Master Thesis

Hide and seek games

P.G.A.M.G Uyttendaele

Master Thesis DKE 09-16

Thesis submitted in partial fulfillement
of the requirements for the degree of Master of Science
of Operations Research at the Department of Knowledge
Engineering of the Maastricht University

Thesis Committee:

Dr. F. Thuijsman
Dr. J. Derks
Dr. ir. J. Uiterwijk

Maastricht University
Faculty of Humanities and Sciences
Department of Knowledge Engineering
Master Operations Research

June 23, 2009

Contents

Preface

In this article we are interested in finding a way to model a classical Hide and Seek Game. Two main approaches will be presented, a simulation based approach and a theoretical one based on a Markov chain procedure. The model will be built in two parts, the first one where the board is a cycle board represented as a sequence of rooms where the last one is connected to the first one. The second part will be a 2-dimensional board where one can imagine that a number of these cyclic games are on top of each other and they are all connected to the one above and the one below. After the evaluation of the performance of the models, they will be compared to related literature. Indeed, one major application of the Hide and Seek Game are predator/prey models. In terms of predator prey models, several studies are based on simulating the interaction of predator and prey, but also are made to understand their behavior. To some extend, we are going to explore the best strategy that a predator could adopt, but also argue some modeling choices.

This article is divided into two parts. The first one is the creation of a model to play a Hide and Seek game, and the second one is comparing the created model with the model of I. Scharf et al. [6].

Numerous thanks for the support in the development of this article needs to be made: thanks to Frank Thuijsman for the collaboration and the guidance on the subject, thanks to Jean Derks for the preliminary work done on the field [3] as well as the collaboration and the guidance on the subject, thanks to A. Bouskila for the article and the explanation on the subject, and finally, thanks to I. Scharf for the source code and the explanation on the previous work done.

Part I

Setting up a model

Chapter 1

Introduction to the model

1.1 Original set up

We are going to use a board that has a cyclic graph shape as a set of the rooms where the Searcher ($\mathcal{S}$) and the Hider ($\mathcal{H}$) are going to play a Hide and Seek game. The board on which the game is played will be referenced as $\mathcal{B}$. $\mathcal{B}$ is represented as a set of rooms $1, 2, ..., n$ where each room i is connected to the next $i + 1$ and the previous one $i - 1$. Also, to close the circle, room n is connected to room 1 therefore room $n + 1 = 1$ and room $1 - 1 = n$. This to avoid the problem of being in an endpoint, and to allow both players to get to any point on the board in at most $\frac{n}{2}$ steps. A similar approach has been used by Zollner and Lima [10], in previous studies.

We assume that $\mathcal{H}$ will hide in a random room as starting point, and that at every moment $t = 1, 2, ...$ he will do one of the following: move to the next room (go to room $i + 1$), move to the previous room (go to room $i - 1$), or stay in the current room with some positive probability. These three choices will be made with respective probabilities p, q, and $(1 - p - q)$.

The same movement procedure applies to $\mathcal{S}$. He may only move to room $i + 1$, $i - 1$ or stay in room i for his next search. However, for the searcher, the movement procedure is not probabilistic, he can select his strategy the way he wants and eventually change during the game.

If $\mathcal{S}$ searches in room i and $\mathcal{H}$ is hidden in that room, he will find $\mathcal{H}$ with probability h_i.

$\mathcal{S}$ has, at every moment t, the possibility to search one room. He can only search the room he is visiting in t, which depends on his movement strategy. His starting point at $t = 1$ can be selected the way he wants.

1.2 Assumptions

In order to have a well defined model, many assumptions are made. Hit probabilities have been introduced in [8], as the probability for $\mathcal{S}$ to find $\mathcal{H}$ when both players are in the same room. Hit probabilities can change depending of the room. The first assumption is that all room hit probabilities will be $h_i = 1$.

Also, we assume that the moving probabilities p and q are known by $\mathcal{S}$, where $\mathcal{H}$ has no knowledge about the strategy of $\mathcal{S}$.

Our last assumption is that $\mathcal{S}$ has limited information on the room where $\mathcal{H}$ is hidden. At every t, after his search, if $\mathcal{S}$ has searched the room where $\mathcal{H}$ was hidden or one of the adjacent rooms, he will be informed of the room where $\mathcal{H}$ was hidden, otherwise he will not get any information. The strategies will be referenced as a triple (*previous*, *next*, *unknown*). Each of the values of that triple consists of the move to perform (i+?) when in one of the three situations. *previous* is the situation where $\mathcal{H}$ is in the adjacent room $i-1$ (or n if $i = 1$), *next* when $\mathcal{H}$ is in the adjacent room $i+1$ (or 1 if $i = n$) and *unknown* when the position of $\mathcal{H}$ is unknown (not in $i-1$ or $i+1$). An example of triple is : $(1, -1, 0)$ [read go $i+1$ when $\mathcal{H}$ is in the previous room, go to $i-1$ when $\mathcal{H}$ is in the next, else stay in room i].

1.3 Goal

There can be many goals for $\mathcal{S}$ but we limited ourselves to these two:

- Find $\mathcal{H}$ as fast as possible
- Find $\mathcal{H}$ in a limited amount of time. We are not interested how fast it is found as long as it is within the given period

Boards containing 2 or 3 rooms will not be explored because these are less interesting. Indeed, in every step, each room can be reached and also $\mathcal{S}$ will always have full information about the position of $\mathcal{H}$.

Chapter 2

Circle Approach

Several ways to solve a hide and seek game can be seen. This article treats two different approaches and compares them. These two different approaches are a simulation based model and an analytical model represented as a Markov chain.

Both models will be compared on two bases:

- The average time needed by $\mathcal{S}$ to find $\mathcal{H}$ with a given set up and strategy;
- The probability for $\mathcal{S}$ to find $\mathcal{H}$ within a given time period.

2.1 Simulation

The simulation based approach is used because it is a way to see the behavior of many strategies and its accuracy can be verified very easily.

The simulation is a straight forward process. A number of rooms is entered with the different parameters for the players and then the simulation is launched. At first a random room is selected as starting point for the two players and then the game starts. As the strategy for $\mathcal{S}$ is determined when in one of the three different situations ($\mathcal{H}$ in previous, next or unknown room), it is just an application of this movement. For $\mathcal{H}$ it is just a random choice distributed according to p and q to know in what room to go next. Once both players are in the same room, the process is stopped and the number of steps needed to reach that situation is computed.

Given this procedure, one can easily derive the average number of steps as well as the average number of steps within a given time period. A slight warning can be given to the user that is trying to find the best strategy and

tries all the strategies: if $\mathcal{S}$ has as strategy (1,-1,0), then the time to find $\mathcal{H}$ is unlimited if both players are not in the same room as starting point. Indeed $\mathcal{S}$ is fleeing $\mathcal{H}$. Every time $\mathcal{S}$ sees $\mathcal{H}$, he is going to a room that is unreachable by $\mathcal{H}$ and when he does not see $\mathcal{H}$, he waits until he sees him coming and starts fleeing him again.

2.2 Markov Chain

The hide and seek game with the rules and assumptions explained in section 1.1 and 1.2 can be explained and solved in terms of a Markov chain.

A Markov chain is a stochastic process $\{X_n, n = 0, 1, 2, ...\}$ that takes on a finite or countable number of possible values. If $X_n = i$, then the process is said to be in state i at time n. We suppose that whenever the process is in state i there is a fixed probability P_{ij} that it will be next in state j. These probabilities are assumed not to change over time. A matrix containing all the P_{ij} can be made. It is called the transition matrix and is referenced as P. [5]

$$P = \begin{pmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ \vdots & \vdots & \vdots & \\ P_{i0} & P_{i1} & P_{i2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The Chapman-Kolmogorov Equation [4] provides a method to compute the n-step transition probabilities. These equations are

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^{n} P_{kj}^{m} \qquad \forall n, m \geq 0, \forall i, j \tag{2.1}$$

From these equations, it can be generalized that P^n gives the n-step transition probabilities for all the states. Each entry P_{ij}^n represents the probability to be in state j after n steps starting in i.

In matrix P some of the states are said to be transient. A state i is said to be transient if, given that we start in state i, there is a positive probability that we will never return to i. If a state i is not transient, then it is called recurrent or persistent. The matrix consisting of those rows and columns that corresponds to the transient states is called P_T [2]. With the help of this matrix P_T the mean time spent in each of the states can be computed with the equation

$$S = (I - P_T)^{-1}$$

If one leaves a persistent state or a recurrent set in the matrix P_T, the inversion of the matrix $(I - P_T)$ will be impossible to compute as the matrix will be singular. A singular matrix is a matrix from which the determinant is 0.

Each entry S_{ij} shows the average number of time that state j is visited when starting in state i before being absorbed in a recurrent state. Similarly,

$$\sum S_{ij}, \qquad j \in T$$

is the average number of visits before absorption when starting in a transient state i. To get the average number of visits before absorption when starting in an arbitrary state (including the non-transient states) where the starting probability distribution among all states is $\frac{1}{n}$ for all the room, we need to calculate

$$\frac{1}{n} \sum \sum S_{ij}, \qquad i, j \in T, \text{ and } n \text{ the number of states in } P$$

We need to divide by n because in this game, both players are placed in one of the states randomly, including the absorbing states. Also, the average number of visits when we start in an absorbing state is 0. This small parameter needs to be taken into account when computing the average number of steps before absorption.

The matrix P_T is a modified matrix P as the recurrent states are not present. Indeed, it gives the average number of visits before absorption when starting in state i, but when we start in a state that is already absorbing, the average number of steps before absorption is 0. For the next step we need to have all the states, therefore we modify all absorbing states from the matrix S. Instead of having a value $P_{kl} = 1$ for the absorbing states, we change them to $P_{kl} = 0$. We will call this new matrix with modified values P_M.

By combining our knowledge of the Chapman-Kolmogorov equation 2.1 and the transformed matrix P_M we can easily compute the probability of absorption within t steps. Still according to the Chapman-Kolmogorov equation, taking the t power of the matrix P will give the probability distribution of being in any state after t steps for each possible initial state. In our P_M matrix, the absorbing states have been modified, which means that after t steps the probability to be in one of these states will be 0. Thus, taking the t

probability distribution P_M^t will give us the probabilities to be in a transient state after t steps. As we want to know the probability to be absorbed after t steps, we will compute 1 - (Probability of still being in a transient state). Also, each row of P_M^t will give us the probability of still being in the system after t steps starting in room i. As we select our starting point randomly with probability $\frac{1}{n}$ for both players, we compute

$$1 - \left(\frac{1}{n} \sum_i \sum_j P_M^t \right) \qquad i, j \in P_M$$

If the selection of the room in which the players starts was not uniformly distributed we would compute

$$1 - \left(\sum_i Pr\left(\text{start in } i\right) \left(\sum_j P_M^t \right) \right) \qquad i, j \in P_M$$

Also we can easily compute the probability of absorption in exactly t steps by computing

$$1 - \left(\frac{1}{n} \sum_i \sum_j \left(P_M^t - P_M^{t-1} \right) \right) \qquad i, j \in P_M$$

As explained before, the hide and seek game can be written into a Markov chain. Each state is referenced as (i, j) where j is the room where $\mathcal{H}$ is hidden and i the room where $\mathcal{S}$ is searching. The probability matrix P depends on the values of p and q but also on the search strategies of $\mathcal{S}$.

Let's assume that we are in the case of a 3 room game, that $\mathcal{H}$ has for probabilities $p = \frac{1}{6}$ and $q = \frac{1}{3}$ and that the strategy for $\mathcal{S}$ is (-1,1,0) then

$P =$

	$(1,1)$	$(1,2)$	$(1,3)$	$(2,1)$	$(2,2)$	$(2,3)$	$(3,1)$	$(3,2)$	$(3,3)$
$(1,1)$	1	0	0	0	0	0	0	0	0
$(1,2)$	0	0	$\frac{1}{2}$	0	0	$\frac{1}{6}$	0	0	$\frac{1}{3}$
$(1,3)$	0	$\frac{1}{2}$	0	0	$\frac{1}{6}$	0	0	$\frac{1}{3}$	0
$(2,1)$	0	0	$\frac{1}{3}$	0	0	$\frac{1}{2}$	0	0	$\frac{1}{6}$
$(2,2)$	0	0	0	0	1	0	0	0	0
$(2,3)$	$\frac{1}{3}$	0	0	$\frac{1}{2}$	0	0	$\frac{1}{6}$	0	0
$(3,1)$	0	$\frac{1}{6}$	0	0	$\frac{1}{3}$	0	0	$\frac{1}{2}$	0
$(3,2)$	$\frac{1}{6}$	0	0	$\frac{1}{3}$	0	0	$\frac{1}{2}$	0	0
$(3,3)$	0	0	0	0	0	0	0	0	1

Keeping only the transient states will give us the matrix

$$P_T = \begin{array}{c|cccccc} & (1,2) & (1,3) & (2,1) & (2,3) & (3,1) & (3,2) \\ \hline (1,2) & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 \\ (1,3) & \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{3} \\ (2,1) & 0 & \frac{1}{3} & 0 & \frac{1}{2} & 0 & 0 \\ (2,3) & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 \\ (3,1) & \frac{1}{6} & 0 & 0 & 0 & 0 & \frac{1}{2} \\ (3,2) & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{2} & 0 \end{array}$$

2.3 Choice of an approach

The simulation approach and the Markov approach give similar results. However, they both have their strengths and weaknesses. The Markov approach gives an answer in a very short time. However, it uses n^4 entries to generate the matrix P, which means that if we have 100 rooms, the matrix is so big that it requires too much memory to generate it and compute P^n. The simulation approach is a fast process for a single game. However, a lot of games needs to be played in order to have a reliable average. When we have a lot of rooms, it also needs lots of time but much less memory for the computation than for the matrix computation. The choice of the method is just a trade off between time and memory.

Chapter 3

2-dimensional Board

The 2-dimensional board game is an extension of the circle game presented in Chapter 2. One can see the board as a series of circle games with the same number of rooms placed on top of each other.

3.1 Extension of the set up

The original set up for the 2-Dimensional board is similar to the one given in 1.1.

The board ($\mathcal{B}$) consists of a number of rows ($y = 1, ..., m$) and columns ($x = 1, ..., n$). Each room will be referenced as $\mathcal{B}(x, y)$. We consider that there is no "end" in the board, therefore each room is connected to its corresponding adjacent: the room at $x - 1$ when $x = 1$ is $x = n$, and the room at $x + 1$ when $x = n$ is $x = 1$. The same identities hold for y. One can now see $\mathcal{B}$ as a torus.

We assume that $\mathcal{H}$ will hide in a random room as starting point, and that at every moment $t = 1, 2, ...$ he will move to one of the adjacent rooms or stay in his room. The probabilities with which he will move will be referenced as: (p, q, r, s) where p and q are the movements on x, r and s are the movements on y. Here p will be the probability to go in $\mathcal{B}(x - 1, y)$, q to go in $\mathcal{B}(x + 1, y)$, r to go in $\mathcal{B}(x, y - 1)$, s to go in $\mathcal{B}(x, y + 1)$ and finally $(1 - p - q - r - s)$ is the probability to stay in $\mathcal{B}(x, y)$.

The same movement procedure applies to $\mathcal{S}$. He may only move to room $\mathcal{B}(x - 1, y)$, $\mathcal{B}(x + 1, y)$, $\mathcal{B}(x, y - 1)$, $\mathcal{B}(x, y + 1)$ or stay in room $\mathcal{B}(x, y)$ for his next search. However, the movement procedure is not probabilistic, he can select his strategy the way he wants and eventually change during the game.

If $\mathcal{S}$ searches in room (x, y) and $\mathcal{H}$ is hidden in that room, he will find $\mathcal{H}$ with probability $h_{x,y}$.

$\mathcal{S}$ has, at every moment t, the possibility to search precisely one room. He can only search the room he is visiting at time t, which depends on his movement strategy. His starting point at $t = 1$ can be selected the way he wants.

3.2 Assumptions

The same assumptions as in 1.2 are made, with the adaptation that it fits the board game.

The first assumption is that all room hit probabilities will be $h_{x,y} = 1$. Hit probabilities have been introduced in [8], they are the probabilities for $\mathcal{S}$ to find $\mathcal{H}$ when both players are in the same room. Hit probabilities depend on the room number.

Also, we assume that the movement probabilities p, q, r and s are known by $\mathcal{S}$, where $\mathcal{H}$ has no knowledge about the strategy of $\mathcal{S}$.

Our last assumption is that $\mathcal{S}$ has limited information on the room where $\mathcal{H}$ is hidden. At every t, after his search, if $\mathcal{S}$ has searched the room where $\mathcal{H}$ was hidden, or in one of the adjacent rooms, he will be informed of the room where $\mathcal{H}$ was hidden, otherwise he will not get any information.

The strategies for $\mathcal{S}$ will be referenced as two vectors. One, as a quadruple $searcher = [previous, next, above, below]$ and the other as a quintuple $unknown = [previous, next, above, below, stay]$. These two vectors expresses the actions that $\mathcal{S}$ will take. The vector *searcher* express the move to perform: $(x+?)$ when $\mathcal{H}$ is in $\mathcal{B}(x-1, y)$ $(previous)$ or in $\mathcal{B}(x+1, y)$ $(next)$, and $(y+?)$ when $\mathcal{H}$ is in $\mathcal{B}(x, y-1)$ $(above)$ or in $\mathcal{B}(x, y+1)$ $(below)$. The vector *unknown* is a set of probabilities of going to the adjacent rooms $\mathcal{B}(x-1, y)$ $(previous)$, $\mathcal{B}(x+1, y)$ $(next)$, $\mathcal{B}(x, y-1)$ $(above)$, $\mathcal{B}(x, y+1)$ $(below)$, or $\mathcal{B}(x, y)$ $(stay)$ that is taken into account when the position of $\mathcal{H}$ is not known by $\mathcal{S}$.

The probability vector *unknown* is used because when the position of $\mathcal{H}$ is not known, $\mathcal{S}$ should be able to make a random walk. However if one of the parameters is set to 1 and the others to 0 it will not be a random walk anymore, $\mathcal{S}$ will always do the corresponding movement when the position of $\mathcal{H}$ is unknown.

The starting point for $\mathcal{S}$ will be chosen randomly because each of the $h_{x,y} = 1$ and because the board is infinite. Thus, there is no advantage in starting in any of the room as they are all subject to exactly the same

conditions. Therefore a random starting point for $\mathcal{S}$ is the best choice that he can make.

3.3 Approaches

Here again, two approaches have been exploited: a simulation based approach and a Markovian model approach.

The two approaches are the same as for the Circle model, therefore one can directly derive the application from section 2.1 and 2.2.

However, with this 2-Dimensional board, it has been realized that the number of entries for creating the matrix for the Markov chain was getting incredibly large. Indeed, as each combination of position for $\mathcal{S}$ and $\mathcal{H}$ was made to create the matrix P, with n the number of rooms, a matrix containing $n^2 \times n^2$ entries was created. As soon as the board starts to get over 100 rooms, which is a board with dimension for example 10×10 the matrix would have incredibly many entries (over 100 000 000 to be exact). Therefore an adaptation needed to be made. Instead of considering all positions of $\mathcal{S}$ and $\mathcal{H}$, a relative distance to $\mathcal{S}$ was made. As $\mathcal{B}$ is a never ending board (each border is connected to the other side) any position can be translated to another perspective. It has been decided to take the perspective of $\mathcal{S}$. This means that we placed $\mathcal{S}$ in the center of the board and computed for every possible relative position of $\mathcal{H}$ to $\mathcal{S}$, considering both players movements. For example if $\mathcal{H}$ moves one room right and $\mathcal{S}$ moves one room left, according to $\mathcal{S}$'s vision of the board, $\mathcal{H}$ will have moved two rooms right. This little change has reduced the number of possible combinations of positions for the players as, relatively only $\mathcal{H}$ is now moving. With this change the matrix complexity has dropped from $n^2 \times n^2$ to $n \times n$ entries.

3.4 Choice of an approach

Both approaches give similar results. However, they both have their strengths and weaknesses. The Markov approach gives an answer in a very short time. However, it uses n^2 entries to generate the matrix P.

The simulation approach is a fast process for a single game. However, a lot of games need to be played in order to have a reliable average. When we have a lot of rooms, it also needs lots of time, but much less memory for the computation than for the matrix computation. The choice of the method is just a trade off between time and memory.

Chapter 4

Results

4.1 Circle Approach

In this section are presented a number of results for the circle approach. They are included in this article as an example because each set up gives different results. The set up used for the different figures is $n = 10$, $p = \frac{1}{3}$ and $q = \frac{1}{2}$.

4.1.1 Average number of steps

The average number of steps needed by $\mathcal{S}$ to find $\mathcal{H}$ with the given set up and all the possible strategies for $\mathcal{S}$ can be computed with the help of the two methods given in Chapter 2. The comparison of the two approaches can be seen in Table 4.1. The simulation is based on $100,000$ runs.

As one can observe in the table, the best strategy with the given set up is the one where $\mathcal{S}$ applies the strategy $(-1, 0, +1)$. Which can be read as when $\mathcal{H}$ is in room $(i-1)$, go to that room, if $\mathcal{H}$ is in room $(i+1)$, wait in the current room, and if the position of $\mathcal{H}$ is unknown, go to room $(i+1)$.

Also one can see that one of the value is infinity, this relates to the warning that was given in Section 2.1.

Finally, one can see that there is a relation between the strategies: the ones that performs the worst are the ones that have strategies $(?, -1, 0)$ and the ones that performs the best are the one that have strategies $(?, 0, +1)$.

4.1.2 Limited time search

When we limit the time search, we calculate the probability to find $\mathcal{H}$ within n steps. Figure 4.1 shows the difference between the theoretical and the

simulated approach of one strategy for $\mathcal{S}$ on the same graph. In figures 4.2, 4.3, and 4.4, all strategies for $\mathcal{S}$ are tried based on the Markov chain approach. In all these graphs, the vertical axis expresses the probability to find $\mathcal{H}$, and the horizontal axis expresses the limited time frame. The output are based on $100,000$ simulations.

When analyzing these figures, one can notice a few things. In Figure 4.2, there is a strategy that always has a probability to catch $\mathcal{H}$ of 0.1. This is explained by the fact that it is the strategy on which the note has been made in section 2.1. However, as $\mathcal{S}$ and $\mathcal{H}$ are placed randomly in the rooms, there is a probability that they are placed in the same room as a starting point. (As the board consists of 10 rooms, there is a probability of $\frac{1}{10}$ that they are placed in the same room right at the beginning.

In Figure 4.3, one can see the curves that stands out and that performs the best are actually the ones that have been shown to performs the best in average when analyzing Table 4.1.

Finally, one can observe with the help of Figure 4.2 and the two other Figures 4.3 and 4.4, that applying a strategy $(?,?,0)$ (meaning that when the position of $\mathcal{H}$ is unknown $\mathcal{S}$ should wait) is the worst strategy to apply when trying to maximize the probability to find $\mathcal{H}$ in a limited amount of steps.

4.1.3 Reliability of the simulation approach

In order to see how reliable the simulation approach is, a statistical test has been performed. As in Table 4.1, the strategy that gives the biggest difference between the simulated value and the Markov value is (+1,0,0); it was used to perform the tests. The mean and the standard deviation of 5000 samples over respectively 10, 100, 1000, 10 000 and 100 000 simulations has been computed. Table 4.2 shows these results. The reduction of the standard deviation follows the general rule that *in order to reduce the standard deviation by a factor 2, you need 4 times as many runs* [9], it is indeed the case: there is a reduction of about $\sqrt{10} = 3.16$ times when we multiply the number of challenges by 10. As an additional piece of information the time to compute the n simulations has been added. As one can see, the time to compute the simulation is linear to the number of simulations.

Table 4.1: Comparison of the average number of steps to find $\mathcal{H}$ with the given strategy for $\mathcal{S}$ computed with the help of a simulation and a Markov chain procedure. $\mathcal{B}$ contain 10 rooms, and the probabilities for $\mathcal{H}$ are $p = \frac{1}{3}$ and $q = \frac{1}{2}$.

Simulation	Markov	Strategy
16.1199	16.0589	(0, 0, 0)
22.679	22.6767	(0, +1, 0)
298.2448	298.3242	(0, -1, 0)
21.9733	22.1619	(+1, 0, 0)
104.971	105.2689	(+1, +1, 0)
∞	∞	(+1, -1, 0)
13.7733	13.6348	(-1, 0, 0)
14.1371	14.1591	(-1, +1, 0)
617.8037	616.3125	(-1, -1, 0)
4.193	4.2	(0, 0, +1)
6.6639	6.6886	(0, +1, +1)
8.2944	8.2803	(0, -1, +1)
4.4714	4.475	(+1, 0, +1)
10.3037	10.302	(+1, +1, +1)
8.746	8.7999	(+1, -1, +1)
3.9717	3.9667	(-1, 0, +1)
6.136	6.1501	(-1, +1, +1)
8.3313	8.3335	(-1, -1, +1)
5.7676	5.771	(0, 0, -1)
5.6047	5.6005	(0, +1, -1)
6.4074	6.4163	(0, -1, -1)
10.5084	10.5062	(+1, 0, -1)
10.8143	10.8031	(+1, +1, -1)
11.4703	11.4979	(+1, -1, -1)
7.4119	7.4104	(-1, 0, -1)
7.3444	7.3347	(-1, +1, -1)
12.9747	13.0612	(-1, -1, -1)

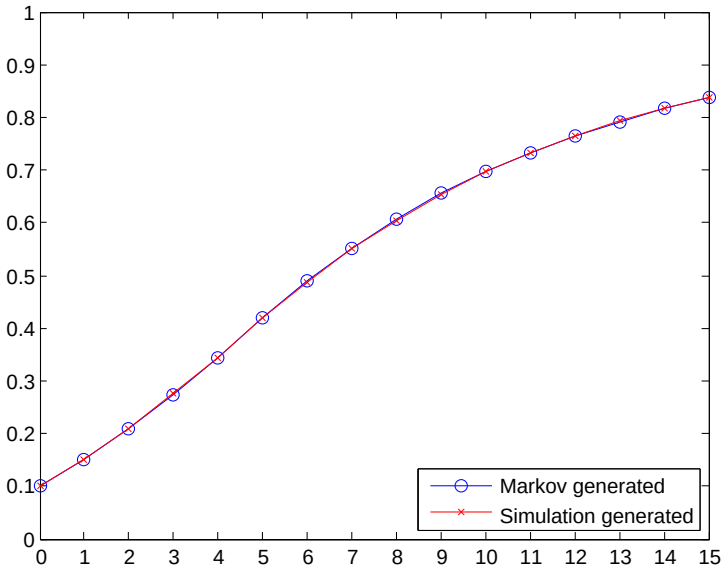

Figure 4.1: Comparison of a simulated limited time search and the theoretical limited time search. $\mathcal{S}$ strategy is (+1,-1,+1)

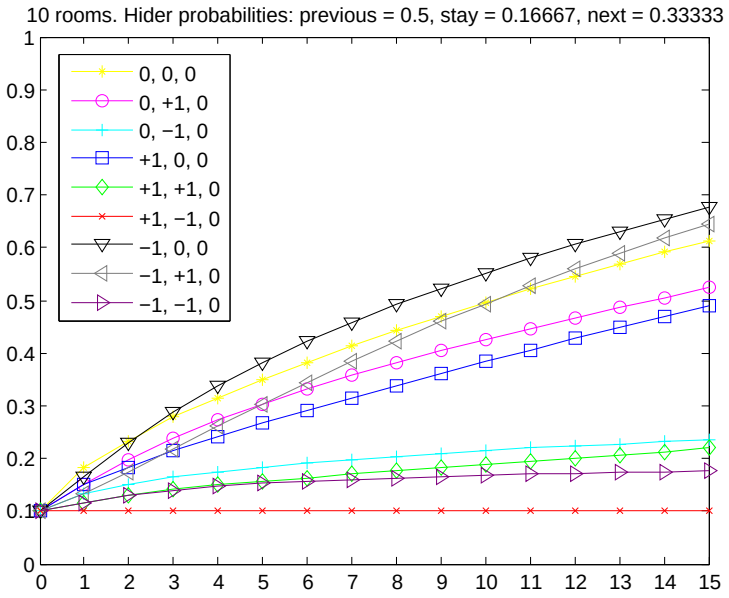

Figure 4.2: Limited time search computed with the help of the Markov chain model.

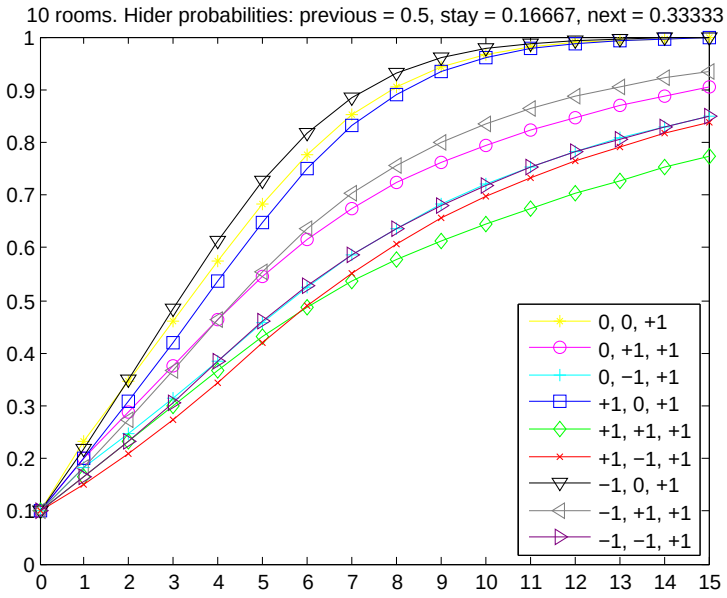

Figure 4.3: Limited time search computed with the help of the Markov chain model.

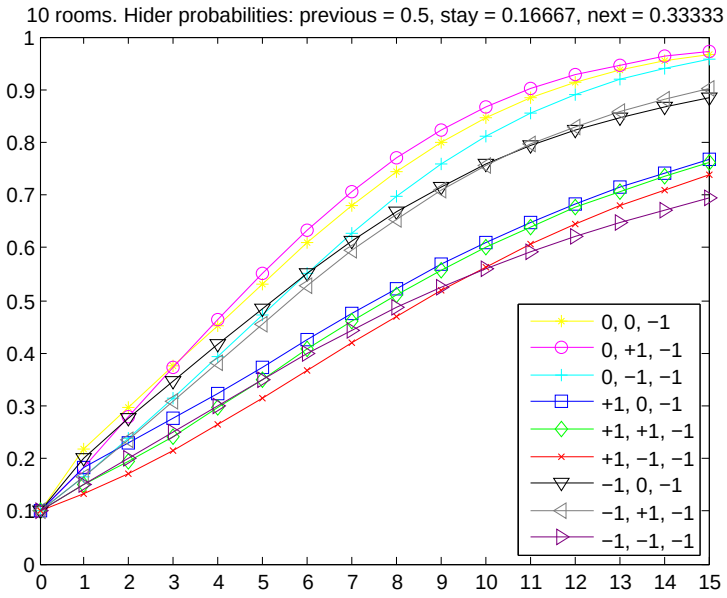

Figure 4.4: Limited time search computed with the help of the Markov chain model.

Table 4.2: Statistical test over the reliability of the number of experiments. The test is performed with 5000 samples of completed experiments, the number of rooms is 10, $p = \frac{1}{3}$, $q = \frac{1}{2}$, and the strategy for $\mathcal{S}$ is (+1, 0, 0)

Number of experiments	Mean	Standard deviation	Time to compute
10	22.0948	7.1513	0.005 s
100	22.0703	2.2515	0.057 s
1,000	22.1548	0.7146	0.374 s
10,000	22.1672	0.2271	4.980 s
100,000	22.1628	0.0714	46.173 s

4.2 2-dimensional Board Game

Here are presented a number of results for the 2-Dimensional game. The same remark as the one made in the previous section is made. All the results are included in this article as an example, because each set up gives different results. The set up used for the different figures will be a 4×4 board with, $p = 0$, $q = \frac{1}{2}$, $r = 0$, $s = \frac{1}{3}$. The vector *searcher* will be analyzed and the vector $unknown = [0.2, 0.2, 0.2, 0.2, 0.2]$.

4.2.1 Average number of steps

The average number of steps needed by $\mathcal{S}$ to find $\mathcal{H}$ with the given set up and all the possible strategies for $\mathcal{S}$ can be computed with the help of the two methods given in Chapter 3. The comparison of the two approaches can be seen in Tables 4.4 and 4.5. The simulation is based on $100,000$ runs. In the table, no values are infinity. This is explained because the random walk allows the two players to meet each other by chance, where in the set up of the circle game there was not such a randomized walk for $\mathcal{S}$.

As one can see in the tables, there is no values that are as big as the one that can be observed in the table of values of the circle game (Table 4.1). The reason for this is that when the position of $\mathcal{H}$ is unknown for $\mathcal{S}$, $\mathcal{S}$ uses its random walk procedure. This allows him to find $\mathcal{H}$ by pure chance. Also, unlike in the circle game, there is not a strategy that stands out of the results as a general best strategy, the best conclusion that could be made was that using a strategy $(1, ?, ?, ?)$ was not a suitable choice.

Still in the table one can see that the best strategy for $\mathcal{S}$ is to do $(0, 1, -1, 1)$ but that some other strategies are close to this one for example $(0, 1, 0, 1)$.

4.2.2 Limited time search

When we limit the time search, we calculate the probability to find $\mathcal{H}$ within n steps. Figure 4.5 shows the difference between the theoretical and the simulated approach of one strategy for $\mathcal{S}$ on the same graph. In figures 4.6, 4.7, and 4.8, some interesting strategies for $\mathcal{S}$ are tried. These figures have been chosen because some of the strategies perform better when n is low and get worse when n gets larger. Indeed, when analyzing the Figure 4.6, one can see that the strategy $(-1, 1, 1, 1)$ do not performs well when the number of steps are limited to 1 or 2, but is quite good when the number of steps is limited to 10.

One may also realize that the starting value for all these figures is the same for all the strategies. Indeed when the time is limited to 0, it means that they are taken into account only when both players are in the same room as a starting point . As the board contains 4×4 rooms then the probability that both players are in the same room at the beginning of the game is $\frac{1}{16}$ which is indeed the value that we observe in the figures.

In all these graphs, the vertical axis expresses the probability to find $\mathcal{H}$, and the horizontal axis expresses the limited time frame. The outputs are based on $100,000$ simulations.

4.2.3 Reliability of the simulation approach

In order to see how reliable the simulation approach is, a statistical test has been performed. The set up used will be the same as the one used for Figure 4.5: *strategy* $= (+1, -1, +1, -1)$. The mean and the standard deviation of 5000 samples over respectively 10, 100, 1000, 10 000 and 100 000 simulations has been computed. Table 4.3 show these results. The reduction of the standard deviation follows the general rule that *in order to reduce the standard deviation by a factor 2, you need 4 times as many runs* [9], it is indeed the case: there is a reduction of about $\sqrt{10} = 3.16$ times when we multiply the number of challenges by 10. In here also, we have included, as an additional piece of information, the time to compute the n simulations has been added. As one can see, the time to compute the simulation is linear to the number of simulations.

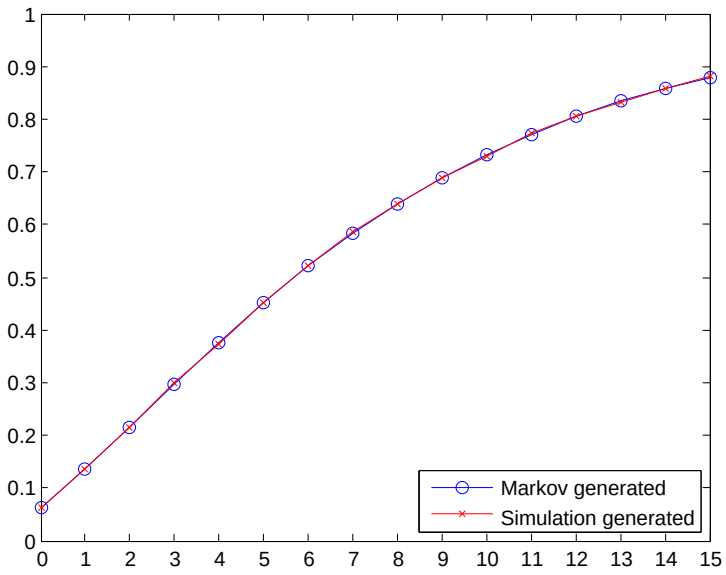

Figure 4.5: Comparison of a simulated limited time search and the theoretical limited time search. The parameters for $\mathcal{S}$ are $strategy = (+1, -1, +1, -1)$ and $unknown = [0.2, 0.2, 0.2, 0.2, 0.2]$

Table 4.3: Statistical test over the reliability of the number of experiments. The test is performed with 5000 samples of completed experiments.

Number of experiments	Mean	Standard deviation	Time to compute
10	7.7153	2.0523	0.008 s
100	7.7931	0.66208	0.065 s
1,000	7.7884	0.2071	0.827 s
10,000	7.7885	0.0656	7.426 s
100,000	7.7885	0.0208	78.391 s

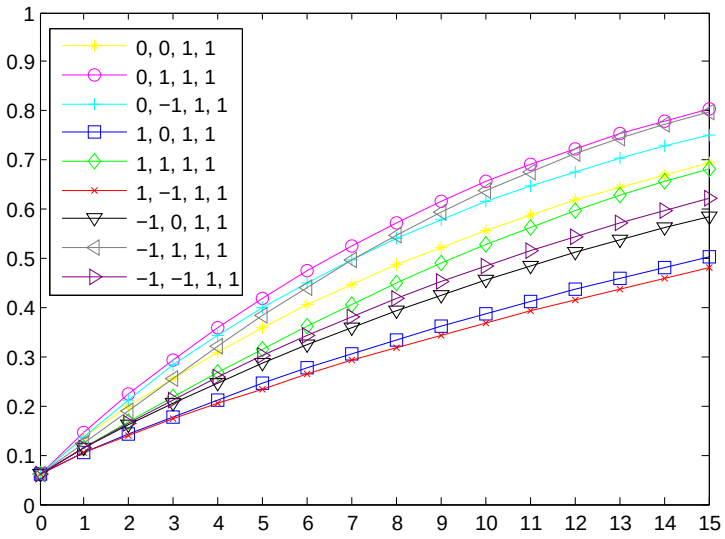

Figure 4.6: Limited time search computed with the help of the Markov chain model.

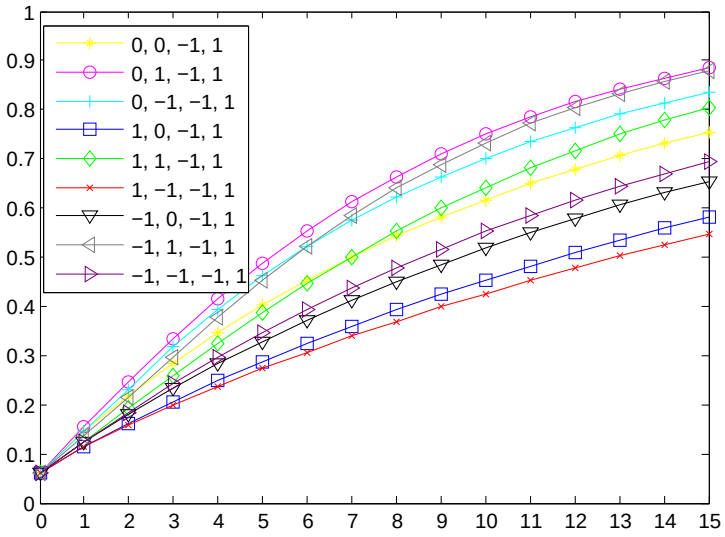

Figure 4.7: Limited time search computed with the help of the Markov chain model.

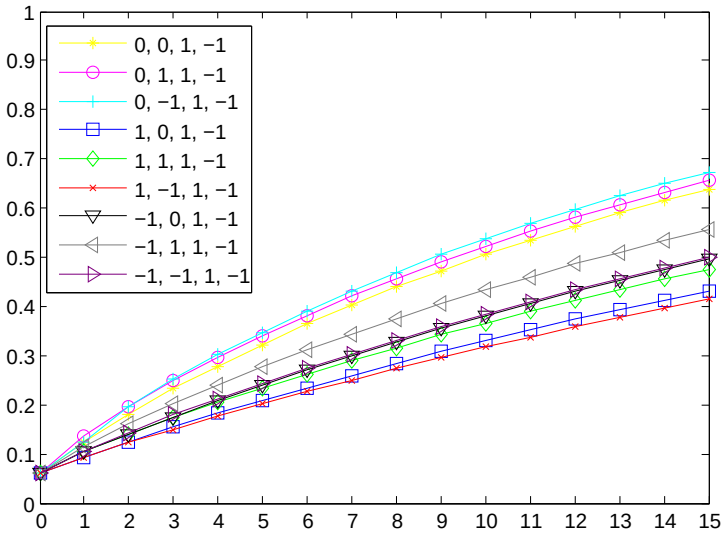

Figure 4.8: Limited time search computed with the help of the Markov chain model.

Table 4.4: Comparison of the average number of steps to find $\mathcal{H}$ with the given strategy for $\mathcal{S}$, computed with the help of a simulation and a Markov chain procedure.

Simulation	Markov	Strategy	Simulation	Markov	Strategy
10.6445	10.6196	(0, 0, 0, 0)	9.6675	9.6923	(0, 0, 0, 1)
10.1683	10.1805	(0, 1, 0, 0)	7.5853	7.5743	(0, 1, 0, 1)
9.8285	9.8144	(0, -1, 0, 0)	8.4006	8.3874	(0, -1, 0, 1)
17.179	17.1687	(1, 0, 0, 0)	15.3406	15.3395	(1, 0, 0, 1)
15.6128	15.518	(1, 1, 0, 0)	10.1528	10.1531	(1, 1, 0, 1)
17.8421	17.8413	(1, -1, 0, 0)	16.3391	16.3413	(1, -1, 0, 1)
14.6483	14.6695	(-1, 0, 0, 0)	12.7076	12.7387	(-1, 0, 0, 1)
12.7284	12.7627	(-1, 1, 0, 0)	7.9411	7.9467	(-1, 1, 0, 1)
14.6717	14.5632	(-1, -1, 0, 0)	11.9369	11.8735	(-1, -1, 0, 1)
14.966	14.9666	(0, 0, 1, 0)	13.0713	13.0514	(0, 0, 1, 1)
14.1064	14.1855	(0, 1, 1, 0)	9.472	9.4541	(0, 1, 1, 1)
13.6563	13.6448	(0, -1, 1, 0)	11.0491	11.0577	(0, -1, 1, 1)
27.5912	27.5286	(1, 0, 1, 0)	22.6473	22.6506	(1, 0, 1, 1)
23.8101	23.9136	(1, 1, 1, 0)	13.0763	13.0219	(1, 1, 1, 1)
29.1073	28.9974	(1, -1, 1, 0)	24.4394	24.478	(1, -1, 1, 1)
22.2174	22.2373	(-1, 0, 1, 0)	17.629	17.7063	(-1, 0, 1, 1)
18.5968	18.5084	(-1, 1, 1, 0)	9.6631	9.6899	(-1, 1, 1, 1)
21.8923	21.9438	(-1, -1, 1, 0)	15.907	15.8798	(-1, -1, 1, 1)
12.8138	12.8673	(0, 0, -1, 0)	10.8899	10.9386	(0, 0, -1, 1)
11.8899	11.8628	(0, 1, -1, 0)	7.4506	7.458	(0, 1, -1, 1)
11.1087	11.1301	(0, -1, -1, 0)	8.6399	8.6893	(0, -1, -1, 1)
22.1575	22.1972	(1, 0, -1, 0)	17.8829	17.9693	(1, 0, -1, 1)
18.1854	18.1368	(1, 1, -1, 0)	9.58	9.5698	(1, 1, -1, 1)
24.1452	24.0167	(1, -1, -1, 0)	19.8888	19.9643	(1, -1, -1, 1)
18.4664	18.4462	(-1, 0, -1, 0)	14.4663	14.572	(-1, 0, -1, 1)
14.8041	14.8422	(-1, 1, -1, 0)	7.8197	7.7885	(-1, 1, -1, 1)
18.3171	18.1865	(-1, -1, -1, 0)	13.0932	13.077	(-1, -1, -1, 1)

Table 4.5: Comparison of the average number of steps to find $\mathcal{H}$ with the given strategy for $\mathcal{S}$, computed with the help of a simulation and a Markov chain procedure.

Simulation	Markov	Strategy
9.9839	10.0029	(0, 0, 0, -1)
9.3171	9.2929	(0, 1, 0, -1)
9.2615	9.2164	(0, -1, 0, -1)
16.0278	15.9311	(1, 0, 0, -1)
13.9267	13.9075	(1, 1, 0, -1)
16.6087	16.511	(1, -1, 0, -1)
13.4448	13.3621	(-1, 0, 0, -1)
11.0101	11.0534	(-1, 1, 0, -1)
13.0131	12.9855	(-1, -1, 0, -1)
15.2384	15.2614	(0, 0, 1, -1)
14.5742	14.5454	(0, 1, 1, -1)
13.9654	13.8512	(0, -1, 1, -1)
28.6501	28.6085	(1, 0, 1, -1)
24.9997	24.933	(1, 1, 1, -1)
30.3144	30.3046	(1, -1, 1, -1)
23.1295	23.1805	(-1, 0, 1, -1)
19.4804	19.4065	(-1, 1, 1, -1)
23.0605	23.0748	(-1, -1, 1, -1)
12.4961	12.4912	(0, 0, -1, -1)
10.9856	10.9692	(0, 1, -1, -1)
10.5094	10.4927	(0, -1, -1, -1)
21.4708	21.5121	(1, 0, -1, -1)
16.1873	16.1543	(1, 1, -1, -1)
23.5547	23.6349	(1, -1, -1, -1)
17.7245	17.7607	(-1, 0, -1, -1)
13.1785	13.1627	(-1, 1, -1, -1)
17.25	17.2784	(-1, -1, -1, -1)

Chapter 5

Discussion

5.1 Beliefs update

In this article, the strategy for the searcher is determined at the beginning of the game. However after analysis, we have discovered that in some of the cases it is possible for $\mathcal{S}$ to know exactly the position of $\mathcal{H}$. Indeed if we keep the example discussed for the 2-dimensional board, the hider has only three moves, he either stays with probability $\frac{1}{6}$, goes to the adjacent room to the right with probability $\frac{1}{2}$ or goes to the adjacent room below with probability $\frac{1}{3}$. There will be a case where the searcher is, at time $t-1$, searching in the adjacent room to the right of $\mathcal{H}$. If the strategy for $\mathcal{S}$ in this particular case is to wait so that $\mathcal{H}$ might come to his room at time t, then no matter what happens $\mathcal{S}$ will know the position of $\mathcal{H}$. If $\mathcal{H}$ stays, $\mathcal{S}$ will still see him and we will be back to the same situation. If $\mathcal{H}$ moves to the room where $\mathcal{S}$ is, the game will be over, and if $\mathcal{H}$ goes below, he becomes out of sight for $\mathcal{S}$. However $\mathcal{S}$ will know with probability 1 that $\mathcal{H}$ is in the room below the one he was at $t-1$ as he knows the strategy of $\mathcal{H}$. With the set up analyzed in this article, when $\mathcal{S}$ does not see where $\mathcal{H}$ is, he makes a random movement. This means that in the situation explained here, even if one can derive the position of $\mathcal{H}$ and the corresponding best reply, we do not use the information to have a sustainable advantage to find $\mathcal{H}$. Some similar issues on beliefs have been analyzed in a previous article [8].

5.2 Past action and observed environment

Another weakness of this model, that can also be related to beliefs updating, is that $\mathcal{S}$ does not remember the past actions and observed environment. To

illustrate this statement, one can think about a hider that always moves with probability 1 in the same direction and $\mathcal{S}$ always missing him by one room. As both players have the same speed, it will be a never ending game. As both players move simultaneously, if $\mathcal{S}$ is in room $n-1$ and $\mathcal{H}$ is in room n, at the next time moment, $\mathcal{S}$ will move to room n and $\mathcal{H}$ will move to room $n+1$. The problem in this small example is that $\mathcal{S}$ has a limited view, thus if he does not chase $\mathcal{H}$, he will lose its trace and return in the state where the position of $\mathcal{H}$ is unknown. If $\mathcal{S}$ had the ability to remember that he had seen $\mathcal{H}$ the best reply would be to move in the opposite direction taking advantage of the infinite universe by catching him in the opposite direction. However as there is no memory of past actions and observed environment this is not possible in our model.

5.3 Hit probabilities

We made the assumption that each room has a hit probability $h_i = 1$. This leads to a big simplification of the model. Indeed, with this assumption there is no probability that a room that is searched still contains $\mathcal{H}$. If the k_i were not all equal to 1, we could have had a best starting point for $\mathcal{S}$ on the board, but also a different searching strategy depending on the distribution of these h_i. We did not looked in these situations but they have been discussed and analyzed in [8].

5.4 Random walk for the 2-dimensional model

As one might have noticed, a random walk procedure has been introduced in the 2-dimensional board. It is not present in the circle game because the board contains only one dimension and one can find a best reply to the walk made by $\mathcal{H}$. In the case of a 2-dimensional board, the procedure is not as easy as the one for the circle game, therefore the random walk was introduced to allow a more realistic behavior. One can imagine random walk that is be biased towards a particular direction or simply has the same probabilities for each of the possible directions. In this article we did not focused on the different possibilities of these random walks and which one would perform better than other depending on the movement procedure of $\mathcal{H}$. A simple test has been performed in section 6.1 to see if some walks were performing better than other but it is the only one that has been made.

5.5 Size of the board

This article did not discuss the impact of different size of board on the results. As one can imagine, it has an impact on the results in terms of average number of steps to find a prey, but not in any of the other parameters. This feature has been discussed and analyzed in the article by Scharf et al. [6].

Chapter 6

Analysis

6.1 Perpendicular movement

In this study, when $\mathcal{S}$ does not see $\mathcal{H}$, he uses a random walk strategy. However, we wanted to see if there was a better alternative than just a random walk. As we assume that the strategy of $\mathcal{H}$ is known, one can think of a better alternative. We performed a simple test: we decided to make $\mathcal{H}$ move in one direction only (always move in the room above) and compare with the different strategies that $\mathcal{S}$ could perform. As Figure 6.1 shows, the results are significantly different when we include a perpendicular movement behavior (meaning that $\mathcal{S}$ moves in a perpendicular direction from the one made by $\mathcal{H}$). It is important to note that we need to have a shift in a direction for both players, but it does not need to be a probability 1 that the players move in the desired direction. However, it still needs to have a general tendency to drift in the desired direction. This warning is given because if both players move in perpendicular direction with probability 1, as their speed are the same, they might always miss each other. This perpendicular movement is also taking advantage of the infinite universe.

6.2 Diagonal movement handling

In the original set up, diagonal movement was not allowed. The design of the model has been made this way because in a traditional grid board, diagonal movement makes some path faster than other especially when a player moves only horizontally and vertically. As a solution one can modify the board by shifting every second row by only a half tile to allow 6 direct neighbors instead of only 4. This gives a hex-grid board looking like Figure

6.2. In order to have an infinite universe with such a configuration, the number of rows must be even, otherwise the top and bottom of the board don't match.

A comparison between the hex-grid board and the regular 2D board has been performed to see if in similar conditions similar results could be observed. Indeed one can conclude that in similar conditions the results match. The test to see if the two boards match, has been performed on a 8×8 board with $\mathcal{H}$ having equal probability to go in any adjacent room and stay in the current, and $\mathcal{S}$ going in the room where $\mathcal{H}$ was seen and otherwise going in a random adjacent room. Another test to see how well a hex-grid board matches a traditional 2-D board, will be made in Chapter 9.

6.3 Vision

Instead of using a vision of only the neighboring rooms, we have extended the vision to see the impact that it would have on the process. This feature has been implemented in a way that implies that any room seen is reachable in the number of steps that extends the vision. If $\mathcal{S}$ sees 3 rooms ahead, then any room that is reachable in three steps is seen, not the other ones. In the case of a circle or a Hex-grid model, it is a straight forward application. However, in a normal 2D board, depending whether the diagonal movement is allowed or not, the visible region will be different. In the article by Scharf et al. [6] the diagonal movement is allowing a square vision while in our model diagonal movement is not allowed, creating a diamond shape visible area. This feature can be easily applied on a hex-grid because all direct neighboring rooms are seen and reachable.

6.4 Energy cost function

An application of the hide and seek game is a predator prey model. A predator needs to feed in order to survive. He can either go and search for food, or wait until a prey passes by. As one can imagine, waiting is less energy consuming than actively searching. Thus we decided to include an energy cost function to see the impact that it would have on the different strategies and on the optimal one. We allowed an initial amount of energy where not moving for one turn consumes some amount of energy and moving consumes more energy. The speed, or in our case the number of rooms that $\mathcal{S}$ can search in one time step, consumes some energy, but this energy is not divided in the number of sub-move: either the predator moves and consumes

Table 6.1: Amount k for which a moving predator is willing to pay in terms of energy per time step while still having an advantage on an ambush predator that consumes only one unit of energy. The amount has been tested on an arena containing 4 preys and of size 20×20

Vision	Affordable amount
1	$k < 1$
2	$k < 2$
3	$k < 4$
4	$k < 5$
5	$k < 7$
6	$k < 7$
7	$k < 7$
8	$k < 9$
9	$k < 9$

the amount of energy to move or he stays and consumes the amount of energy when not moving. We wanted to compare the probability of finding a prey by a predator being ambushed, and therefore waiting, and by a predator being active and searching for the preys. We decided to see if there was a threshold that could connect the number of prey and the cost of moving. We used an observable region that corresponds to the speed of the predator. If a predator can move n tiles away, then he will be able to see all the rooms reachable in n steps. By moving m steps, the predator will make his observed area bigger, even if he has only $n - m$ move left, he will still see n tiles away from this new position. It has been realized that the amount of prey is not the important factor (as one can see in Figure 6.3), but the amount of energy that the predator would consume for a given speed was the one that matters (as one can see in Figure 6.4). Table 6.1 has been made to get some insight about the result of the introduction of the cost function and understand a bit about the underlying dynamics. In this table the value k is the amount of energy that the moving predator can consume while still having an bigger probability to find a prey than an ambush predator. The table shows some thresholds that are based on a series of graph as the one that is display in Figure 6.4. As one can understand the accuracy and the confidence intervals haven't been tested, the table is just there to show that there is indeed different thresholds for the different speeds.

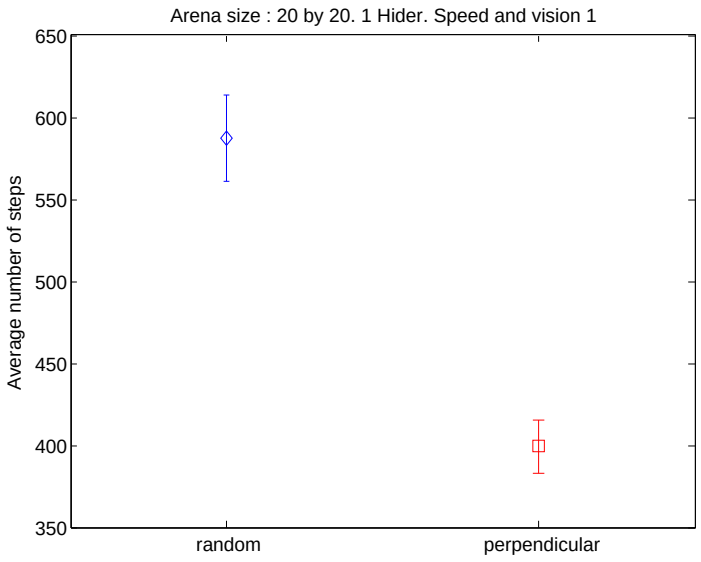

Figure 6.1: Comparison of a pure random strategy for the searcher to a perpendicular movement strategy.

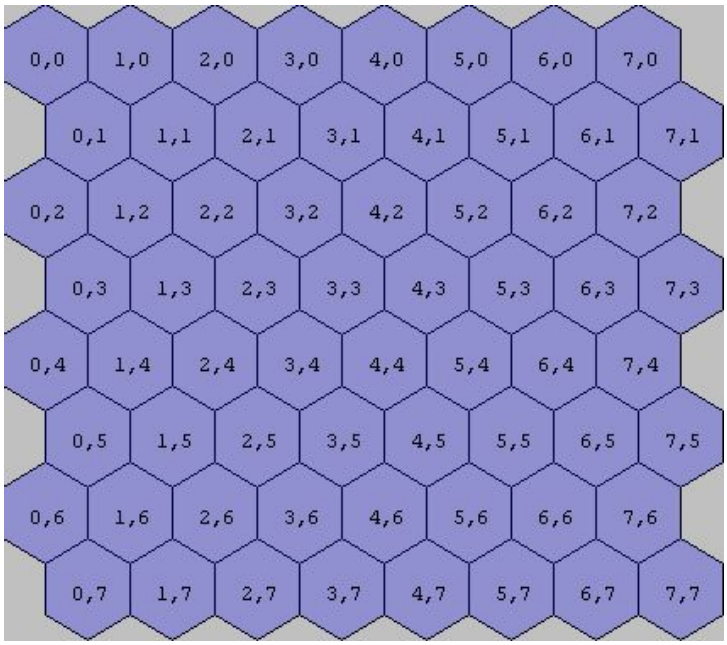

Figure 6.2: Hex-grid example

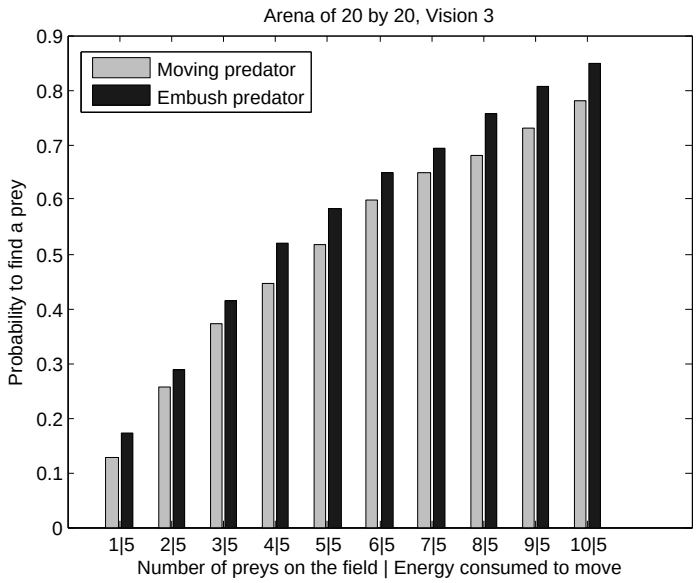

Figure 6.3: Probability to find a prey with an original amount of energy of 20 units, depending on the number of preys

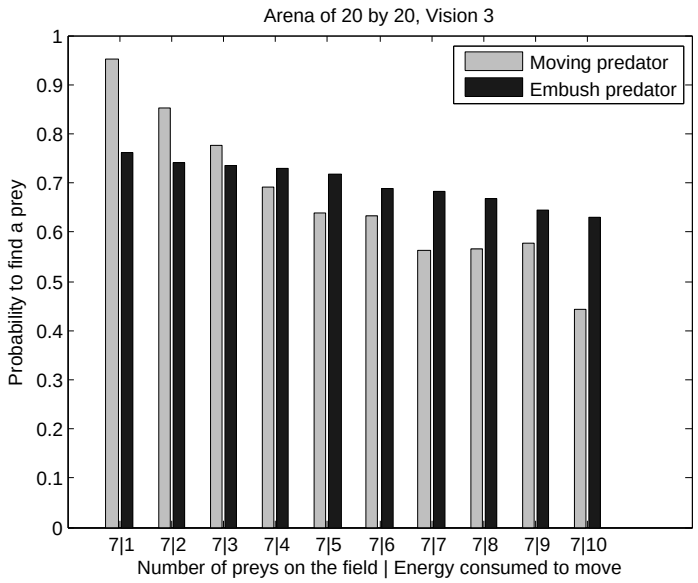

Figure 6.4: Probability to find a prey with an original amount of energy of 20 units, depending on the number of the amount paid for a movement.

Chapter 7

Conclusion

In conclusion, the theoretical model described here is a good alternative to the simulation model. The Markov chain model is a really good and reliable model but is really hard to modify due to the explosion of the complexity of the created matrix. The modification of the Markov model is complicated because any introduction of new parameters changes the way the matrix is built, and if there are multiple predators or preys, the complexity is growing. The complexity grows very fast with multiple preys and this makes it hard to use in an easy way. As an alternative, one can use the Simulation model. It is a model that is more flexible and that does not grow as fast in complexity, but that takes much more time to run and compute to have a reliable output. The different studies that have been made were not deeply analyzed because they were mainly out of the scope of this research but can lead to interesting results and applications in terms of real life simulation.

Part II

Comparing the model with related literature

Chapter 8

Verification

8.1 Introduction

As mentioned in the Preface, a similar study has been performed by Scharf et al. [6]. As a verification to our work, it has been decided to run tests under similar conditions and see if the models could match their results. As the diagonal movement is not handled in our situation, we have modified the moving condition of Scharf et al. to have a better match between the models. Also, we have allowed our model to handle multiple $\mathcal{H}$ (preys). It is to the attention of the reader that a review of the article has been made by Avgar et al. [1] which have been replied by Scharf et al. [7].

8.2 Speed Impact and Vision

In the article [6], the impact of the speed of the hider and the searcher has been studied. We have run experiments under similar conditions to see if our model could give similar results. Figures 8.1 and 8.2 show the results obtained with the help the simulation explained in this article. The results show similar behavior when the velocity changes. It also means that one could model the problem in terms of Markov chain.

8.3 Discussion

While running tests to see if both our model and the model by Scharf et al. could match, a few questions were raised. In this section we are going to discuss the interpretation they have made about the speed and the vision of

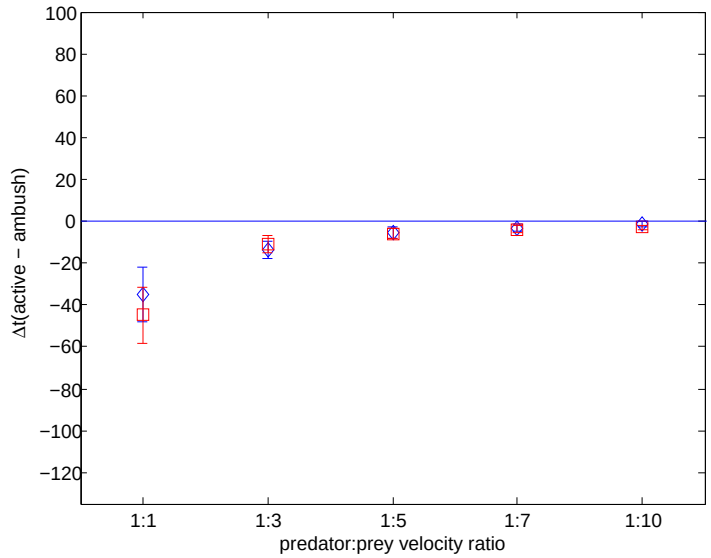

Figure 8.1: Impact of the Velocity on the time to capture a prey. Arena size: 20x20. Diamond : result by Scharf et al. Squares : results by our method.

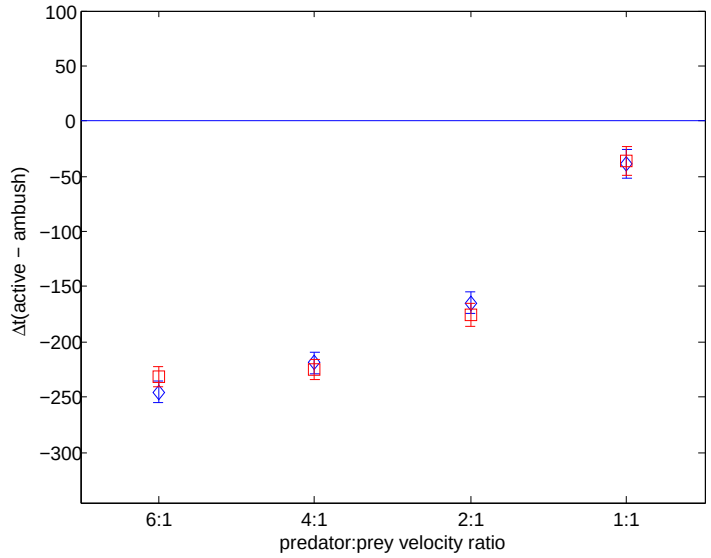

Figure 8.2: Impact of the Velocity on the time to capture a prey. Arena size: 20x20. Diamond : result by Scharf et al. Squares : results by our method.

the predator and how the choices influence the outcome while they will be analyzed in Chapter 9.

8.3.1 Speed

It has been realized after a few tests that an important factor to couple with speed was the vision. Indeed, if $\mathcal{S}$ has the ability to move two times faster than $\mathcal{H}$, it is important that he sees at least two rooms ahead in each direction otherwise he makes a random move in a direction while being blind for its second move thus he does not anticipate its next move. One can imagine that if $\mathcal{H}$ is at only two rooms distance, $\mathcal{S}$ could catch $\mathcal{H}$ in one turn if he can anticipate its second move. This feature is not of importance when talking about $\mathcal{H}$ because, in the assumption made by Scharf et al. and in our model, $\mathcal{H}$ does not try to avoid $\mathcal{S}$. Thus even if he could see far away, he would not use this advantage.

8.3.2 Vision

In terms of vision, the interpretation that Scharf et al. made was found very restrictive. Indeed, as soon as a prey is seen, it is assumed to be caught while being only capable of moving one tile at the time. This raised a big question, if a predator can see, and move instantly n tiles away, why not make these n steps to extend the observed area if no prey is in sight. Still in their interpretation there is no coupling between speed and vision. A predator can eat instantly a prey that is n tiles away but is not able to move there under normal conditions. This feature also makes the active searcher performing worse than if he was moving more than one tile away and able to observe the neighborhood. In our view, a plausible interpretation would be to make the observable area depending on the speed of the predator. This way if a predator sees a prey in the limit of its observable region, he can catch it, but at the same time, if he sees nothing, he can still use the energy he would have used to catch the prey to look around and maybe take advantage for the next moving sequence by spotting the prey.

Chapter 9

Experiments

9.1 Vision and speed

A few experiments have been made to see the impact of the interpretation of the vision. The figures 9.1 and 9.3 show the combination of the speed and vision. This means that when a predator is moving 6 times faster than the prey, he can move and see 6 tiles away. In these graphs, seeing is not catching. In order to give more insight about the meaning of the graph, two additional graphs have been added, Figure 9.2 and 9.4. They show the average number of steps needed by an ambush predator to capture the prey.

As one can see in the results that an ambush predator that can see 6 tiles away and can go in one move all the way to the prey performs much better than a predator that can move 6 tiles but that cannot see one tile away. This can be derived with the help of Figure 9.3 and 9.4: the first one shows that the difference in time, when the predator is ambush to when the predator is moving, is much lower and the second one shows that the average is lower when the vision is extended.

The results are of course not unexpected, therefore, a comparison between the *see is catch* and the combination of *move and see* has been performed. In the first case, the predator sees 6 tiles away and catches the prey instantaneously if he sees any, but moves only one tile per turn. In the second case, the predator can move and see 6 tiles away, but he still needs to make all the moves to get to the prey, so the catch is not instantaneous. Figure 9.5 shows that there is indeed a big difference in these two approaches.

In Figure 9.6 and 9.7, we have tried to see if there was a possibility to compare the model by Scharf et al. allowing diagonal movement and the

Hex-grid model. One can see that these models are comparable. It means that it could be a good alternative to remove different path lengths in a traditional grid board.

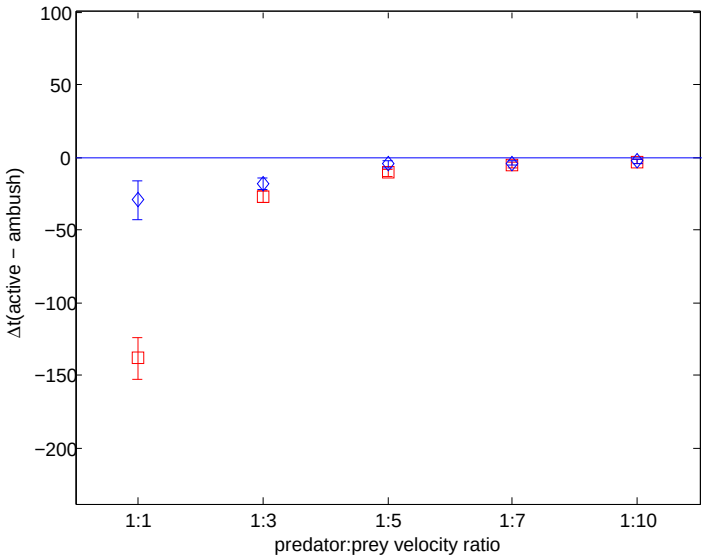

Figure 9.1: Impact of the vision on the time to capture a prey. Arena size: 20x20. Diamond: Vision is always 0. Square: Speed and Vision combined.

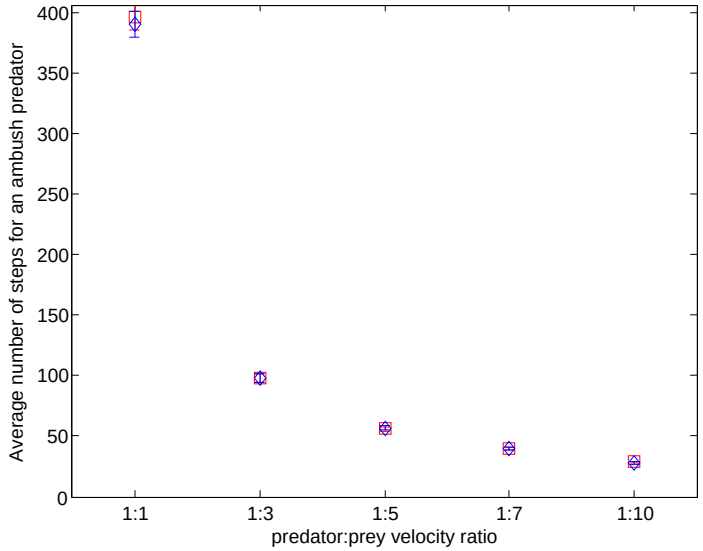

Figure 9.2: Impact of the vision on the time to capture a prey. Arena size: 20x20. Diamond: Vision is always 0. Square: Speed and Vision combined.

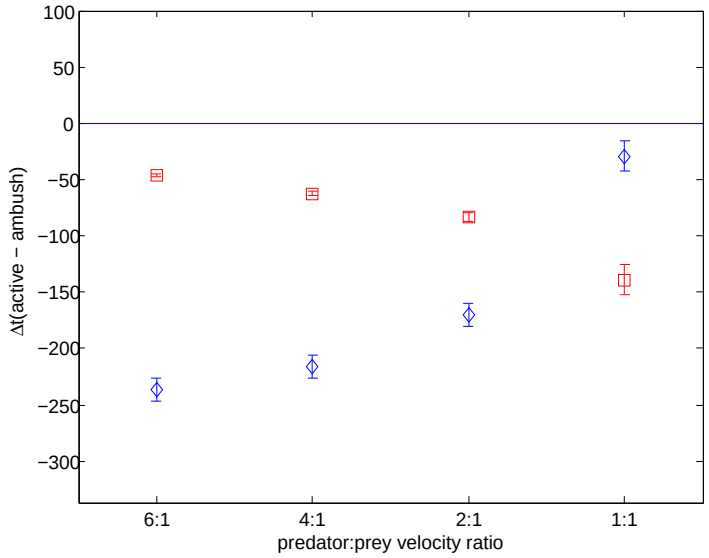

Figure 9.3: Impact of the vision on the time to capture a prey. Arena size: 20x20. Diamond: Vision is always 0. Square: Speed and Vision combined.

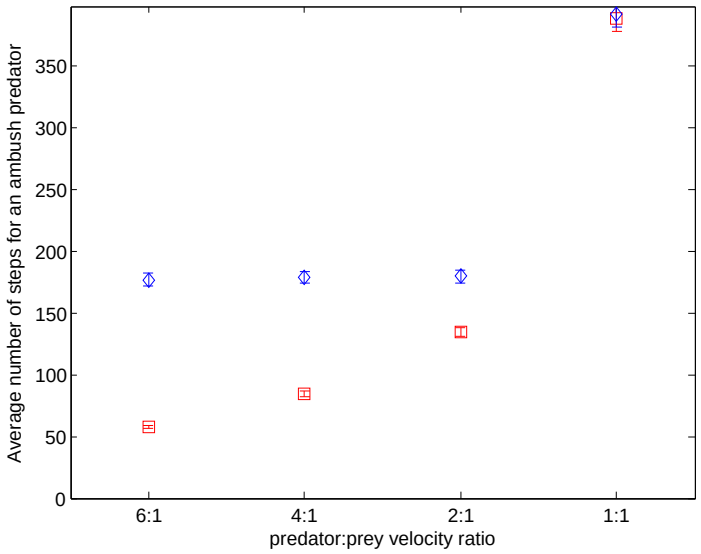

Figure 9.4: Impact of the vision on the time to capture a prey. Arena size: 20x20. Diamond: Vision is always 0. Square: Speed and Vision combined.

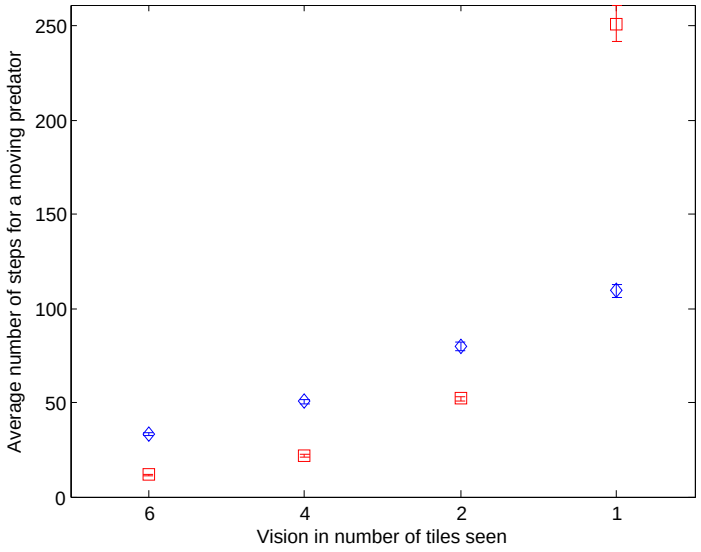

Figure 9.5: Impact of the vision on the time to capture a prey. Arena size: 20x20. Diamond: The speed is always 1, but see is catch. Square: Speed and Vision are combined but see is not catch.

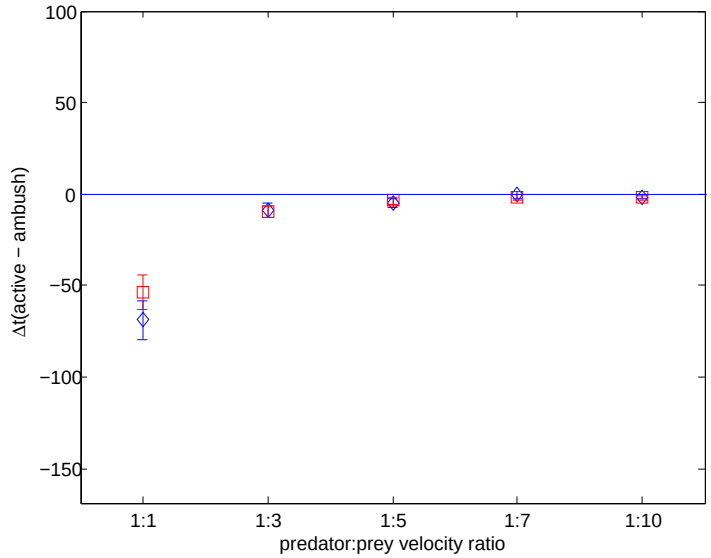

Figure 9.6: Comparison of the Scharf implementation (squares) and the Hex-Grid implementation (diamonds)

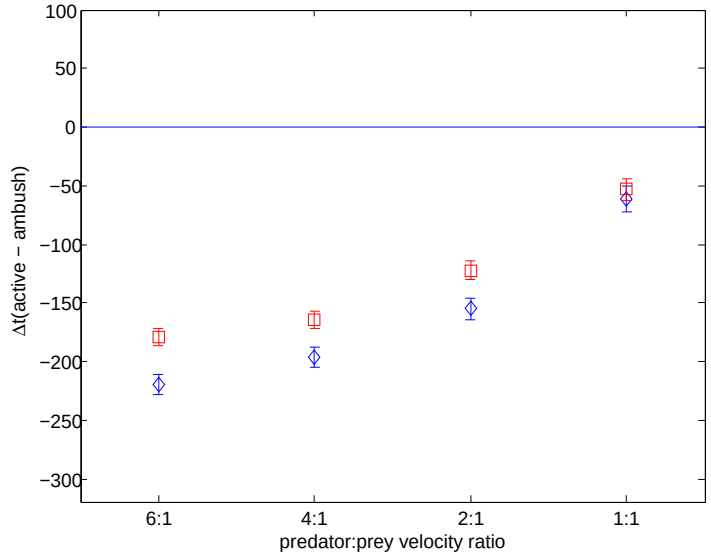

Figure 9.7: Comparison of the Scharf implementation (squares) and the Hex-Grid implementation (diamonds)

Chapter 10

Conclusion

In conclusion the choices of the interpretation of the speed, the vision, and any other parameter have to be made with great care. Indeed, they will influence the entire simulation, the results and, as one may understand, the reliability of this simulation. During our research it has been realized that even the order of update between the movements makes a difference and gives different results. It is important to know that a choice of an implementation should be made to simulate a behavior in a particular environment and that it might not be applicable for a slightly different one. One should ask the help of a biologist or another relevant expert in the desired field to simulate the appropriate behavior and get a more reliable simulation.

Bibliography

[1] T. Avgar, N. Hovitz, L. Broitman, and R. Nathan. Notes and comments. how movement properties affect prey encounter rates of ambush versus active predators : A comment on Scharf et al. *The American Naturalist*, 172(4), 2008.

[2] V. Chvatal. *Linear Programming.* W.H. Freeman and Company, 1983.

[3] J. Derks. A note on hide-and-seek games: some first investigations and model building, 2008.

[4] I. Gikhman and A. Skorohod. *The theory of stochastic processes 1.* Springer, 2004.

[5] S. Ross. *Introduction to Probability Models.* Academic Press, 9-th edition, 2007.

[6] I. Scharf, E. Nulman, O. Ovidia, and A. Bouskila. Efficiency evaluation of two competing foarging modes under different conditions. *The American Naturalist*, 168(3), 2006.

[7] I. Scharf, O. Ovidia, and A. Bouskila. Prey encounter rate by predators: Discussing the realism of grid-based models and how to model the predator's foarging mode: A reply to Avgar et al. *The American Naturalist*, 172(4), 2008.

[8] P. Uyttendaele. Hide and seek games. *Bachelor Thesis Universiteit Maastricht*, 2008.

[9] R. Walpole, R. Meyers, S. Meyers, and K. Ye. *Probability & Statistics for Engeneers & Scientists.* Pearson Education International, 8-th edition, 2007.

[10] P. Zollner and S. Lima. Search strategies for landscape-level interpatch movements. *Ecology*, 1999.