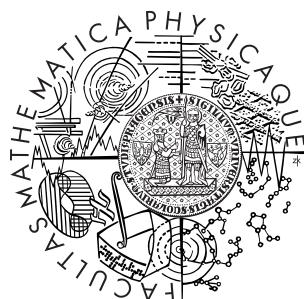


Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS



Vojtěch Kopal

Komunikace a paměť pro plausibilní agenty Communication and memory in plausible agents

Department of Theoretical Computer Science and
Mathematical Logic

Supervisor: Mgr. Ondřej Sýkora

Study programme: General Computer Science

2011

Na tomto místě mohou být napsána případná poděkování (vedoucímu práce, konzultantovi, tomu, kdo půjčil software, literaturu, poskytl data apod.).

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne

Vojtěch Kopal

Contents

1	Introduction	6
2	Related work	8
2.1	Agents	8
2.2	Spatial resource-bounded memory	10
2.2.1	Resource-bounded reasoning	10
2.2.2	Short-term and long-term memories	10
2.2.3	Computational memory architectures	11
2.2.4	How Place and Objects Combine?	12
2.2.5	Inspirations for my work	13
3	Used methods and algorithms	14
3.1	Growing Neural Gas	14
3.1.1	Topology learning	14
3.1.2	Experiments on dynamic data	16
3.2	Grid	16
4	Simulation and used memory architectures	20
4.1	Simulation	20
4.2	Environment	20
4.3	Agent	20
4.4	Communication	21
4.5	Decision making	22
4.6	Memories	22
5	Implementation	25
5.1	Simulation application	25
5.2	Memories	25
5.3	Growing neural gas as a memory	26

5.4	Grid as a memory	27
6	Experiments	29
6.1	Experimental settings and methodology	29
6.2	Homogeneous agent set comparison with communication	29
6.2.1	Random agent	31
6.2.2	PR agent	33
6.2.3	GNG agent	34
6.2.4	Grid agent	35
6.3	Mixed environment	36
6.3.1	GNG+Grid+PR+Random agents	36
6.3.2	GNG+Grid+Random agents	38
6.3.3	GNG+Grid agents	40
7	Discussion	41
8	Conclusions	43
	Bibliography	44

Název práce: Název bakalářské práce

Autor: Jméno autora

Katedra (ústav): Název katedry či ústavu

Vedoucí bakalářské práce: Jméno se všemi tituly, event. pracoviště

e-mail vedoucího: e-mailová adresa vedoucího

Abstrakt: V předložené práci studujeme ... Uvede se abstrakt v rozsahu 80 až 200 slov. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sit amet sem. Mauris nec turpis ac sem mollis pretium. Suspendisse neque massa, suscipit id, dictum in, porta at, quam. Nunc suscipit, pede vel elementum pretium, nisl urna sodales velit, sit amet auctor elit quam id tellus. Nullam sollicitudin. Donec hendrerit. Aliquam ac nibh. Vivamus mi. Sed felis. Proin pretium elit in neque. Pellentesque at turpis. Maecenas convallis. Vestibulum id lectus. Fusce dictum augue ut nibh. Etiam non urna nec mi mattis volutpat. Curabitur in tortor at magna nonummy gravida. Mauris turpis quam, volutpat quis, porttitor ut, condimentum sit amet, felis.

Klíčová slova: klíčová slova (3 až 5)

Title: Název bakalářské práce v angličtině

Author: Jméno autora

Department: Název katedry či ústavu v angličtině

Supervisor: Jméno s tituly jako v české verzi, event. pracoviště

Supervisor's e-mail address: e-mailová adresa vedoucího

Abstract: In the present work we study ... Uvede se anglický abstrakt v rozsahu 80 až 200 slov. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sit amet sem. Mauris nec turpis ac sem mollis pretium. Suspendisse neque massa, suscipit id, dictum in, porta at, quam. Nunc suscipit, pede vel elementum pretium, nisl urna sodales velit, sit amet auctor elit quam id tellus. Nullam sollicitudin. Donec hendrerit. Aliquam ac nibh. Vivamus mi. Sed felis. Proin pretium elit in neque. Pellentesque at turpis. Maecenas convallis. Vestibulum id lectus. Fusce dictum augue ut nibh. Etiam non urna nec mi mattis volutpat. Curabitur in tortor at magna nonummy gravida. Mauris turpis quam, volutpat quis, porttitor ut, condimentum sit amet, felis.

Keywords: klíčová slova (3 až 5) v angličtině

Chapter 1

Introduction

In a modern society the amount of information is far behind what one can remember or even process. If we understand that, we realize how important it is to be able to delegate the thinking amongst group. The decision making in groups and teams is a topic covered by several papers. [citation required] Supposing we have limited capacity of memory, we have to distribute the knowledge amongst people around us and communicate with each other so as to gather facts which we currently need to make the decision.

Our decisions are either consciously or subliminally based on our needs or drives - former term might be rather connected with human behavior, latter term is used for plausible agents. As in microeconomics we can use an utility as a measure of relative satisfaction [13] and see how one manages fulfilling their needs. While attaining the goals we use a knowledge which we store in our memory and which we update regularly. With infinite memory we wouldn't have any problems to store all information and use it when required; however, we don't have such memory - our memory is limited.

What I mean by saying "not to have enough space in our memory" is one is not able to remember everything. Certain pieces of information are fading away as time goes or as one is learning new facts. I want to observe if and how an intensive communication can substitute insufficient memory space with the condition of constant level of utility.

Is it obvious that adding the ability of communication improves the agents' chances to survive in the environment.

I want to observe the relation between amount of communication and used spatial resource-bounded memory.

This thesis consists of [N] parts. First, I will introduce the topic of agent

and possible memory implementations based on concrete examples (Chapter 2). Then I will explain what algorithms, such as growing neural gas, I am going to use in the program (Chapter 3). In *Chapter 4* there is a description of the simulation, agents, their memory and the communication.

Chapter 2

Related work

I will use this chapter as an insight into the world of agents and spatial memory. I hope that you will not be disappointed, since there is no 007 in following lines. [bullshit]

2.1 Agents

There are several ways how to explain what or who an agent is. Apart from systems of agents used in philosophy or sociology, we can see a first modern use of agency and agents in economy where economists have substituted the human with a simpler agent. They intended to simplify their economic models to be able to actually simulate something. Buyers and sellers are typical examples of agents used in simplified market model in microeconomics (see [1]). In this context agents are entities in the model which can act based on situation in the model.

For area of artificial intelligence we can use the definition of an agent which can be found in [10]. It cannot be more simple:

Definition 1 *Agent is just something that acts.*

Of course it is as general as it could be and for my purposes it is too simple, so I will use another definition which meets better the context of my work.

Definition 2 *Agent is something that senses the environment and affects it using its actuators.*

Having that defined we continue to specific kinds of agent. In this thesis I use several slightly varying terms about agents: *rational*, *autonomous*, *plausible* and *believable*.

A rational agent refers back to economics where we can find a definition of rational behaviour. Even though it is rather a hypothetical model, as people are usually irrational in their decisions from the economics perspective, there is yet nice definition whereby a rational agent acts as if balancing costs against benefits to arrive at action that maximizes personal advantage (Milton Friedman (1953), Essays in Positive Economics). So simply he does what is or perhaps might be best for him based on his current knowledge of the world.

On the other hand, the rational behaviour might be understood in a completely different way. Plausible agents are such agents, where the basic approach is to implement human-like internal processes. One of the well-known example is neural networks, although they are usually used in quite simplified way. Since it is really difficult to implement completely plausible agent, one can see research teams focusing on a specific part of the complex human being.

Autonomous agents are those agents which are capable of accomplishing useful tasks or are effective problem solvers [1]. A

Believable agents are personality-rich autonomous agents with the powerful properties of characters from the arts [1]. Now there is just the autonomous agent left. An autonomous agent should be able to accomplish useful task or be an effective problem solver. I would like to add one more term which is going to fit the agents I use.

Belief-Desire-Intention (BDI) agency model implements the three parts agent's belief, desire and intention and use them when comes to reasoning. A BDI agent is particular part of bounded rational agent who use those three parts to separately prepare plans which are later executed. What a BDI distinguishes from a simple reactive agent [nevysvetleny pojem] is a reactive agent creates immediate decisions based on current state of environment and inputs of his sensors. On the other hand, a BDI agent uses the three parts:

- **belief** represents the agent's informational state, for example sensory inputs and information in his memory,
- **desire** is the agent's motivational state, what he needs to approach, for example he is hungry and he needs to find appropriate food,
- **intention**, on the other hand, is his immediate decision how he attaining the goal he desires, in other words it is execution of plan, for example next move.

2.2 Spatial resource-bounded memory

A memory is something what changes a reactive agent into an agent with ability to learn. It can be used for learning consequences of agent's acts, conditional dependencies in the agent's world (citation for bayesian networks), or spatial information about the environment [neni zcela jasne, ze jde o dva rozlisitelne typy mezi kterymi si mas vybrat]. The latter one is a kind of memory I used for agents in my simulation.

A spatial memory is used when agent needs to navigate in usually two or three dimensional space. In short it is a component of an agent which says him where to go when he needs or want to do something. There are several different approaches and a couple of examples are going to be covered in this section. I am going to introduce several existing implementations of spatial memory. Mainly I will focus on if and how they have dealt with bounded resources - either due to implementation restrictions, or when approaching plausibility in their models.

2.2.1 Resource-bounded reasoning

Rational agents cannot be expected to be able to compute a load of data in a constant time or in a time in which the environment doesn't change much. That is why we have to take into account bounded resources when simulating plausible or rather real agents. What we want to avoid here is the computation of plan takes a long period of time during which the environment changes significantly. As they have mentioned in [4], we could separate plan computation from executing the plan, whereby the plan is prepared over several executions. In that case we need either to be able to perfectly predict the future, or base our plan on data which does not change at all or is frozen for the given period of time.

2.2.2 Short-term and long-term memories

Generally, when I talk about remembering something, I should mention two terms: a long-term memory (LTM) and a short-term memory (STM). Both of which describes a capacity for holding certain amount of information in mind. Apart from the varying amount, the memories differ in availability of such information and a period of time the memories last.

In 1968 Richard Atkinson and Richard Shiffrin suggested in [2] a memory model parted into three components:

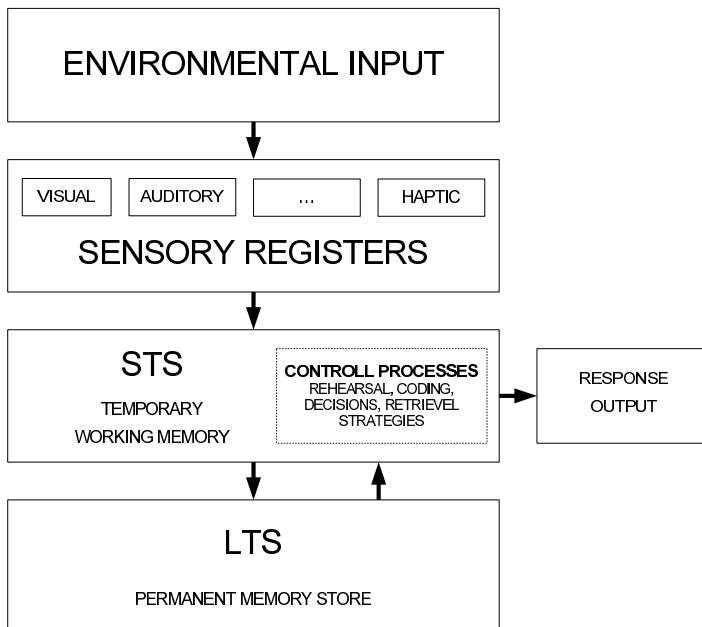


Figure 2.1: Information flow in the memory system has been depicted in [3].

- (a) sensory register being able to store only a relatively small amount of data for a short period of time,
- (b) short-term store with ability to store also limited amount of data, but for quite a longer period of time,
- (c) long-term store with a huge capacity for nearly unlimited time.

We [obecne my nebo ty v praci? pokud ty, tak mixujes We a I] use the **short-term memory** for storing pieces of information for relatively short period of time. That could be seconds or minutes. A number of entities we are able to hold in our STM was researched by George Miller in 1956 as it is mentioned in [11]. The outcome of his work was the magical number 7+2, which is the number of similar small things we can remember in STM.

A long-term memory, on the other hand, is used for information we do not think of consciously, but pieces of such information are important for our everyday life. The capacity of LTM memory is unknown, as there is no way how to find it.

2.2.3 Computational memory architectures

Computational memory architectures for autobiographic agents interacting in complex virtual environment suggested by Ho in [7].[konec vety nasledovany ma-

lym pismenem] works with both short-term and long-term autobiographic memory, where they have observed agent's ability to survive comparing to purely reactive agency model. Moreover, they researched whether the narrative communication amongst agents somehow positively influence those agents. They have separately experimented with three types of agents: purely reactive (PR), short-term memory (STM) and long-term memory (LTM). Purely reactive agent walks randomly around the environment avoiding obstacles and searching for resource objects to fulfill his needs. What a pure reactivity means is the agent moves randomly until an event occurred such as a collision with obstacle or a resource object detection.

STM agents in [7] further extend the model of purely reactive agents and add a Track-back memory system in addition to the reactive behaviour. Each time an agent deals with an event (e.g. collision, or resource object) he puts such information into his memory. They refer to this as an event-based memory entry making mode. Those events are kept in a linear list of a finite size, whereby the oldest events are cut off. The memory is used when an internal variable is over threshold. That is the moment when agent searches in his memory for an information about relevant resource object. If he succeeded, he retrospectively undoes all memorized states leading to the relevant one. So, what they actually store in memory is an agent's current state: where he was and what he perceived. While attaining imperfection in retrieving information from short-term memory, they introduced noise distortion using Gaussians.

Long-term memory model is mostly based on psychological autobiographic memory models. There are three parts that are involved in the reasoning process: Event specific knowledge (ESK), Event reconstruction process (ER) and Event filtering and ranking process.

2.2.4 How Place and Objects Combine?

This paper written by Brom et al [5] is mainly focused on plausible behaviour while searching things in structured spaces such as flats. Apart from others who previously researched the area of spatial memory for plausible agents, they suggested a model for an agent which could successfully live in a dynamic environment with objects which could be moved without the agent's involvement.

In this model the environment consists of abstract and concrete areas such as rooms and pieces of furniture respectively. Those areas are combined into a tree structure used in the model where e.g. the flat is a root node, the immediate child nodes are rooms and, finally, concrete pieces of furniture are child nodes

below room nodes. They have created four different categories for objects to reflect varying probabilities of changing object's location which fact simulates a presence of another agent in the environment. So the observed agent does not leave the flat and he is alone there.

During their experiments, what is interesting for my work is they subsequently observed the ability of the model to emerge the searching rules from scratch, to relearn the rules in case of changing particular settings, and if the merged rules meet with the human behaviour, i.e. they are believable. [možna jsem to cítit moc rychle, ale nepochopil jsem, k čemu ten strom ve výsledku pomohl]

2.2.5 Inspirations for my work

The suggested RTM and LTM models of agents in [7] are more than interesting to be implemented in my simulation [ohání se svéj práci, ale vlastné jsi nikde nerekl, co je její cílem, takže nemůžu posoudit, jestli je to vzdálené tak zajímavý]. There has to be a couple of minor changes, though, as I am working with a simpler environment comparing to the one used in the work described above. The changes will influence the structure of memory records in both RTM and LTM. Also the communication protocol will be different and I am going to introduce it later in this thesis. Although it is not going to be part of this particular thesis, having compared their memory models is definitely promising.

In my project the environment I want to use doesn't have differentiated area as is in [5], that means it is homogeneous. For the purpose of implementing an agent with a similar [aspell] spatial "What-where" memory model [wtf], I will use different spatial organization which could be similarly structured into a grid [jaký výpočetní grid?]. The suggested model, however, is usable for objects which are moved around the environment and I don't have such objects. Objects in my environment are generated around distribution place and locations of those places are something what could be learnt using the "What-Where" model.

Chapter 3

Used methods and algorithms

In previous chapter I have introduced you to several kinds of agents, how they can be used and also what a spatial memory is. I have briefly prepared you for the next chapters, where I will explain my contribution to this area. This chapter is going to cover the used algorithms and computational methods I have studied and implemented in my work.

The first subsection discusses on the implementations of agents' memory and in detail describes fundamental parts. Both the Growing Neural Gas and the Quad..blah [tohle asi neni idealni] are used as memory storages to handle spatial information about the environment with bounded resources.

3.1 Growing Neural Gas

3.1.1 Topology learning

Processing an enormous spatial data about an environment is computationally demanding when for example we want to navigate in that environment. A topology learning or recognition can help us to create a representation such as topological map which can be viewed as a graph and which makes reasoning in that environment much easier. Rather complex understanding of topology in an indoor space using Bayesian programming has been shown in [12]. It goes much farther than I need to.

Based on competitive Hebbian learning (CHL) method [9] and Neural Gas (NG) [8] Bern Fritzke suggested earlier mentioned Growing Neural Gas [6], an unsupervised learning method for finding a topological structure which reflects the topology of the data distribution. Although the combination of both CHL

and NG is an effective method for topology learning, there are some flaws in practical application as it requires an initial setup of number of nodes/centers that are used. This fact prevents the method from adequately describing the topology, when a different number of nodes would work better.

As Fritzke described the algorithm uses a set of nodes and edges that connects the nodes. A simplified describtion of algorithm from [6] in context of two-dimensional space follows:

1. Add two nodes at random position onto canvas
2. Generate input signal based on the data distribution (its probability density)
3. Find the nearest node n_1 and second nearest node n_2 to the signal
4. Increment the age of all edges leading from node n_2
5. Add the squared distance between the input signal and the nearest unit in input space to a local counter variable $\Delta error(n_1)$
6. Moved node n_1 and its topological neighbors towards the signal (according to parametres $epsilon_{winner}$ and $epsilon_{neighbour}$)
7. Remove all edges with an age larger than a_{max}
8. Generate new nodes (see [6]) using variable $alpha$
9. Decrease all error variables by multiplying them with a constant $beta$
10. Go to 1.

For the purpose of this work I want to use this algorithm to learn a topology of data which dynamically changes through the time. We have to setup the variables for this agorithm $alpha$, $beta$, $epsilon_{winner}$, $epsilon_{neighbour}$ and maximal number of nodes. In following subsection I am going to introduce you to the experimenting with this algorithm.

```

procedure Score()
  (px , py , pvar) <- GetProbableGauss()
  (rx , ry , rvar) <- GetRealGauss()

  sqDistance <- (px - rx)*(px - rx) + (py - ry)*(py - ry)
  sqSize <- (pvar + rvar)*(pvar + rvar)

  score <- sqDistance / sqSize

  return score
end

```

Figure 3.1: The *SCORE* method

3.1.2 Experiments on dynamic data

As I have mentioned previously I had to setup the variables so as to be able to use Growing Neural Gas method properly. To attain this goal I have made a Java programm which tests various combinations of variables' values and finds the best one. It has sequently run the algorithm for a given number of steps and measured the *score* (see 3.1).

[proc zrovna takhle ten algoritmus, jak z toho poznam, ze tohle score je takhle ok?]

A total number of possible combinations is equal to 19712. For each such a combination I have run 10000 steps of GNG learning sequence and measured the average score. Each sequence took aproximately 108 seconds [to je jak brambor, kdyz tu neni konfigurace] and all the experiment was computed paralelly using 30 threads.

The best results which were avarage $SCORE < 10^{-11}$ is shown in a table 3.1.

3.2 Grid

The idea for this data structure representing resource-bounded memory is based on [5]. What differs in my work from their observed environment is agents in my simulation act in a homogeneous space which cannot be differentiated in a way the mentioned simulation does. To solve this issue I have simply differentiated

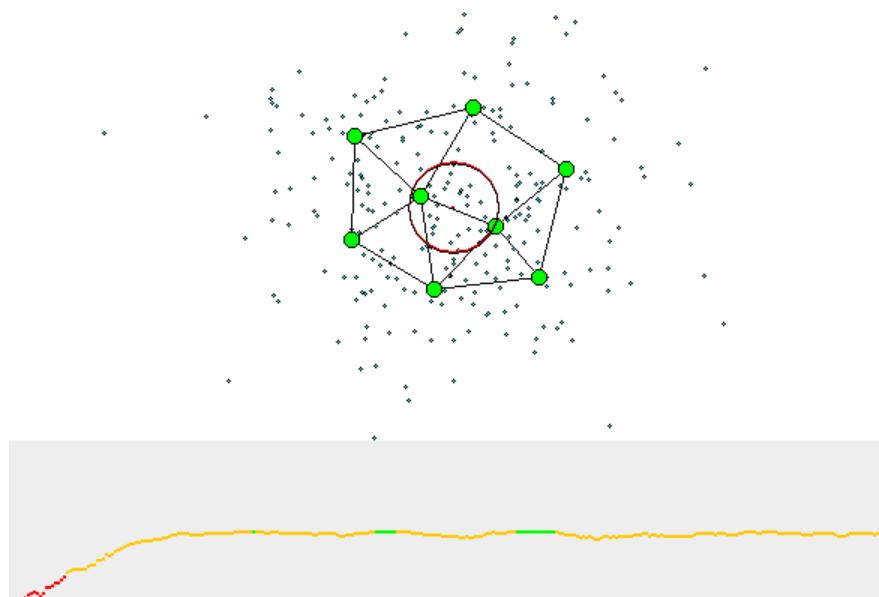


Figure 3.2: Screenshot showing the process of searching optimal variable values. The visualization has been made for testing purposes in the first place, but it nicely shows what had been lately parallelly computed. The bottom part of the picture shows the SCORE value throughout the simulation. (Red color means $SCORE > 0.05$, orange is $SCORE > 0.001$ and green is $SCORE \leq 0.001$)

- $\alpha \in \{0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0\}$
- $\beta \in \{0.0, 0.00001, 0.00005, 0.0001, 0.001, 0.005, 0.01, 0.5, 0.1, 0.5, 1.0\}$
- $\epsilon_{winner} \in \{0.0, 0.001, 0.005, 0.01, 0.1, 0.2, 0.5, 1.0\}$
- $\epsilon_{neighbour} \in \{0.0, 0.0001, 0.0006, 0.001, 0.005, 0.05, 0.1, 0.2\}$
- $maxNodes \in \{4, 8, 16, 32\}$

Figure 3.3: Domains of variables for the experimental learning of best values.

α	β	ϵ_{winner}	$\epsilon_{neighbour}$	$numNodes$	$SCORE$
0.0	1.0	0.0050	0.0	16	$3.8 * 10^{-12}$
0.5	0.0	0.01	1.0E-4	8	$5.1 * 10^{-12}$
0.5	0.0010	0.1	0.0010	8	$8.8 * 10^{-12}$
0.5	1.0	0.0	1.0E-4	8	$3.1 * 10^{-12}$
0.8	1.0E-5	0.0010	1.0E-4	32	$7.4 * 10^{-12}$
0.8	1.0E-5	0.0050	6.0E-4	8	$4.3 * 10^{-12}$

Table 3.1: Variable values with best average SCOREs

the environment into grid 4x4, where each cell works as the place in [5].

Each cell is given two variables *positive* and *negative* both of which are set to zero and increased throughout the simulation. When an agent sees at least half of that area determined by the cell, if he sees any food, he increase the *positive* variable. If the agent search for food and he cannot see any, he increase the *negative* variable.

When the grid is later asked whether there is food at specific cell, it answers according to this method with parametr α to be found:

$$ANSWER = \alpha \times positive - negative \quad (3.1)$$

Similary I will use this structure to keep spatial information about the environment in the simulation.

Chapter 4

Simulation and used memory architectures

In this chapter I will describe the simulation, environment and agent's reasoning and communication how it is used in later experiments.

4.1 Simulation

The **simulation** consists of a set of agents, a set of generators and a set of pieces of food. According to given settings it sequentially processes a number of steps, each of which invokes an agents' life step and eventually generating new food.

It can also contain a couple of monitors which observe the environment or agents.

4.2 Environment

The environment is a two-dimensional space which contains agents and food. Agents can move around and eat the food which is randomly distributed using the food generators.

4.3 Agent

As I mentioned previously an agent is an entity in the environment which moves and interact with the world around. The interaction is done through eating

food which is a part of the environment and through communication with other agents. The latter one actually changes agents' beliefs about the environment.

Agent has his needs which influences his decisions as fulfilling his needs keeps him alive. When his internal variables of needs is higher than?

There are four types of agents each of which is different in the way they decide about next step. If one is hungry and sees a food (i.e. there is a piece of food in the sight distance) then they choose to go after this food. If there is no desired food around they go searching for it and that is when differs the agents' actions.

- *random agent* moves randomly around the environment,
- *pure reactive agent* sees the whole environment, i.e. they always sees a desired piece of food,
- *grid agent* implements a memory based on clustering the space into a grid,
- *GNG agent* implements a memory based on growing neural gas.

4.4 Communication

Apart from what agent sees, there is another way how the agents gather information about the environment. They communicate. It is quite simple way of sharing information. When suggesting an implementation for communication I had to create a unified protocol which could have been used throughout types of agents. Thereby I have tried to have this communication protocol as simple as possible.

Moreover, although all agents have a kind of knowledge about the environment they are not able to answer easily, when they are asked about a specific food location. Since the food appears in environment according to given normal distribution, it is not clear what should be an answer for such question. A couple of possible kinds of answers follows.

First and the most simple answer might be saying exact X,Y coordinates of the food location as it is stored in agent's mind. Additionally, there would be a noise added to such an answer, having in mind that the answer should not be perfect and there is always a distortion and imperfection in our answers based on how a person is certain about his answer.

Another way and possibly more plausible one might be answering by a direction (an angle) with an approximate distance. What both the first suggested

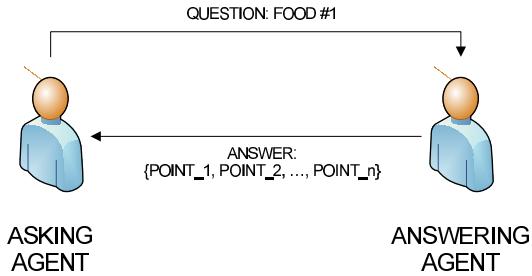


Figure 4.1: Simple communication protocol

XY answer and this one have in common is the answers are hard to combine with the learning method used in GNG memory. GNG works with samples of data which sequentially influence the neural network. Both kinds of answers could be used if agents would ask more often or the agent's answer would be a sample of points rightaway.

Having such conditions I have suggested and implemented a communication 4.2 where the answer consists of several sample points which are generated according agent's knowledge.

4.5 Decision making

While searching for food each type of agents makes the decision where to go next. This process is either done randomly or following ones knowledge of the environment around. A simple diagram of decision making follows.

The diagram 4.2 shows how an agents decides what to do. In fact it is common for all types of agents described in this thesis, although the first step "Put known food location into memory" is omitted in case of *random* and *PR agents*.

4.6 Memories

There are two types of memory which should allow agents to improve their lifespan comparing to a random agent. Those are memories based on a growing neural gas and a spatial grid.

The *GNG memory* uses a self-teaching neural network which has been described in 3.1. The neural network allows the agent to learn approximate location where the food is distributed. Each food kind is given a single neural network which tries to learn the distribution reflecting the data inputs.

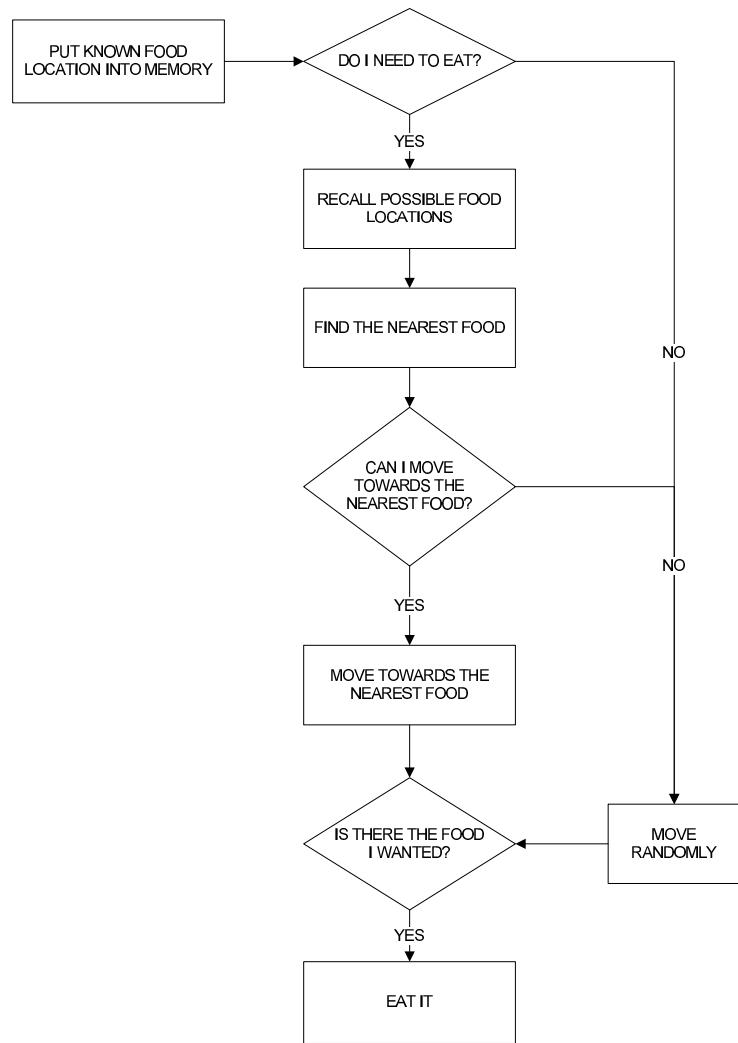


Figure 4.2: How the agent decides what to do next

The *grid memory* divides the environment into a grid so as to simplify the space and restrict total size of data structure used to describe the space.

Chapter 5

Implementation

5.1 Simulation application

Simulation was implemented as a Java application using JDK 1.6.

5.2 Memories

Each memory implementation implements following IMemory interface:

```
interface IMemory {  
    Learn(integer foodKind, Point[] locations);  
    Point[] GetSample(integer foodkind);  
    (Point, integer) GetExpectedGauss(int foodkind);  
}
```

Through the *Learn method* the memory gets new inputs and thus it learns. *GetSample method* returns a sample of food locations according to the agent's believes, it means that based on the information in memory it tries to generate several samples. *GetExpectedGauss method* returns expected Gaussian distribution (x,y location and variance) for the food kind given as a parameter.

The expected Gaussian distribution is not calculated every time the *GetExpectedGauss method* is called, but it precalculated after the new inputs are processed.

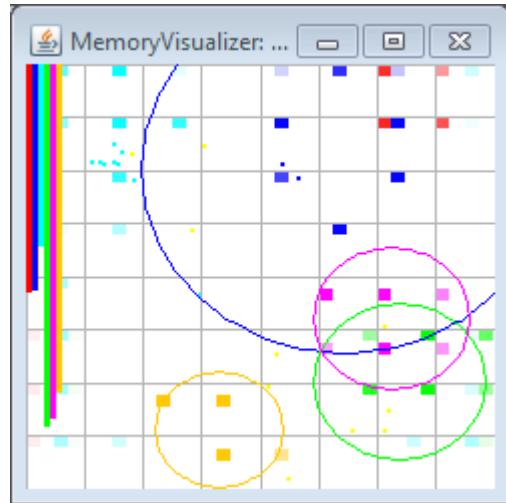


Figure 5.1:

5.3 Growing neural gas as a memory

Growing neural gas has been explained previously in 3.1. In this section I will explain how that algorithm was used to learn positions of food sources. For this purpose the GNG uses five nodes.

For each kind of food there is a separate GNG engine which learns the believed location of such food.

```
procedure Learn(integer foodkind , Points[] locations )
    gngEngine <- GetEngineByFoodKind(foodkind)
    gngEngine . SetDiscreteSignals(locations )
end
```

SetDiscreteSignals method assigns known food locations (array of x,y points) to the neural network so as to be later processed using the GNG algorithm described in 3.1. At the end of day, i.e. end of each simulation step, the learnt information is processed by the GNG.

When the memory is asked about expected Gauss distribution, it uses the five nodes in the GNG to compute it.

5.4 Grid as a memory

As in GNG memory implementation, the grid memory uses *Learn method* to update current state.

```

procedure Learn( integer foodkind , Points [] locations )
    grid <- GetGridByFoodKind( foodkind )
    hasInput <- CreateEmptyMatrix( cols , rows )

    for (x, y) in locations do
        begin
            (gridX , gridY) <- GetGridCoords(x, y)
            grid [gridX , gridY]++;
        end

        for (i , j) in grid do
            begin
                cell <- grid [i , j]
                if hasInput [i , j] > 0 then
                    cell . IncPositive()
                else
                    if node is in sight distance then
                        cell . IncNegative()
            end
    end

```

An actual value of the cell is computed using following formula:

$$value = positive - negative/\alpha \quad (5.1)$$

if($value < 0$) $value = 0$

α is set to 2.

The gaussian distribution is computed following way:

- x, y position is computed as a weighted mean of cells:

$$\frac{\sum CellValue_i \times (x_i, y_i)}{\sum CellValue_i} \quad (5.2)$$

- variance is computed as normalized sum of weighted distances:

$$\frac{\sum DistanceToCenter_i \times CellValue_i}{\alpha \times NumPositiveCells \times MaxCellValue} \quad (5.3)$$

Chapter 6

Experiments

6.1 Experimental settings and methodology

All following experiments are run using a default setup as it is described in this section. Each of the experiments is run on a quadcore *Intel Core i5 with 2,4 GHz and 6 GB RAM*.

Environment is set to be a square matrix with *64 x 64 dimension*. All agents start in the middle of the environment. There are *six kinds of food* which are randomly positioned in the environment and which generate a piece of food each *50 steps*.

Since an environment contains of six food kinds, an agent has six internal variables for each such food kind (see 6.1). By default they are set to 0 and are increased by *0.001 each step* in simulation. When they are equal to 1 (or higher), the agent dies.

6.2 Homogeneous agent set comparision with communication

In this experiment I will compare avarage life span and efficiency of groups which contains of agents with only single type of memory. Thereby you can see which of the used memory implementation works better in memory homogeneous environment.

What I assume is the *random agents* are about to expire almost immediately as they had no chance to find all the food. While the *PR agents* should approach their goals easily, thereby they will stay alive. Both results of *GNG agent* and

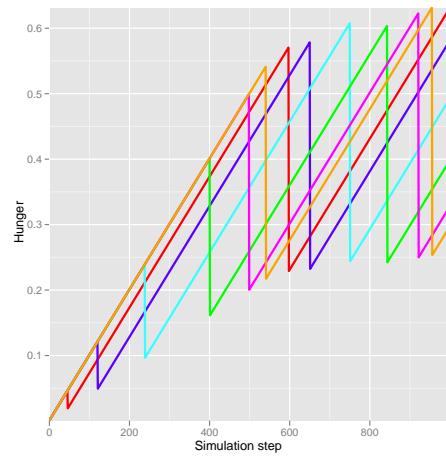


Figure 6.1: An example of agent's first 1000 steps showing the needs for each kind of food.

grid agent are matters of the experiment and I can only expect them not to be worse than *random agent* and not to be better than *PR agent*.

6.2.1 Random agent

As for the random agents I assumed that they will immediately pass away without communication. And as you can see in 6.2(a) it happened to be true.

On the other hand, if I allow them to communicate with each other it might happen they improve their chances. Although they will not last through the entire simulation (see 6.7(a)), the result is that they have managed to slow down [it] (see 6.3).

Agent kind	median	mean	min	max
Random agents with communication	1	0.939	0.55	1
Random agents without communication	1	1	1	1

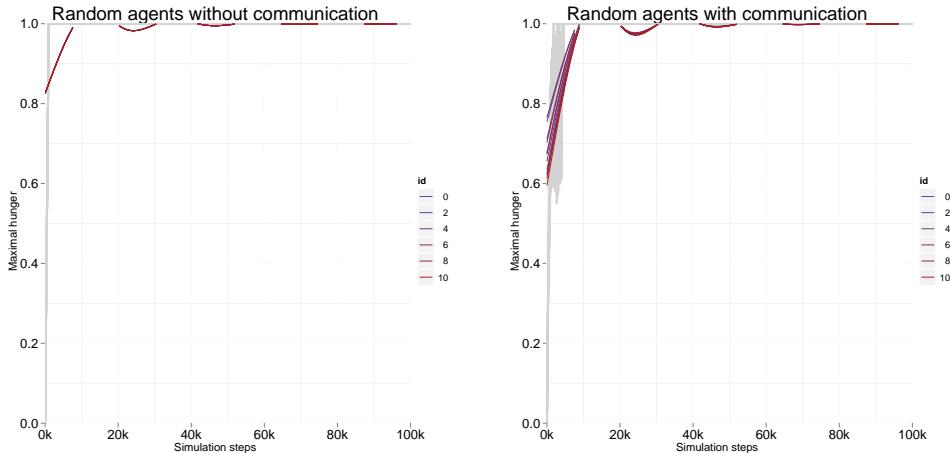


Figure 6.2: Random agent *without communication* fades out quickly.

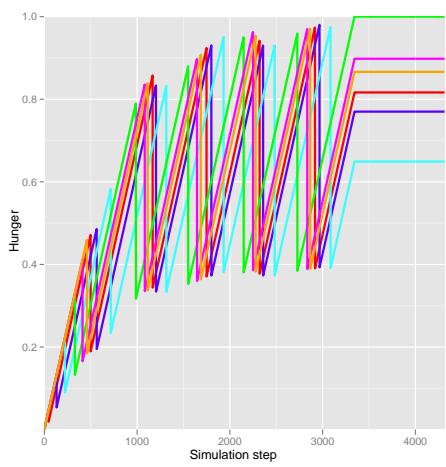


Figure 6.3: A beginning of life of a random agent with communication.

6.2.2 PR agent

In case of the *pure reactive agents* there is not much to compare between simulation with and without the communication, because the communication is not used since the PR agents see all the environment. So both graphs 6.4(a) and 6.4(b) are the same.

Agent kind	median	mean	min	max
PR agents with communication	0.69	0.689	0.54	0.86
PR agents without communication	0.69	0.689	0.54	0.86

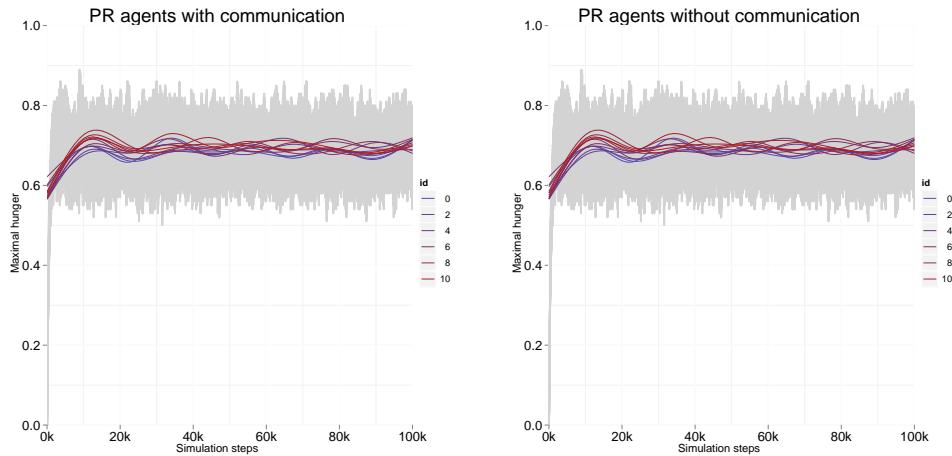


Figure 6.4: PR agent with and without communication.

6.2.3 GNG agent

GNG agents need more information to be able to learn and to survive that is why I do not expect them to deal with it well. In case of a simulation without communication they might end up similarly to random agents, they should do better if they communicate.

Agent kind	median	mean	min	max
GNG agents with communication	0.69	0.683	0.49	0.83
GNG agents without communication	1	1	1	1

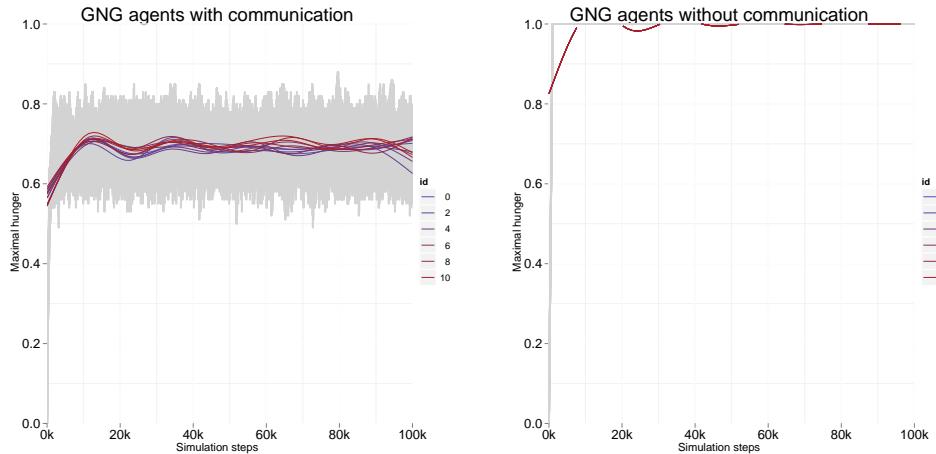


Figure 6.5: GNG agent with and without communication.

6.2.4 Grid agent

I assumed they will do similarly like GNG agent, i.e. they fail to survive without communication, which assumption ended up to be wrong. Grid agents are able to learn quickly the environment just by moving around randomly and there is the chance that a couple of them survive.

I have run 20 simulations with grid agents to verify the probability of grid agent's learning the environment. As in the standard experiments there were 12 agents and the following values shows the result of the tests:

$$mean=5.56, median=5.5, min=1, max=12$$

Agent kind	median	mean	min	max
Grid agents with communication	0.81	0.826	0.54	1
Grid agents without communication	0.65	0.771	0.44	1

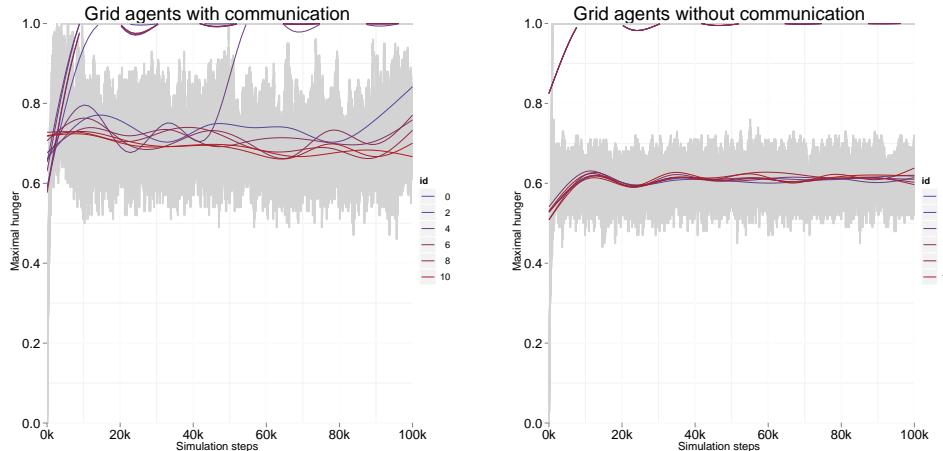


Figure 6.6: Grid agent with and without communication.

6.3 Mixed environment

Second part of the experiments are about environment where different kinds of agents are trying to fulfill their needs. I will observe whether some kind of agents prosper from a presence of other agents or if they are not that successful as they were in a homogeneous environment in previous experiments.

6.3.1 GNG+Grid+PR+Random agents

First I will start with combination of all kinds of agents, whereby since there are those PR agents the others have an advantage of the perfect source of information.

Results of a simulation with communication follow:

Agent kind	median	mean	min	max
GNG agents	0.7	0.699	0.57	0.86
Grid agents	0.7	0.697	0.54	0.83
PR agents	0.69	0.689	0.55	0.81
Random agents	0.7	0.704	0.57	0.87
All agents	0.7	0.698	0.54	0.87

Results of a simulation without communication follow:

Agent kind	median	mean	min	max
GNG agents	1	1	1	1
Grid agents	0.61	0.721	0.47	1
PR agents	0.58	0.576	0.44	0.68
Random agents	1	1	1	1
All agents	1	0.824	0.44	1

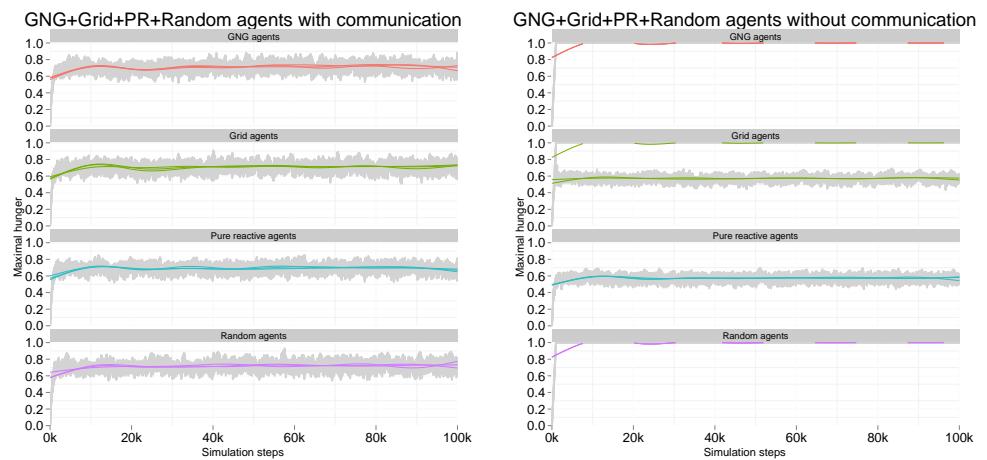


Figure 6.7: GNG, grid, PR and random agents together. Owing to PR agents and the communication they all do well.

6.3.2 GNG+Grid+Random agents

In this experiment I have omitted pure reactive agents and thus left others on their own. For better comparision I have added the difference between values in the current experiment and the previous one (GNG+Grid+PR+Random). I used red colour for a decrease and green colour for an increase, although it is an improvement if they values are lower.

Agent kind	median	mean	min	max
GNG agents	0.69 -0.01	0.691 -0.008	0.52 -0.05	0.86
Grid agents	0.7 +0.003	0.700 +0.01	0.55 +0.04	0.87
Random agents	0.7 -0.002	0.702 -0.02	0.55 -0.02	0.85
All agents	0.7 -0.02	0.698 -0.02	0.52 -0.02	0.87

Agent kind	median	mean	min	max
GNG agents	1	1	1	1
Grid agents	1 +0.39	0.747 0.026	0.39 -0.08	1
Random agents	1	1	1	1
All agents	1 +0.092	0.916 -0.05	0.39	1

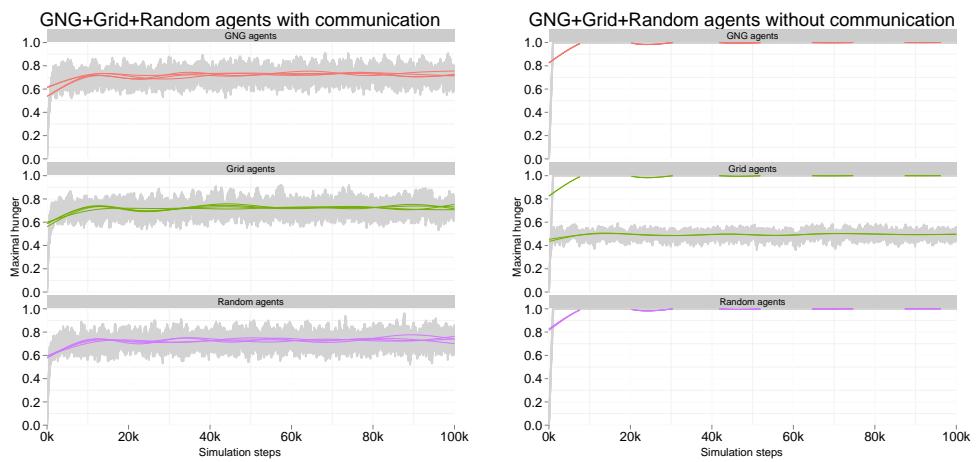


Figure 6.8: GNG, grid, PR agents together with and without the communication.

6.3.3 GNG+Grid agents

In this experiment I have observed a simulation with just memory agents in it. Again you can see highlighted differences between new values from this experiment and the values from the previous one (GNG+Grid+Random).

Agent kind	median	mean	min	max
GNG agents	0.72	0.715	0.5	0.89
	+0.03	+0.024	-0.02	+0.03
Grid agents	0.72	0.721	0.57	0.89
	+0.02	+0.021	+0.02	+0.02
All agents	0.72	0.718	0.5	0.89
	+0.02	+0.020	-0.02	+0.02

Agent kind	median	mean	min	max
GNG agents	1	1	1	1
Grid agents	1	0.755	0.4	1
	+0.008	+0.01		
All agents	1	0.877	0.4	1
	-0.039	0.01		

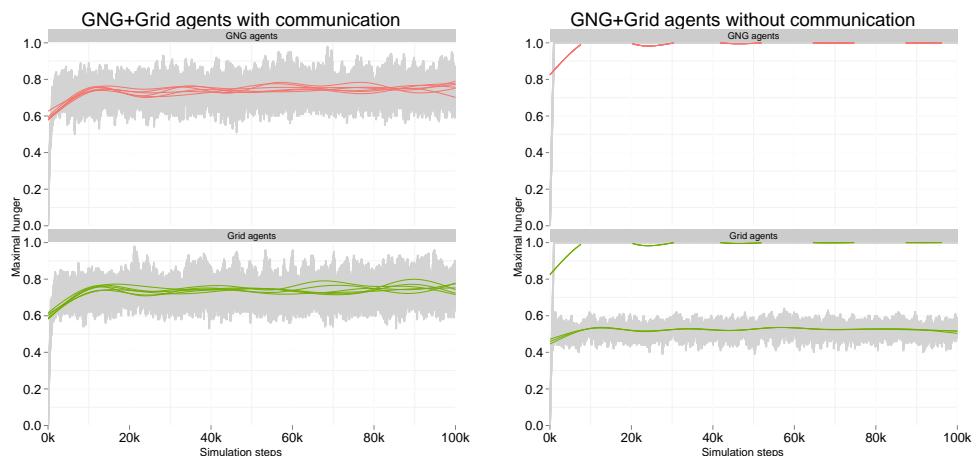


Figure 6.9: GNG and grid agents together.

Chapter 7

Discussion

In previous chapter you have seen experimental runs of the simulation where I have compared the level of overall hunger of agents throughout the simulation's steps. Now I am going to sum up those results and discuss possible outcomes.

Simulation settings (size of the environment, number and distribution of resources) allowed random agents to survive few steps when they could communicate with each others. Thereby the communication was strong tool for agents how to improve their results.

The communication was not always an improvement as you can see in 6.6 where the grid agents were less efficient when they were allowed to share information about the environment. The communication brings more income information and thus increases the ration between positive and negative variables (see 3.2) and disorients agents. Owing to communicatio it is hard to decide which of gng and grid agent was better.

As I have already clarified the grid agent has perfect results without using communication as soon as he survives first thousand steps. On the other hand, he obviously fails to use communication as an improving factor. That is something where the gng agents win.

Second part of experiments is about comparing simulation with more kinds of agents at once. Thereby I could observe how they compete against each others. So I have observed GNG+Grid+PR+Random, GNG+Grid+Random and GNG+Grid. Apart experiments with communication, I have also run simulation whereby the agents were not able to communication, although in such conditions the result could not be different from single-kind setups.

Since there are pure reactive agents in the first experiment (GNG+Grid+PR+Random) the agents have a perfect source of information and thus easily succeeded. Hav-

ing omitted PR agents I could have observed that gng agents have bettered and, on the other hand, the grid agent become worse. Furthermore, when you look at results of GNG+Grid simulation where the random agents are missing, you can see that both grid and gng agents become worse, although the random agents should be much helpfull.

What is different between random agents and memory agents is the random agent answers only correct positions, when he is asked, and the memory agent usually answers using his believes.

There were minor problems I had to deal with, for example I had to setup variables for the GNG algorithm. What I have learnt from that is the algorithm is expected to slowly converge to the learnt topology. On the other hand, I have used it for unncertain dynamically changing data which were mostly based on other agents' believes. That is probably why much simpler memory structure used for grid agents has better results.

Chapter 8

Conclusions

I have created a multi-agent simulation with four different kinds of agents each of which differs in their approach to fulfill their needs. What they had to succeed in was they were put inside a two dimensional environment whereby they had to learn positions of six food resource so as to be able to survive. They were pure reactive agent, random agent, GNG agent and grid agent. Latter two had a memory to learn those positions. GNG agent used implemantion of growing neural gas, an unsupervised neural network, and grid agent used data structure inspired by [5].

In the experiments I have compared those agents in their efficiency. First I have set up environments in which was only a single kind of agent and after that I have combined different kinds of agents together. The results of their efficiency measured by the overall hunger were presented in graphs and statistical variables such as mean, median, maximal and minimal value.

Bibliography

- [1] PhD thesis.
- [2] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. In K. W. Spence and J. T. Spence (Eds.), *The Psychology of learning and motivation: Advances in research and theory* (vol. 2)., pages 89 – 105, 1968.
- [3] Richard C Atkinson and Richard M Shiffrin. The control of short-term memory. *Scientific American*, 225(2), 1971.
- [4] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3):349–355, 1988.
- [5] Cyril Brom, Tomáš Korenko, and Jiří Lukavský. How do place and objects combine? "what-where" memory for human-like agents. In *Proceedings of the 9th International Conference on Intelligent Virtual Agents*, IVA '09, pages 42–48, Berlin, Heidelberg, 2009. Springer-Verlag.
- [6] Bernd Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, 1995.
- [7] Wan Ching Ho, Kerstin Dautenhahn, and Christopher L. Nehaniv. Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connect. Sci.*, 20:21–65, March 2008.
- [8] T. M. Martinetz and K. J. Shulten. A "neural-gas" network learns topologies". In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 397–402. North-Holland, Amsterdam, 1991.

- [9] Thomas Martinetz. Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. In Stan Gielen and Bert Kappen, editors, *Proc. ICANN'93, Int. Conf. on Artificial Neural Networks*, pages 427–434, London, UK, 1993. Springer.
- [10] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall, 3 edition, December 2009.
- [11] Robert J. Sternberg. *Cognitive Psychology*. Wadsworth Publishing, August 2002.
- [12] A. Tapus, G. Ramel, L. Dobler, and R. Siegwart. Topology learning and recognition using bayesian programming for mobile robot navigation. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3139 – 3144 vol.4, sept.-2 oct. 2004.
- [13] Hal R. Varian. *Intermediate microeconomics : a modern approach / Hal R. Varian*. W.W. Norton, New York :, 1987.