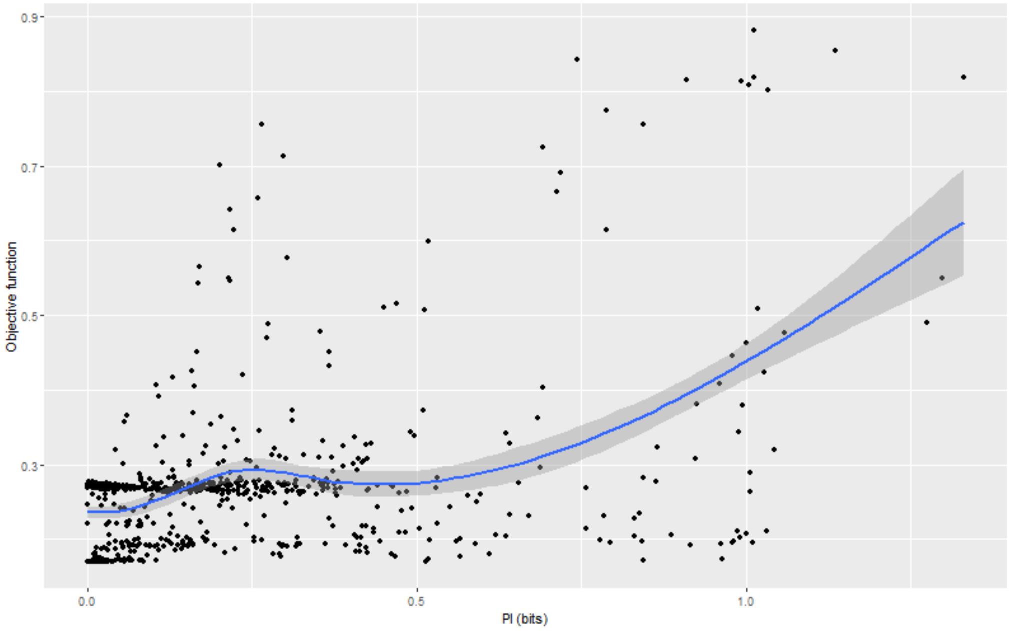
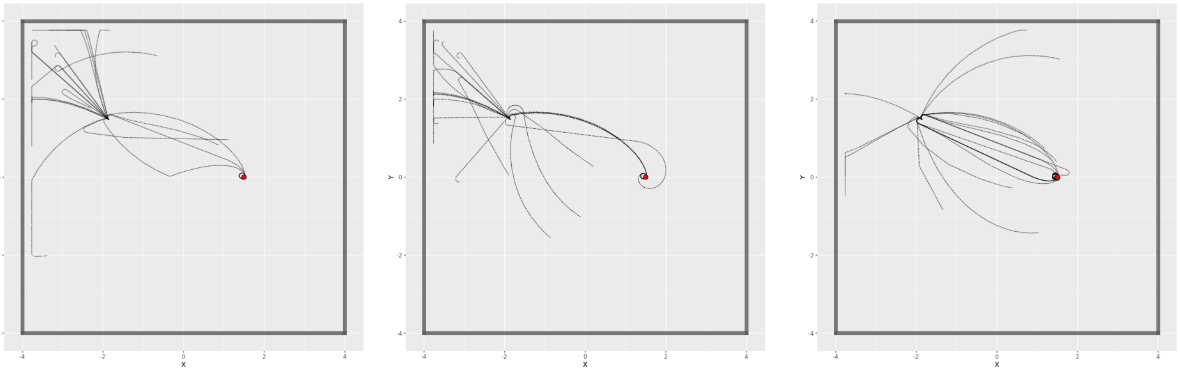


Figure 5.51: PI - F_{pt} plot for networks with $N = 100$.



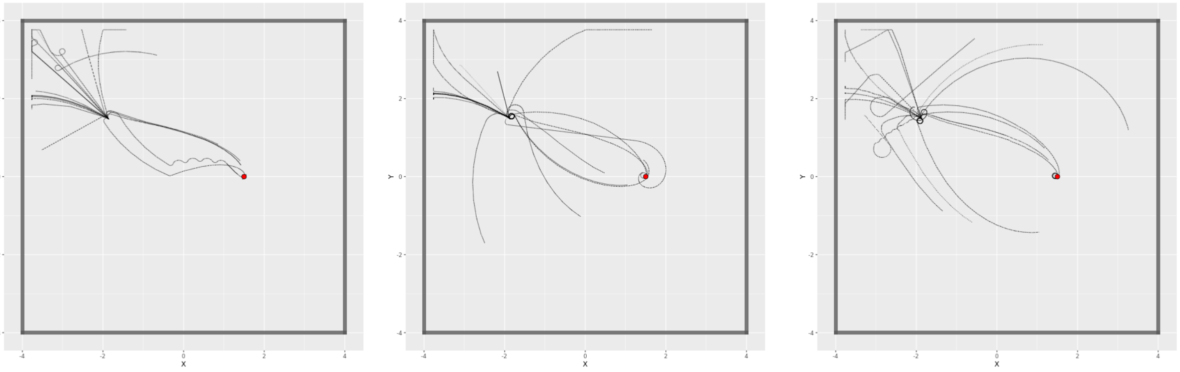
(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.52: phototaxis paths for top 20 robots with respect to PI.

robots that perform anti phototaxis. So, despite having higher correlation than SE, PI is definitely not correlated with F_{pt}.

TE

TE correlation with F_{pt} has similar values to correlation between PI and F_{pt}. Values are between 0.20 and 0.27, which suggest a weak moderate correlation with the chosen objective function. TE - F_{pt} plots are shown in figures 5.54, 5.55 and 5.56.



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.53: phototaxis paths for top 20 robots with respect to TE.

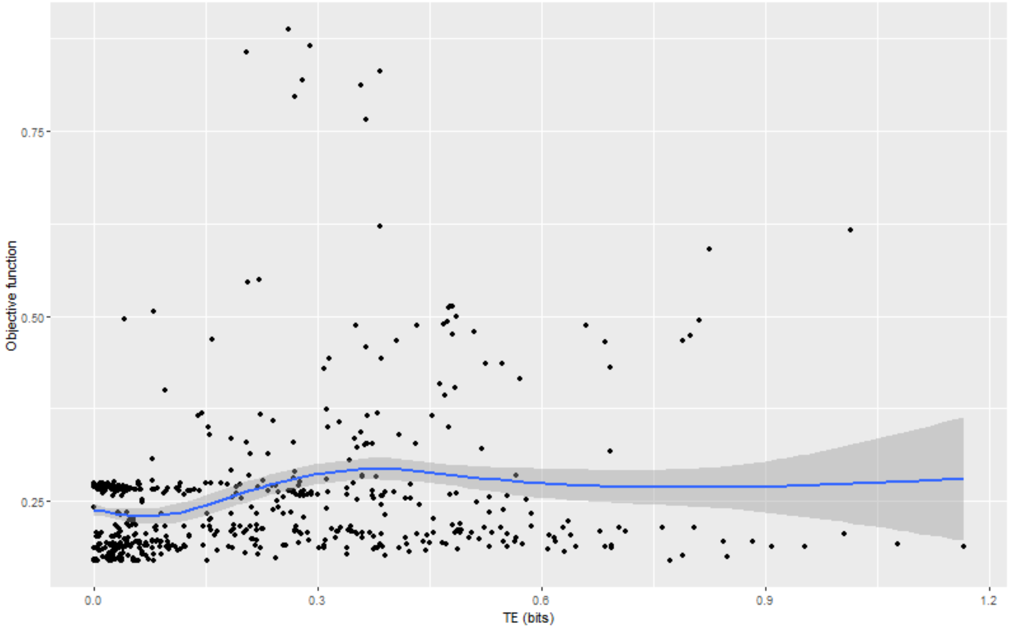


Figure 5.54: TE - F_{pt} plot for networks with $N = 20$.

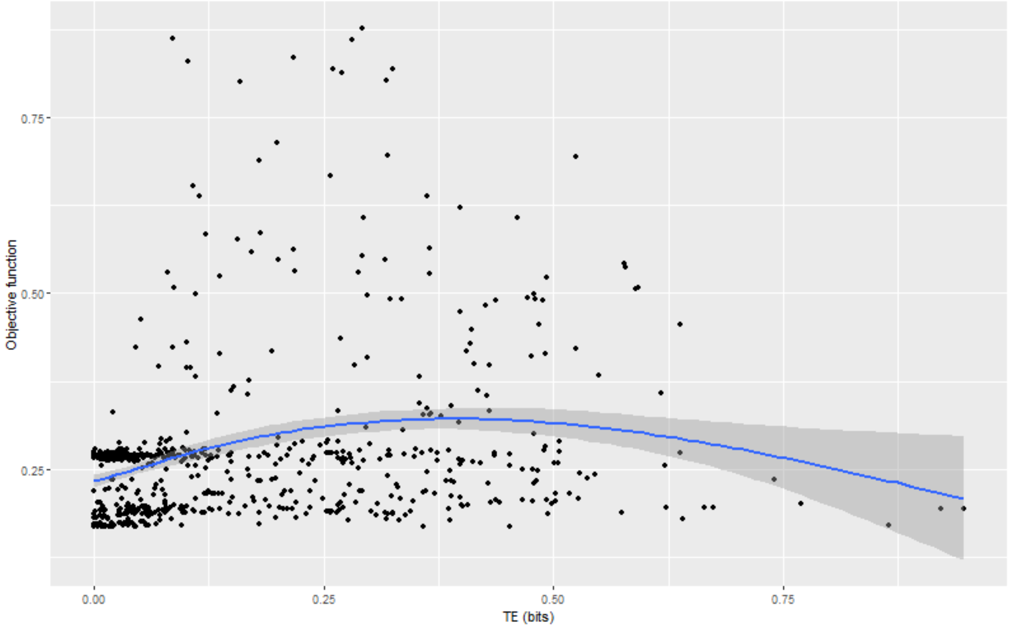


Figure 5.55: TE - F_{pt} plot for networks with $N = 50$.

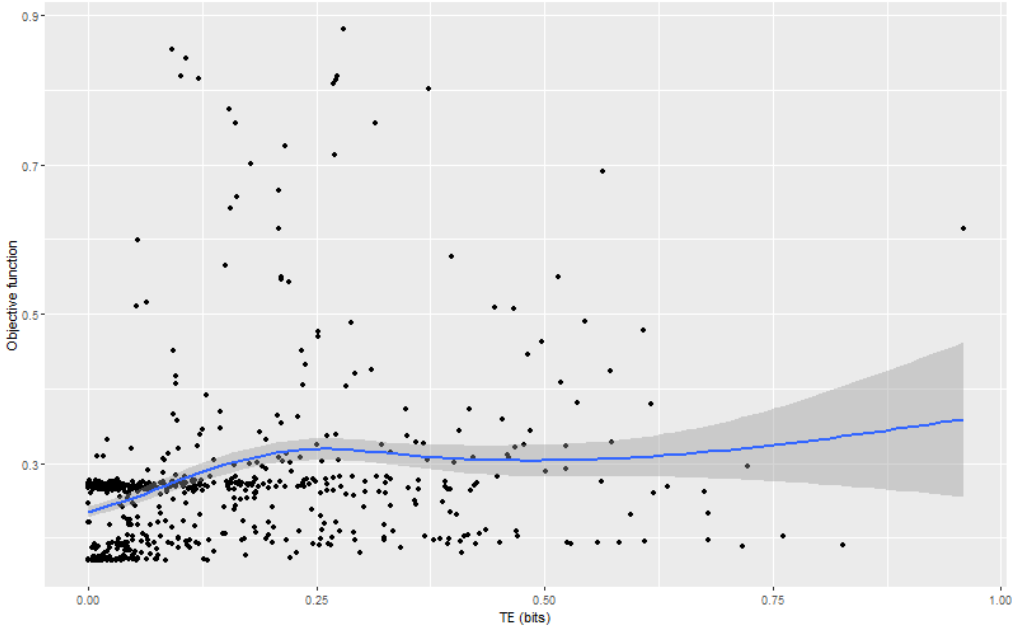


Figure 5.56: TE - F_{pt} plot for networks with $N = 100$.

We observe that there is no clear relation between the two function. The explanation for this phenomenon is the same in the PI analysis. The chosen objective function rewards robots that move towards the light source but this is completely arbitrary. Both robots that move towards the light source and robots that perform anti phototaxis could have medium or high PI value. In all three plots we notice that robots with highest score have a TE value of almost 0.25 bits.

Paths of the top 20 robots with respect to TE are reported in figure 5.53. Like for the top 20 robots with respect to PI, there are two distinct types: the ones performing phototaxis (even in a sub optimal way) and the ones performing the opposite task. We conclude asserting that there is no positive or negative correlation between transfer entropy from sensors to actuators and the chosen objective function.

RTE

RTE correlation with F_{pt} goes from 0.04 to 0.12, meaning that the two functions are fully independent to each other. RTE - F_{pt} plots are shown in figures 5.57, 5.58 and 5.59. We observe that in proximity of RTE equals to 0.05 bits there are robots with

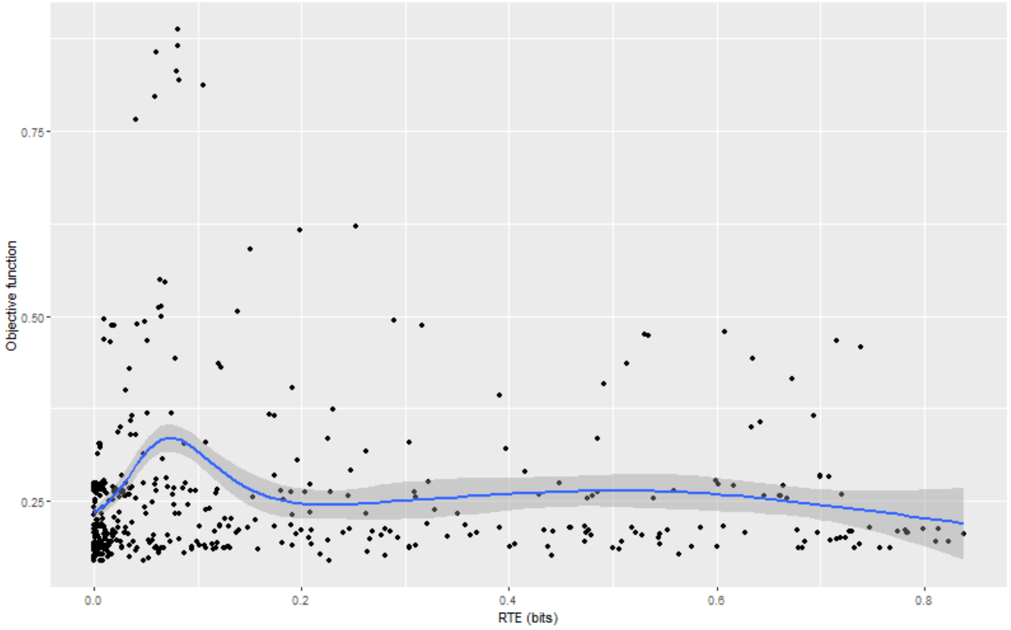


Figure 5.57: RTE - F_{pt} plot for networks with $N = 20$.

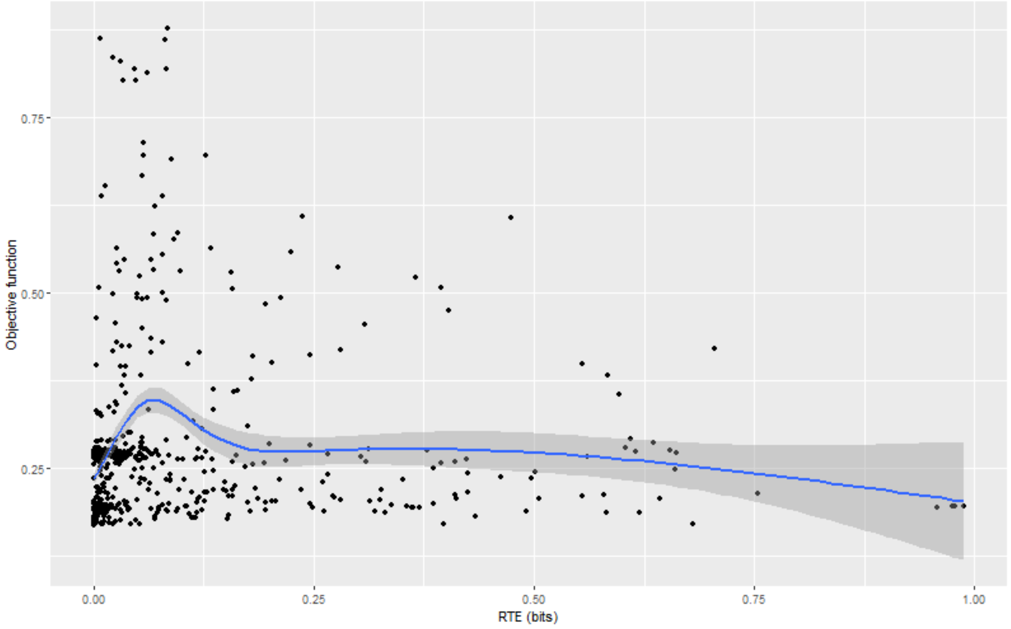


Figure 5.58: RTE - F_{pt} plot for networks with $N = 50$.

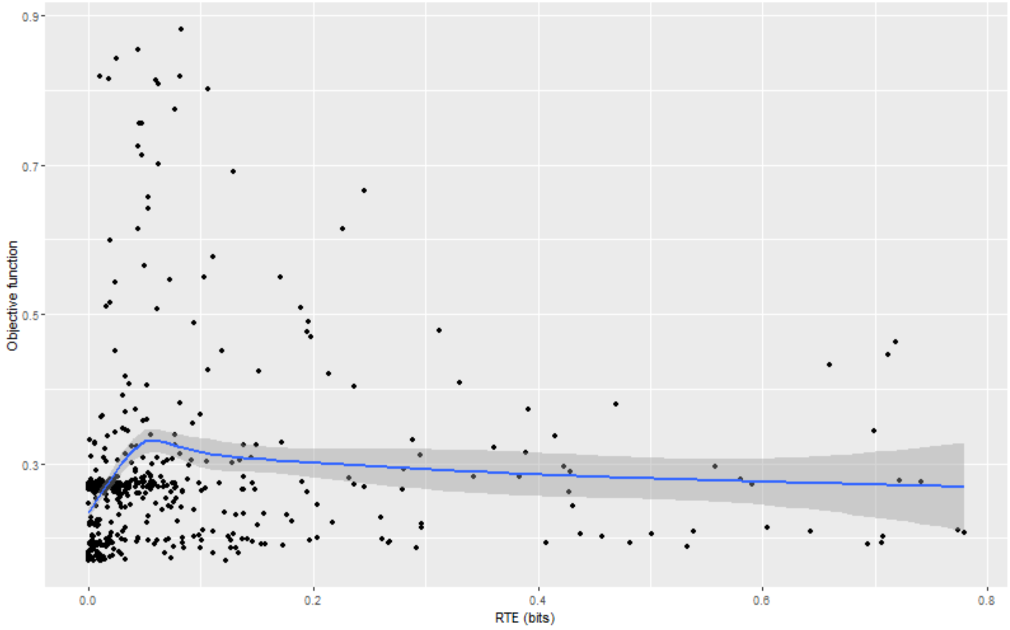
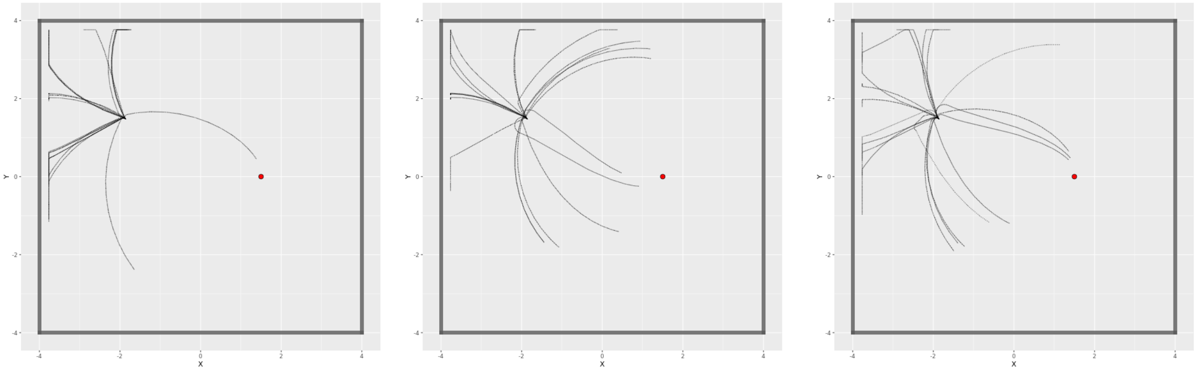


Figure 5.59: RTE - F_{pt} plot for networks with $N = 100$.

the highest score. Despite that, transfer entropy from actuators to sensors has no clear relation with F_{pt}.

Paths of the top 20 robots with respect to PI are reported in figure 5.60. We observe that in this case the robots with highest RTE prefer to run around the light source rather than directly approach. RTE is clearly the less correlated metric above the four.



(a) $N = 20$ (b) $N = 50$ (c) $N = 100$

Figure 5.60: phototaxis paths for top 20 robots with respect to RTE.

5.4 Switched controllers

We select the 20 best RBNs with respect to objective functions and metrics from all tasks grouped by number of nodes (2% of all RBNs for group). So in total we consider $20 * 5 * 3 * 3 = 900$ RBNs, possibly there are duplicates in case a RBN has more metrics in the top 20 of its group. We run these RBNs in the other two tasks that they have not seen. Our analysis focuses on verifying if among "good" RBNs based on objective function or that have high metrics there are RBNs capable of performing well in other tasks. Input sensors are changed accordingly to the type of sensor used in the task. Because we used 8 sensors in every experiment this operation is straightforward.

We define that a RBN is sufficiently good in one specific task if for that task it performs better than the best random walk robot. Thresholds are:

- 0.0461 for obstacle avoidance;
- 1.9633 for path following;
- 0.3447 for phototaxis.

In the following sections we report the results of the switching with focus on those RBNs that have at least two objective functions above the thresholds. For ease of readability we call RBNs with that characteristic "super" RBNs (SRBNs). For each sections the number of analyzed RBNs is 180 (top 20 per tasks per nodes number).

5.4.1 Objective functions

By using the top 20 RBNs with respect to the objective function of the original task we obtain the SRBNs reported in table 5.7. SRBNs are 33 (33/180 = 18.3%) and none of them has all three objective functions greater than the thresholds. SRBNs are divided as follow:

- from obstacle avoidance we have 7 RBNs (7/90 = 7.8%), all performing well in obstacle avoidance, 6 also in path following and 1 in phototaxis;
- from path following we have 24 RBNs (24/90 = 26.7%), all performing well in path following, 6 in obstacle avoidance and 18 in phototaxis;

Obstacle avoidance					Path following					Phototaxis					RBN ID
F_{oa}	SE	PI	TE	RTE	F_{pf}	SE	PI	TE	RTE	F_{pt}	SE	PI	TE	RTE	
0.061	2.6601	0.533	0.0953	0.0227	2.1878	5.0423	1.0262	0.4121	0.0702	0.1699	1.0971	0.0037	0	0	OAN20E645
0.058	2.7439	1.048	0.7022	0.2938	3.8465	3.8156	1.4447	1.0304	0.3292	0.1701	1.0996	0.0074	0	1e-04	OAN20E58
0.0011	0	0	0	0	4.6434	2.4429	1.0766	1.0075	0.066	0.3525	1.2963	0.9968	0.5917	0.007	PFN20E986
0.0011	0	0	0	0	4.0952	2.2436	1.0232	0.8598	0.2753	0.3549	1.6261	1.0511	0.5292	0.6432	PFN20E101
0.0011	0	0	0	0	3.9885	3.4876	1.2595	0.6911	0.0894	0.4833	2.1156	1.0361	0.3969	0.297	PFN20E945
0.0104	2.2242	0.8655	0.477	0.2962	3.9033	1.9001	0.3908	0.2707	0.4164	0.3524	1.4214	0.3582	0.2469	0.6291	PFN20E128
0.0348	2.1087	1.2789	0.5674	0.504	3.8188	2.4677	1.0425	1.0059	0.0602	0.3466	1.0656	0.6292	0.3086	0.3086	PFN20E829
0.0011	0	0	0	0	3.5392	1.8833	1.0228	0.7101	0.3985	0.3562	1.222	1.017	0.37	0.6675	PFN20E746
0.0011	0	0	0	0	3.5392	1.8852	1.0225	0.7114	0.3984	0.3557	1.2335	1.0119	0.3807	0.6648	PFN20E195
0.0736	3.0258	0.4206	0.363	0.0546	3.3627	2.5489	0.769	0.5838	0.0525	0.2584	0.7924	0.6021	0.333	0.0601	PFN20E747
0.0011	0	0	0	0	3.5994	2.7118	1.0272	0.8525	0.1365	0.6155	2.6002	1.4578	1.014	0.1983	PTN20E430
0.0011	0	0	0	0	3.3719	4.1844	0.748	0.2806	0.0557	0.5491	1.4051	0.2304	0.221	0.0643	PTN20E194
0.0677	2.864	0.5358	0.1495	0.0428	2.134	4.3754	0.7864	0.4005	0.0789	0.2132	1.5168	0.5407	0.3834	0.1204	OAN50E442
0.0624	2.7772	0.4666	0.0901	0.0113	2.4214	4.7853	0.9102	0.2792	0.0418	0.1692	1.3429	0.0045	0.0046	0	OAN50E659
0.0576	2.4281	0.7144	0.1019	0.034	0.1226	2.783	0.0071	0.0029	2e-04	0.3801	1.4916	0.1741	0.107	0.0335	OAN50E948
0.0011	0	0	0	0	4.6907	2.3482	1.0044	1.0016	0.0749	0.3501	1.3187	1.0195	0.5908	0.0084	PFN50E17
0.0058	0	0	0	0	3.8558	1.9525	1.0029	0.6983	0.1514	0.3541	1.4581	0.93	0.2725	0.1926	PFN50E672
0.0011	0	0	0	0	3.7432	2.8538	0.3417	0.2574	0.1951	0.3595	1.512	0.2405	0.1807	0.3132	PFN50E277
9e-04	0.267	0.2671	0.0101	0	3.7184	2.6283	1.0364	0.8893	0.2233	0.3783	1.5539	0.6137	0.3637	0.0914	PFN50E67
0.0011	0	0	0	0	3.708	1.8349	1.016	0.7075	0.4364	0.3581	1.2098	1.0216	0.3494	0.6841	PFN50E28
0.0011	0	0	0	0	3.6132	2.3843	1.0546	0.7829	0.3473	0.4807	2.1623	1.0117	0.4059	0.6956	PFN50E577
0.0011	0	0	0	0	3.5352	1.857	1.0218	0.7138	0.4014	0.3561	1.222	1.0167	0.3984	0.6471	PFN50E194
0.0011	0	0	0	0	3.5189	2.8962	0.5523	0.3325	0.1859	0.3823	1.6394	0.2019	0.4159	0.4154	PFN50E666
0.068	2.4317	0.4387	0.1152	0.0119	3.603	5.0208	0.9586	0.2661	0.0391	0.1692	1.2083	0	0	0	PFN50E296
0.0641	2.3208	0.4091	0.0866	0.0164	3.3758	4.4463	0.8539	0.3127	0.0529	0.1692	1.3429	0.0045	0	1e-04	PFN50E979
0.0623	2.7772	0.4674	0.0923	0.0113	2.42	4.7856	0.9125	0.2799	0.0425	0.1695	1.3491	0.0135	0	1e-04	OAN100E101
0.0586	2.5361	0.4589	0.1572	0.0153	3.4359	4.7231	0.2896	0.0833	0.0315	0.2889	3.367	0.5975	0.1126	0.0347	OAN100E257
0.0011	0	0	0	0	4.3874	1.9034	1.0082	0.9299	0.0688	0.361	1.2339	1.0252	0.5771	0.0167	PFN100E745
0.0011	0	0	0	0	4.2714	2.4181	0.438	0.2712	0.2742	0.4795	2.0616	0.3692	0.2529	0.6435	PFN100E965
0.0011	0	0	0	0	3.5369	2.7382	0.2482	0.2514	0.0826	0.4341	1.5244	0.2004	0.2978	0.1798	PFN100E778
0.0521	2.4431	0.613	0.363	0.0452	3.1586	3.7271	0.9601	0.6568	0.0615	0.1891	2.2278	0.0073	0.0012	0.0192	PFN100E526
0.067	2.4073	0.4377	0.0752	0.0171	2.8385	4.3834	0.783	0.2069	0.0579	0.1692	1.3429	0.0045	0	1e-04	PFN100E117
0.0534	2.729	0.2597	0.1651	0.0187	2.3044	3.6296	0.7677	0.3981	0.0703	0.1883	1.3697	0.1438	0.0972	0.0021	PFN100E777

Table 5.7: SRBNs obtained by the top 20 RBNs with respect to the objective functions. Objective function score that exceeds the corresponding thresholds is written in red.

- from phototaxis we have 2 RBNs ($2/90 = 2.2\%$), both performing well in phototaxis and path following.

By selecting the RBNs using directly the objective function we have a number of SRBNs that we expect to be greater than the SRBNs from other metrics. We observe that the majority of SRBNs came from the path following task. Among the three task path following is the most difficult, especially given the path we have chosen. So more than one RBN over four that is capable of accomplish path following is also able to perform well other tasks.

5.4.2 Sensor entropy

SRBNs obtained by top 20 RBNs with respect to SE are reported in table 5.8. There are 12 SRBNs ($12/180 = 6.6\%$), divided as follow:

Obstacle avoidance					Path following					Phototaxis					RBN ID
F_{oa}	**SE**	**PI**	**TE**	**RTE**	F_{pf}	**SE**	**PI**	**TE**	**RTE**	F_{pt}	**SE**	**PI**	**TE**	**RTE**	
0.058	2.7439	1.048	0.7022	0.2938	3.8465	3.8156	1.4447	1.0304	0.3292	0.1701	1.0996	0.0074	0	1e-04	OAN20E58
0.061	2.6601	0.533	0.0953	0.0227	2.1878	5.0423	1.0262	0.4121	0.0702	0.1699	1.0971	0.0037	0	0	OAN20E645
0.0011	0	0	0	0	3.2021	4.6964	0.5603	0.3454	0.0947	0.3679	1.5989	1.0446	0.4542	0.6711	PFN20E143
0.0542	2.3108	0.5094	0.1362	0.0123	2.6069	5.0548	0.8786	0.2877	0.0406	0.1699	1.0971	0.0037	0.0037	0	PFN20E662
0.0594	2.0853	0.2261	0.0382	0.0026	2.764	5.0351	0.8259	0.2844	0.0389	0.1692	1.2083	0	0	0	PFN20E253
0.0612	1.9702	0.3953	0.1478	0.0121	2.4214	4.7853	0.9102	0.2799	0.042	0.1692	1.3429	0.0045	0	1e-04	PFN20E284
0.0546	2.9785	0.4986	0.4225	0.0515	2.7611	4.3234	0.9535	0.6949	0.0789	0.1813	0.9493	0.0725	0.0398	0.0046	PFN20E188
0.0615	2.5163	0.3903	0.1852	0.0282	2.8486	4.2579	0.9594	0.3342	0.0486	0.2577	1.1753	0.5357	0.6126	0.5216	PFN20E84
0.0594	2.0853	0.2261	0.0382	0.0026	2.7632	5.0351	0.8259	0.2844	0.0389	0.1692	1.2083	0	0	0	PFN50E700
0.068	2.4317	0.4387	0.1152	0.0119	3.603	5.0208	0.9586	0.2661	0.0391	0.1692	1.2083	0	0	0	PFN50E296
0.047	2.1672	0.1101	0.1856	0.0348	1.9946	4.947	0.5524	0.2808	0.077	0.1734	1.4361	0.0168	0.0039	0.0512	PFN50E284
0.0483	3.0196	0.3696	0.0955	0.0275	3.0355	4.0262	0.3473	0.1769	0.0288	0.2153	2.4865	0.5844	0.1063	0.0108	OAN100E100

Table 5.8: SRBNs obtained by the top 20 RBNs with respect to sensor entropy. Objective function score that exceeds the corresponding thresholds is written in red.

- from obstacle avoidance we have 3 RBNs ($3/90 = 3.3\%$), all performing well in obstacle avoidance and also in path following;
- from path following we have 9 networks ($9/90 = 10\%$), all performing well in path following, 8 also in obstacle avoidance and 1 in phototaxis;
- from phototaxis we have none.

As expected with SE we obtain less SRBNs than by using the objective functions. In particular we observe that there are no SRBNs from phototaxis task. This is most likely due to the fact that SE is not positive correlated with F_{pt}. As in the previous case from path following comes the majority of SRBNs. Moreover there is not a SRBN able to perform well all the three tasks.

5.4.3 Predictive information

By selecting the best RBNs by PI we obtain the SRBNs reported in table 5.9. There are 18 SRBNs (18/180 = 10%) divided as follow:

Obstacle avoidance					Path following					Phototaxis					RBN ID
F_{oa}	SE	PI	TE	RTE	F_{pf}	SE	PI	TE	RTE	F_{pt}	SE	PI	TE	RTE	
0.058	2.7439	1.048	0.7022	0.2938	3.8465	3.8156	1.4447	1.0304	0.3292	0.1701	1.0996	0.0074	0	1e-04	OAN20E58
0.0011	0	0	0	0	3.9885	3.4876	1.2595	0.6911	0.0894	0.4833	2.1156	1.0361	0.3969	0.297	PFN20E945
0.0011	0	0	0	0	2.9121	2.8942	1.2025	0.3671	0.161	0.3585	1.177	0.2436	0.4019	0.3427	PFN20E641
0.0011	0	0	0	0	4.6434	2.4429	1.0766	1.0075	0.066	0.3525	1.2963	0.9968	0.5917	0.007	PFN20E986
0.0348	2.1087	1.2789	0.5674	0.504	3.8188	2.4677	1.0425	1.0059	0.0602	0.3466	1.0656	0.6292	0.3086	0.3086	PFN20E829
0.0261	2.9118	1.1739	0.789	0.5578	3.3618	2.8884	1.027	0.8349	0.1322	0.4006	2.6536	0.9069	0.6241	0.4139	PFN20E914
0.0011	0	0	0	0	4.0952	2.2436	1.0232	0.8598	0.2753	0.3549	1.6261	1.0511	0.5292	0.6432	PFN20E101
0.0011	0	0	0	0	3.5392	1.8833	1.0228	0.7101	0.3985	0.3562	1.222	1.017	0.37	0.6675	PFN20E746
0.0011	0	0	0	0	3.5994	2.7118	1.0272	0.8525	0.1365	0.6155	2.6002	1.4578	1.014	0.1983	PTN20E430
0.0011	0	0	0	0	3.6132	2.3843	1.0546	0.7829	0.3473	0.4807	2.1623	1.0117	0.4059	0.6956	PFN50E577
9e-04	0.267	0.2671	0.0101	0	3.7184	2.6283	1.0364	0.8893	0.2233	0.3783	1.5539	0.6137	0.3637	0.0914	PFN50E67
0.0011	0	0	0	0	3.5352	1.857	1.0218	0.7138	0.4014	0.3561	1.222	1.0167	0.3984	0.6471	PFN50E194
0.0011	0	0	0	0	3.708	1.8349	1.016	0.7075	0.4364	0.3581	1.2098	1.0216	0.3494	0.6841	PFN50E28
0.0011	0	0	0	0	4.6907	2.3482	1.0044	1.0016	0.0749	0.3501	1.3187	1.0195	0.5908	0.0084	PFN50E17
0.0058	0	0	0	0	3.8558	1.9525	1.0029	0.6983	0.1514	0.3541	1.4581	0.93	0.2725	0.1926	PFN50E672
0.0011	0	0	0	0	4.3874	1.9034	1.0082	0.9299	0.0688	0.361	1.2339	1.0252	0.5771	0.0167	PFN100E745
0.0521	2.4431	0.613	0.363	0.0452	3.1586	3.7271	0.9601	0.6568	0.0615	0.1891	2.2278	0.0073	0.0012	0.0192	PFN100E526
7e-04	0	0	0	0	2.614	3.3409	0.896	0.6134	0.0787	0.3796	1.5229	0.9953	0.6162	0.4698	PTN100E238

Table 5.9: SRBNs obtained by the top 20 RBNs with respect to predictive information. Objective function score that exceeds the corresponding thresholds is written in red.

- from obstacle avoidance we only have 1 RBN (1/90 = 1.1%), performing well in obstacle avoidance and in path following;

- from path following we have 15 RBNs (15/90 = 16.7%), all performing well in path following, 1 also in obstacle avoidance and 16 in phototaxis;

- from phototaxis we have 2 RBNs (2/90 = 2.2%), both performing well in phototaxis and path following.

As before none SRBN is able to perform sufficiently well all three tasks. We observe that also this time the majority of SRBNs comes from path following. Obstacle avoidance is the less frequent task because PI is not correlated with F_{oa}.

5.4.4 Transfer entropy

By using TE we obtain the SRBNs in table 5.10. We have some network repetitions from the ones obtained by PI. In total there are 22 SRBNs (22/180 = 12.2%) and they are divided as follow:

Obstacle avoidance					Path following					Phototaxis					RBN ID
F_{oa}	SE	PI	TE	RTE	F_{pf}	SE	PI	TE	RTE	F_{pt}	SE	PI	TE	RTE	
0.058	2.7439	1.048	0.7022	0.2938	3.8465	3.8156	1.4447	1.0304	0.3292	0.1701	1.0996	0.0074	0	1e-04	OAN20E58
0.0011	0	0	0	0	4.6434	2.4429	1.0766	1.0075	0.066	0.3525	1.2963	0.9968	0.5917	0.007	PFN20E986
0.0348	2.1087	1.2789	0.5674	0.504	3.8188	2.4677	1.0425	1.0059	0.0602	0.3466	1.0656	0.6292	0.3086	0.3086	PFN20E829
0.0011	0	0	0	0	4.0952	2.2436	1.0232	0.8598	0.2753	0.3549	1.6261	1.0511	0.5292	0.6432	PFN20E101
0.0261	2.9118	1.1739	0.789	0.5578	3.3618	2.8884	1.027	0.8349	0.1322	0.4006	2.6536	0.9069	0.6241	0.4139	PFN20E914
0.0011	0	0	0	0	3.2807	1.9329	0.9989	0.7557	0.3173	0.3538	1.2099	1.0143	0.4447	0.7104	PFN20E113
0.0011	0	0	0	0	3.5392	1.8852	1.0225	0.7114	0.3984	0.3557	1.2335	1.0119	0.3807	0.6648	PFN20E195
0.0011	0	0	0	0	3.5392	1.8833	1.0228	0.7101	0.3985	0.3562	1.222	1.017	0.37	0.6675	PFN20E746
0.0011	0	0	0	0	3.5994	2.7118	1.0272	0.8525	0.1365	0.6155	2.6002	1.4578	1.014	0.1983	PTN20E430
0.0011	0	0	0	0	4.3383	2.3203	1.0242	1.0126	0.0801	0.4661	2.1614	1.0576	0.7886	0.0513	PTN20E427
0.0011	0	0	0	0	4.6907	2.3482	1.0044	1.0016	0.0749	0.3501	1.3187	1.0195	0.5908	0.0084	PFN50E17
9e-04	0.267	0.2671	0.0101	0	3.7184	2.6283	1.0364	0.8893	0.2233	0.3783	1.5539	0.6137	0.3637	0.0914	PFN50E67
0.0011	0	0	0	0	3.6132	2.3843	1.0546	0.7829	0.3473	0.4807	2.1623	1.0117	0.4059	0.6956	PFN50E577
0.0011	0	0	0	0	3.5352	1.857	1.0218	0.7138	0.4014	0.3561	1.222	1.0167	0.3984	0.6471	PFN50E194
0.0011	0	0	0	0	3.708	1.8349	1.016	0.7075	0.4364	0.3581	1.2098	1.0216	0.3494	0.6841	PFN50E28
0.0058	0	0	0	0	3.8558	1.9525	1.0029	0.6983	0.1514	0.3541	1.4581	0.93	0.2725	0.1926	PFN50E672
7e-04	0	0	0	0	2.8039	3.2992	0.9535	0.6749	0.0738	0.3775	1.5383	1.0318	0.6032	0.4599	PFN50E249
0.0011	0	0	0	0	2.3642	3.766	0.6578	0.6058	0.0876	0.3647	1.1765	0.192	0.159	0.0322	PFN50E359
6e-04	0	0	0	0	2.544	3.4303	1.0102	0.6783	0.0896	0.5366	2.3614	1.0165	0.578	0.277	PTN50E17
0.0011	0	0	0	0	4.3874	1.9034	1.0082	0.9299	0.0688	0.361	1.2339	1.0252	0.5771	0.0167	PFN100E745
0.0521	2.4431	0.613	0.363	0.0452	3.1586	3.7271	0.9601	0.6568	0.0615	0.1891	2.2278	0.0073	0.0012	0.0192	PFN100E526
7e-04	0	0	0	0	2.614	3.3409	0.896	0.6134	0.0787	0.3796	1.5229	0.9953	0.6162	0.4698	PTN100E238

Table 5.10: SRBNs obtained by the top 20 RBNs with respect to transfer entropy. Objective function score that exceeds the corresponding thresholds is written in red.

- from obstacle avoidance we only have 1 SRBN (1/90 = 1.1%) performing well in obstacle avoidance and in path following;
- from path following we have 17 SRBNs (17/90 = 18.8%), all performing well in path following, 1 also in obstacle avoidance and 14 in phototaxis;
- from phototaxis we have 4 networks (4/90 = 4.4%), both performing well in phototaxis and path following.

We observe similar results to the ones obtained by PI. The lack of good controllers for obstacle avoidance is still explained by the absence of correlation between TE and F_{oa}.

5.4.5 Reverse transfer entropy

SRBNs obtained by RTE are reported in table 5.11. We have 18 SRBNs ($18/180 = 10\%$) divede as follow:

Obstacle avoidance					Path following					Phototaxis					RBN ID
F_{oa}	SE	PI	TE	RTE	F_{pf}	SE	PI	TE	RTE	F_{pt}	SE	PI	TE	RTE	
0.058	2.7439	1.048	0.7022	0.2938	3.8465	3.8156	1.4447	1.0304	0.3292	0.1701	1.0996	0.0074	0	1e-04	OAN20E58
0.0104	2.2242	0.8655	0.477	0.2962	3.9033	1.9001	0.3908	0.2707	0.4164	0.3524	1.4214	0.3582	0.2469	0.6291	PFN20E128
0.0011	0	0	0	0	3.5392	1.8833	1.0228	0.7101	0.3985	0.3562	1.222	1.017	0.37	0.6675	PFN20E746
0.0011	0	0	0	0	3.5392	1.8852	1.0225	0.7114	0.3984	0.3557	1.2335	1.0119	0.3807	0.6648	PFN20E195
0.0011	0	0	0	0	3.2807	1.9329	0.9989	0.7557	0.3173	0.3538	1.2099	1.0143	0.4447	0.7104	PFN20E113
0.0011	0	0	0	0	4.0952	2.2436	1.0232	0.8598	0.2753	0.3549	1.6261	1.0511	0.5292	0.6432	PFN20E101
0.0011	0	0	0	0	2.9121	2.8942	1.2025	0.3671	0.161	0.3585	1.177	0.2436	0.4019	0.3427	PFN20E641
0.0011	0	0	0	0	3.708	1.8349	1.016	0.7075	0.4364	0.3581	1.2098	1.0216	0.3494	0.6841	PFN50E28
0.0011	0	0	0	0	3.5352	1.857	1.0218	0.7138	0.4014	0.3561	1.222	1.0167	0.3984	0.6471	PFN50E194
0.0011	0	0	0	0	3.6132	2.3843	1.0546	0.7829	0.3473	0.4807	2.1623	1.0117	0.4059	0.6956	PFN50E577
9e-04	0.267	0.2671	0.0101	0	3.7184	2.6283	1.0364	0.8893	0.2233	0.3783	1.5539	0.6137	0.3637	0.0914	PFN50E67
0.0011	0	0	0	0	3.7432	2.8538	0.3417	0.2574	0.1951	0.3595	1.512	0.2405	0.1807	0.3132	PFN50E277
0.0011	0	0	0	0	3.5189	2.8962	0.5523	0.3325	0.1859	0.3823	1.6394	0.2019	0.4159	0.4154	PFN50E666
0.0108	1.6504	1.0947	0.649	0.2541	2.3047	1.8445	0.9232	0.508	0.1599	0.3496	1.3348	0.9912	0.5074	0.1639	PFN50E649
0.0058	0	0	0	0	3.8558	1.9525	1.0029	0.6983	0.1514	0.3541	1.4581	0.93	0.2725	0.1926	PFN50E672
2e-04	0	0	0	0	2.1285	3.6949	0.6533	0.2777	0.0772	0.355	1.6006	0.4434	0.4284	0.5969	PTN50E201
0.0011	0	0	0	0	4.2714	2.4181	0.438	0.2712	0.2742	0.4795	2.0616	0.3692	0.2529	0.6435	PFN100E965
0.0047	0	0	0	0	3.7368	2.377	1.0181	0.8543	0.1438	0.4466	2.2824	0.9794	0.4811	0.7118	PTN100E571

Table 5.11: SRBNs obtained by the top 20 RBNs with respect to reverse transfer entropy. Objective function score that exceeds the corresponding thresholds is written in red.

- from obstacle avoidance we only 1 SRBN ($1/90 = 1.1\%$) performing well in obstacle avoidance and in path following;
- from path following we have 15 SRBNs ($15/90 = 16.7\%$), all performing well in path following, 1 also in obstacle avoidance and 16 in phototaxis;
- from phototaxis we have 2 SRBNs ($2/90 = 2.2\%$), all performing well in phototaxis and path following.

Like in the other SRBNs the majority of them came from path following. Because also RTE is not correlated with F_{oa} we do not have many SRBNs good in obstacle avoidance.

Overall there are 103 SRBNs with repetitions, the unique ones are 53. The percentage of SRBNs over the unique RBNs taken into account is $53/637 = 8.3\%$. They are divided as follow:

- from obstacle avoidance 8 unique SRBNs (15.1%);
- from path following 38 unique SRBNs (71.7%);
- from phototaxis 7 unique SRBNs (13.2%).

SRBNs repartition is summarized in table 5.12. The majority of them comes from path following and we explain this thanks to the high positive correlation between the metrics and F_{pf}. However these networks are suitable also for obstacle avoidance or phototaxis. We known that especially for phototaxis metrics are not positive correlated with the chosen objective function of that task. Indeed, SRBNs that have high metrics in path following and also perform well in phototaxis have for the latter task medium or low metrics values. We observe a similar behaviour for SRBNs from path following also good in obstacle avoidance: all metrics are low except for SE that is correlated with F_{oa}. SRBNs from obstacle avoidance and phototaxis that perform well in path following have instead general high metrics values for the latter task due to high correlation with F_{pf}.

Table 5.12: summary of SRBNs repartition: cells hold the number of SRBNs that perform well in the task in the row (starting task) and in the task in the column (switched task). X indicates that all SRBNs perform well.

	Obstacle avoidance	**Path following**	**Phototaxis**
Obstacle avoidance	X	7	1
Path following	13	X	25
Phototaxis	0	7	X

Conclusion

In this work we conducted an analysis over critical random Boolean networks used as control software for robot in different tasks. The analysis is focused over the network configurations and the correlation between information theory metrics and task objective function. We chose three different critical random Boolean network configurations: $\{K = 3, p = 0.79, N = 20\}$, $\{K = 3, p = 0.79, N = 50\}$ and $\{K = 3, p = 0.79, N = 100\}$. These RBNs are used in experiments as control software for robot. We defined three well known tasks in robotics: obstacle avoidance, path following and phototaxis. For each of them we ran 1000 experiments per network configuration. We also defined a random walk controller as a benchmark for the RBNs. We ran additional 1000 experiments per task with the random walk controller. Experiments are all reproducible through the ARGoS simulator software [20], the robot used is the foot-bot [5]. For each RBN experiment we computed four metrics built from information theory measures: the sensor entropy, the predictive information between sensors and actuators, the transfer entropy from sensors to actuators and the transfer entropy from actuators to sensors. For all experiment we computed also an handwritten objective function that is unique for each task.

We acknowledged that by simply creating a significant statistical number of critical RBNs some of them are able of performing extremely well the given task, outperforming the random walk controller. We also tested that the number of nodes of a critical RBN is statistically relevant to obtain a fitter population of controllers for a given task. In particular RBNs with $N = 50$ or $N = 100$, given the robot and task configurations we have chosen, are better than RBNs with $N = 20$ in all tasks. We found that in obstacle avoidance there is a mildly positive correlation between sensor entropy and the objective function. In path following both the two transfer entropies are strongly positively correlated with the objective function, also predictive information is positively correlated, sensor entropy only partially. In phototaxis no metric is positively nor negatively correlated with the chosen objective function. These results suggest that transfer entropy can be used as-is as a not handwritten metric factor in **automatic design** of control software for robots for path following. Also predictive information in path following and sensor entropy in obstacle avoidance could be used with preliminary changes.

Finally we switched the best 20 RBNs with respect to network configuration, metric and task to test them in the other two tasks. We found that a critical RBN is capable of performing well more than one task. In particular we observed that RBNs from path following that perform it well are more inclined to perform well also in the other two tasks rather than RBNs from obstacle avoidance and phototaxis. This last result suggests that a critical RBN fitted for a task could be be able to accomplish other tasks without the need to be modified. This phenomenon in biology is called **phenotypic plasticity**: when the same genotype produces more than one phenotype in different environments. In [8] is shown that phenotypic plasticity can be reached in robots with online adaptation. With this work we found that is possible to have some critical RBNs from a population that exhibit phenotypic plasticity by simply random generate them without the need to use online adaptation.

More research is needed on critical RBNs that have phenotypic plasticity to exploit their potential in applications. Also more study on different tasks could be helpful to explore other relations between information theory measures and robot's ability in achieving the goal. Definitely new tests on the same tasks, but with different configurations from the ones we analyzed, are needed to verify that the discovered relations between objective functions and metrics are generalizable.

Appendix A

Additional material

Videos of some of the more interesting experiments can be found at this link.

Bibliography

[1] Ahlswede, R., and Csiszár, I. Common randomness in information theory and cryptography. i. secret sharing. *IEEE Transactions on Information Theory 39*, 4 (1993), 1121–1132.

[2] Aldana, M. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena 185*, 1 (2003), 45–66.

[3] Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B 63*, 3 (2008), 329–339.

[4] Bialek, W., and Tishby, N. Predictive information, 1999.

[5] Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., Vaussard, F., Bleuler, H., and Mondada, F. The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010), pp. 4187–4193.

[6] Borst, A., and Theunissen, F. E. Information theory and neural coding. *Nature neuroscience 2*, 11 (1999), 947–957.

[7] Braccini, M., Montagna, S., and Roli, A. Self-loops favour diversification and asymmetric transitions between attractors in boolean network models. In *Artificial Life and Evolutionary Computation* (Cham, 2019), S. Cagnoni, M. Mordonini, R. Pecori, A. Roli, and M. Villani, Eds., Springer International Publishing, pp. 30–41.

[8] Braccini, M., Roli, A., and Kauffman, S. A. Online adaptation in robots as biological development provides phenotypic plasticity, 2020.

[9] Derrida, B., and Pomeau, Y. Random networks of automata: A simple annealed approximation. *Europhysics Letters (EPL) 1*, 2 (jan 1986), 45–49.

[10] Edlund, J., Chaumont, N., Hintze, A., Koch, C., Tononi, G., and Adami, C. Integrated information increases with fitness in the evolution of animats. *PLoS computational biology 7* (10 2011), e1002236.

[11] Gershenson, C. Classification of random boolean networks. In *Proceedings of the Eighth International Conference on Artificial Life* (Cambridge, MA, USA, 2002), ICAL 2003, MIT Press, p. 1–8.

[12] Gnucci, A. *Online adaptation in Boolean network robots.* PhD thesis, University of Bologna, 2020.

[13] Hankerson, D. R., Johnson, P. D., and Harris, G. A. *Introduction to Information Theory and Data Compression*, 2nd ed. Chapman &amp; Hall, Ltd., GBR, 2003.

[14] Kauffman, S. Homeostasis and differentiation in random genetic control networks. *Nature 224* (1969), 177–178.

[15] Kauffman, S. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology 22*, 3 (1969), 437–467.

[16] Lizier, J. T., Pritam, S., and Prokopenko, M. Information dynamics in small-world boolean networks. *Artificial life 17*, 4 (2011), 293–314.

[17] Montagna, S., Braccini, M., and Roli, A. The impact of self-loops in random boolean network dynamics: A simulation analysis. In *Artificial Life and Evolutionary Computation* (Cham, 2018), M. Pelillo, I. Poli, A. Roli, R. Serra, D. Slanzi, and M. Villani, Eds., Springer International Publishing, pp. 104–115.

[18] Nykter, M., Price, N. D., Aldana, M., Ramsey, S. A., Kauffman, S. A., Hood, L. E., Yli-Harja, O., and Shmulevich, I. Gene expression dynamics in the macrophage exhibit criticality. *Proceedings of the National Academy of Sciences 105*, 6 (2008), 1897–1900.

[19] Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics 33*, 3 (1962), 1065–1076.

[20] Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence 6*, 4 (2012), 271–295.

[21] Roberto Ierusalimschy, Luiz Henrique de Figueiredo, W. C. Lua 5.4 reference manual, 2020. [Online; accessed 13-February-2021].

[22] Roli, A., and Braccini, M. Attractor landscape: A bridge between robotics and synthetic biology. *Complex Systems 27* (10 2018), 229–248.

[23] Roli, A., Manfroni, M., Pinciroli, C., and Birattari, M. On the design of boolean network robots. vol. 6624, pp. 43–52.

[24] Schreiber, T. Measuring information transfer. *Physical review letters 85*, 2 (2000), 461.

[25] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal 27*, 3 (1948), 379–423.

[26] Vinga, S. Information theory applications for biological sequence analysis. *Briefings in Bioinformatics 15*, 3 (09 2013), 376–389.

[27] Wikipedia contributors. Conditional mutual information — Wikipedia, the free encyclopedia, 2020. [Online; accessed 5-February-2021].