Hide and seek games

Philippe Uyttendaele

June 16, 2008

Abstract

In this article is presented a classical Hide and Seek game. According to his initial beliefs, the purpose for the searcher is to find the hider as fast as possible, while the hider wants to be hidden as long as possible. Multiple search methods are presented and evaluated on a simulation based model. The model presented is based on Matrix Games.

1 Introduction

In this article we are interested in finding an optimal starting point for the searcher in a classical Hide and Seek Game. One person, denoted by $\mathcal{H}$, has a number of hiding places, here referred to as rooms, at his disposal. At the start of the game he chooses a room to hide in, knowing that he will not be able to change room during the game. A second person denoted by $\mathcal{S}$ starts the search for $\mathcal{H}$. $\mathcal{S}$ may search in the order that he wants but he may only search one room at each step $i = 1, 2, ...$ In each room the visibility is limited. This is represented by the fact that $\mathcal{S}$ finds $\mathcal{H}$ with hit probability t_r in room r. These hit probabilities don't change during the process and both players are assumed to know these probabilities [3]. The goal is to find a stable situation where $\mathcal{S}$ minimizes the expected number of search to find $\mathcal{H}$ while $\mathcal{H}$ wants to maximize that number. This is not trivial. Indeed, as both players know each other's best strategy, it makes the process difficult. Assume that $\mathcal{S}$ knows $\mathcal{H}$'s strategy. $\mathcal{S}$ adapts himself to find $\mathcal{H}$ as fast as possible. Knowing that $\mathcal{S}$ knows his strategy, $\mathcal{H}$ can change it so that the strategy of $\mathcal{S}$ is no longer suitable. But again, knowing that $\mathcal{H}$ knows that $\mathcal{S}$ knows $\mathcal{H}$'s best strategy, $\mathcal{S}$ can adapt his strategy. Here again if $\mathcal{H}$ knows, that $\mathcal{S}$ knows, that $\mathcal{H}$ knows, that $\mathcal{S}$ knows $\mathcal{H}$'s best strategy, $\mathcal{H}$ can adapt. This leads to an endless chicken and egg paradox, where no solution can be found! This mean that we are interested in a strategy for both player where the knowledge of the other's strategy will not change the player's optimal strategy.

2 Model

The model described here is the classical Hide and Seek game as described in the Introduction. The model is based on a Matrix Game with a few more assumptions. It is assumed that each search made by $\mathcal{S}$ in a room takes one time unit. These one time unit searches will be called Stages and the process of the search for $\mathcal{H}$ until he is found will be called a Challenge. In this article we are interested in the starting strategy that will minimize the number of stages needed by $\mathcal{S}$ to find $\mathcal{H}$.

2.1 Implementation of the model

The problem can be handled in two main different manners, a simulation based approach and a mathematical approach. Both methods have advantages and inconveniences. The mathematical approach can be made such that it is deterministic. It is a strength because it means that nothing variate and no errors due to randomness are present. However, it also means that a random choice is never made and decisions are made according to strict rules. A good example is that if $\mathcal{S}$ has the choice between two rooms that have the same probability to contain $\mathcal{H}$, always the same room will be chosen first. In contrast, the simulation based approach is a simulation of each challenge of a hide and seek game. This is a non-deterministic approach, because whether the object is found or not can be handled by a random number generator; therefore two consecutive runs will not especially give the same outcome. Also the choice for a room, when two rooms have the same probability to contain $\mathcal{H}$, can be made randomly. All kinds of randomized actions will be taken which can vary the choices of a room or vary the strategies. These aspects have also a backside, as it is a non-deterministic process, an average over a large number of challenges needs to be made in order to have a reliable outcome. Another important matter is the time consumption of both methods. As the mathematical model can create a challenge almost instantly, and only one is needed, the simulation based model is a lot more time consuming. As, in the simulation based model, each stage is simulated and a number of challenges is needed to have a reliable average number of stages, the time needed is linear with the number of challenges needed. The choice for the implementation of the model was a simulation based model. The choice for that particular implementation was made because it offers more flexibility in the choice of the decisions; they can be made deterministic as well as probabilistic. Another motivation for that choice was that it offers a better understanding of the process at each step. In this model, a number of challenges is made and the average over these challenges is taken in order to have a good approximation of the number of steps needed to find $\mathcal{H}$. However, a comparison with the deterministic approach has been made in order to see if the theory matches the simulation. The result of this comparison is that the two models match and give the same outcome. Not exactly, because one is deterministic and the order in which the rooms are presented matters, where it is not the case with the other model. Also, the outcomes are similar but not exactly the same due to the random factors.

2.2 Matrix Games

The process of finding a best strategy for $\mathcal{S}$ and $\mathcal{H}$ is based on matrix games. A matrix game is a matrix $A = (a_{ij})$ that defines a game for two players. The *row player* selects one of the rows $i = 1, 2, ..., m$ and the *column player* selects one of the columns $j = 1, 2, ..., n$ independently. The resulting payoff to the row player by the column player is a_{ij}. The row player may decide to mix his strategy so that each row i will be selected with a probability x_i where $\sum_i x_i = 1$. The column player may use such mixed action as well [1].

In our case, the row i corresponds to the room where $\mathcal{H}$ is hidden and the column j to the room where the $\mathcal{S}$ starts his search. The entry a_{ij} corresponds to the average number of steps needed by $\mathcal{S}$ to find $\mathcal{H}$, when $\mathcal{S}$ start the search in room j and $\mathcal{H}$ is hidden in i.

2.3 Beliefs

For the search method of $\mathcal{S}$, we need to make use of so called beliefs or subjective probabilities. We assume that $\mathcal{S}$ considers a weight $h_r > 0$ for each room r, expressing his belief that $\mathcal{H}$ is hidden in room r. The higher h_r, the higher $\mathcal{S}$ beliefs that $\mathcal{H}$ is hidden in room r. Given the beliefs h_1,...,h_n, $\mathcal{S}$ should choose the room that gives him the *best chance* to find $\mathcal{H}$. However, *best chance* can be represented in several way. This is why we introduce the use of procedures to determine these rooms. The procedures for the selection of the rooms will be called the Search Methods. The different Search Methods will be discussed in section 2.5.

2.4 Bayesian updating

After looking in a room, $\mathcal{S}$ has to update his beliefs according to the well known Bayesian rules: his new belief h'_r is computed as follows :

$$h'_r = \Pr\left[\mathcal{H} \text{ in room } r \middle| \text{ not found in } r\right]$$

$$= \frac{\Pr\left[\mathcal{H} \text{ in room } r \text{ and not found in } r\right]}{\Pr\left[\mathcal{H} \text{ not found in } r\right]}$$

With $\Pr[\mathcal{H} \text{ not found in } r] = 1 - \Pr[\mathcal{H} \text{ found in } r] = 1 - h_r t_r$, we have

$$h'_r = \frac{h_r(1 - t_r)}{1 - h_r t_r}$$

Also the beliefs for the other rooms $i \neq r$ are updated as follows:

$$h'_i = \Pr\left[\mathcal{H} \text{ in room } i \middle| \mathcal{H} \text{ not found in } r\right] = \frac{h_i}{1 - h_r t_r}$$

Now $\mathcal{S}$ searches in a room where his belief is the highest according to the search method chosen. From this procedure one should note that the selection of the room is totally determined by the search method and this Bayesian updating. Two challenges that have the same initial beliefs have in average the same outcome.

2.5 Search Methods

Several search methods for the selection of the next room to search have been explored. This was done to see if assumptions about the behavior of the process would fit the outcome of the simulations. The search methods explored for $\mathcal{S}$ are the following:

- The first one is when only the room with the highest belief is searched. This will be called No Forecasting or F-0.

- The second one is a myopic one where the belief and the chance of finding the object are taken into account with the formula $h_i \times t_i$. This is a one step ahead forecasting procedure where $\mathcal{S}$ maximizes his one step probability of finding $\mathcal{H}$. It will be referenced as F-1.

- The third one is a two step forecasting method. The room selected is the one where the value $h_i t_i + h'_j t_j$ is maximal (i and j can be the same). Where h'_j is the new belief for room j after unseccessfully searching in room i with belief h_i. It will be referenced as F-2.

For each of these procedures, if the searcher is indifferent between two rooms, a random choice between these rooms is made for the selection of the next one to search. The room to search next is calculated, but after every step, the calculation is re-evaluated, this means that the two step forecasting procedure is not a planning for two following steps but only a discrimination for the one time next step.

It turned out that F-2 was not giving a better performance than F-1. It was actually performing worse. Investigation about this phenomenon was therefore needed. A probabilistic decision was tried, where, at each step, not only the room with the highest belief is searched but every room is searched with some positive probability corresponding to the range of beliefs for the rooms. This means that all the rooms have some positive probabilities to be searched at every step. The following example illustrates this probabilistic method. If three rooms have to be searched and the beliefs for the rooms are [0.12 0.46 0.42] then the probability to search in each of the room will be [0.12 0.46 0.42] instead of [0 1 0] when only the best room gets searched.

3 Matrix creation

In order to see the best strategy for $\mathcal{S}$, the game has been presented as a matrix game. Each entry a_{ij} corresponds to room i where $\mathcal{H}$ is hidden and room j where $\mathcal{S}$ starts searching. For this matrix an optimal strategy for $\mathcal{H}$ is calculated that maximize the minimum number of steps needed by $\mathcal{S}$ to find him. With this knowledge, $\mathcal{S}$ adopt the initial beliefs that would lead to the least number of search to find $\mathcal{H}$.

3.1 One entry in the matrix

In order to be able to calculate the entries of the matrix, multiple challenges of a hide and seek game were made. One challenge is a search procedure that lasts until $\mathcal{H}$ is found. The average over these challenges is made and then taken as one entry in the matrix. Such an average is taken so that the fluctuation due to randomness in the room selection is reduced.

3.2 Best strategy selection

The beliefs play a big role in the selection of the room during the process. As only the initial setup of the beliefs will determine the outcome of the challenge (see section 2.4), it is of great importance to find the best strategy to adopt by $\mathcal{S}$ to find $\mathcal{H}$ as fast as possible. In order to find the optimal strategy for $\mathcal{S}$, the value of the game is computed after each game and the corresponding optimal mixed strategy for $\mathcal{S}$ is adopted. These mixed strategies consist in the optimal play for both players. The calculation of the best strategy is based on the Generalized Simplex Method [2]. The strategy adopted by $\mathcal{S}$ will be the best so that no matter what action $\mathcal{H}$ takes, he can do at most the value of the game. A memory factor has been introduced to avoid a pure strategy jumping between rooms and to obtain convergence. As observed in the selection and the update of the beliefs, the best strategy is in accordance to the current beliefs. If only the new strategy is taken into account for the update, then the beliefs are taken in a way that only one room is searched very often compared to the others. Here is an illustration of this phenomenon:

$$\begin{pmatrix} 3.3592 & 3.8156 \\ 4.5849 & 4.1837 \end{pmatrix}$$

In this matrix, the value of the game corresponds to the entry $a_{2,2}$, therefore the best strategy for $\mathcal{H}$ is to hide in room 2. As $\mathcal{S}$ adapts his beliefs to $\mathcal{H}$'s best strategy, he will have beliefs [0.01 0.99]. Then the new matrix will be computed:

$$\begin{pmatrix} 8.3677 & 13.1484 \\ 3.5137 & 2.5105 \end{pmatrix}$$

At this second stage, the value of the game corresponds to the entry $a_{1,1}$, and the best strategy for $\mathcal{H}$ to hide in room 1. As $\mathcal{S}$ adapts his beliefs to $\mathcal{H}$'s best strategy, he will have beliefs [0.99 0.01]. Then the new matrix will be computed:

$$\begin{pmatrix} 2.0073 & 2.9843 \\ 10.7413 & 7.9072 \end{pmatrix}$$

As one can understand, here the best strategy for $\mathcal{H}$ is again to play room 2. No matter how many games we will play, we will see that the strategy for $\mathcal{S}$ is jumping from one room to the other after each iteration.

A jumping pure strategy result can be seen in Figure 1 and Figure 2. Figure 1 shows the values of the game of such a strategy, and Figure 2 shows the value of the beliefs for the selection of the rooms. It can be observed in Figure 1 that the value of the game is jumping up and down. This can be linked to Figure 2 directly where the beliefs are alternating from two rooms to one room giving a value of the game that is also alternating. In Figure 1, the first 5 values are increasing. This is a natural phenomenon, because each matrix has an impact on the next and on the corresponding optimal strategy. As the starting beliefs are taken arbitrarily, the first few steps have a different strategy than the remaining ones but converge to that jumping strategy phenomenon. This is observed after the startup period in Figure 1.

The memory factor is a factor that increases the weight of the previous beliefs. This allows the strategy for $\mathcal{S}$ to stabilize and converge to a situation where the number of stages is minimal because the previously taken beliefs still have an impact on the new ones. The old strategy is not forgotten and old strategies are kept in the new beliefs. The formula for the application of this factor is:

$$b' = \alpha \times b + (1 - \alpha) \times c$$

Where b' is the new belief value, b the old belief value, c the optimal strategy for $\mathcal{S}$, and α the memory factor.

In the implementation of the model, the memory factor used is $\alpha = 1 - \frac{1}{n}$ with $n = 2, 3, ...$ Where n is an increasing counter that increases with challenges that lead to an increase in the value of the game. This memory factor has been chosen because it is a factor that does not increase too rapidly.

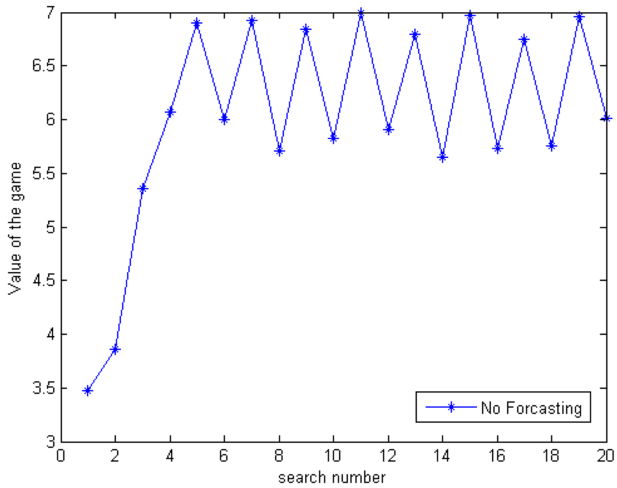

Figure 1: Plot of the value of the game when no memory factor in introduced

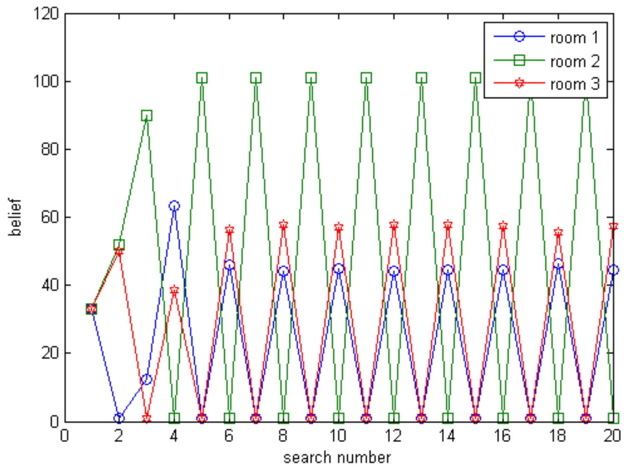

Figure 2: Selection of the room when no memory factor is introduced

4 Statistical Test

A statistical test has been made because the graph of the value of the game was not a stable descending line, but was oscillating instead. A check whether the value of the game could be wrongly calculated or exposed to a layer variation needed to be made. This could be due to not enough challenges played before averaging to get an entry (see Section 3.1 for more information). If the standard deviation is too big, one action can become better than another, where it is just an artifact from the simulation. This will lead to a reinforcement of a belief, where it is just a noise factor. The mean and the standard deviation of the entry according to the number of challenges were calculated to see how many were needed to have a good approximation of an entry.

4.1 One entry

The first statistical tests were based on a single entry to see what would be the impact of many rooms. The population size for the test is 5000. An F-1 procedure is taken for the beliefs. The base point was to see the behavior of the system with multiple rooms when the theoretical chance of finding $\mathcal{H}$ in these rooms would be equal to 0.1, regardless of the number of rooms.
The rooms have therefore been chosen with the according hit probabilities (t_i):

- 2 rooms with respectively probability 0.3162 and 0.3162 to find the object.
- 3 rooms with respectively probability 0.5623, 0.5623 and 0.3162 to find the object.
- 4 rooms with respectively probability 0.7499, 0.7499, 0.5623 and 0.3162 to find the object.

Table 1 represents the reliability of an entry a_{ij} of the matrix according to the number of challenges performed in order to get the entry. As explained before the beliefs are based on the mixed strategy that will lead to the lowest value of the game. According to Cantelli's inequality *No more than* $\frac{1}{1+k^2}$ *of the values are more than* k *standard deviations away from the mean on one side* [4]. This means that if, between two entries, the standard deviation is bigger than half of the difference of the mean, there is a 20% chance that these two entries will vary in such a way that one is bigger than the other whereas the average shows that it should be the opposite. This is a bad effect because the corresponding mixed strategy will lead to a selection of beliefs that is based on a challenge that is not reflecting the problem. A reinforcement of beliefs from "wrong information" will be observed and, therefore, a worse performance in the next run. With Cantelli's inequality, we can assume that the chance that two entries will be inverted should not exceed 5%. It is only if the difference of the means does not exceed 5 times the standard deviation, that percentage can be achieved. 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* [5]. It can be observed that in Table 1, 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.

4.2 All entries

A comparison inside the final matrix has been made to see if there was a big fluctuation between the different entries in the matrix itself. The comparison is based on 3 rooms with hit probabilities respectively of 0.5623, 0.5623, and 0.3162. The number of challenges is 100 to generate one entry in the matrix. The beliefs are the same for each of the rooms and finally the population

# of Rooms	# of Challenges	Mean	Std. dev.
2 Rooms	10	5.7018	1.7496
	100	5.6625	0.52912
	1000	5.6647	0.16891
	10000	5.6705	0.05445
3 Rooms	10	6.9393	1.7933
	100	6.9153	0.56109
	1000	6.9212	0.17828
	10000	6.9205	0.05695
4 Rooms	10	7.6337	1.9793
	100	7.5967	0.61539
	1000	7.6051	0.19893
	10000	7.6006	0.06210

Table 1: Calculation of the mean and standard deviation of the value of one entry on a different number of challenges

Mean	3.9738	4.5406	4.8259
	4.5405	3.9770	4.8172
	7.5671	7.5652	6.9139
Standard deviation	0.4865	0.4376	0.4293
	0.4428	0.4762	0.4280
	0.5063	0.4975	0.5696

Table 2: Comparison of the mean and the standard deviation of all the entries in the matrix

of the statistical test is 5000 for each entry. These challenges are based on the F-1 strategy. Table 2 shows the results of this procedure. As one can see the different entries in the matrix of the standard deviation show that the standard deviations do not vary so much between entries; they are in the same range of values. Also, when $\mathcal{S}$ searches first in the room where $\mathcal{H}$ is hidden, it is observed that the standard deviation is bigger. This can be explained because it is possible that $\mathcal{H}$ is found directly in many of the challenges and this leads to an increase of the standard deviation. The importance of having a small standard deviation between the entries was discussed before. Here it can be seen that 100 runs is too few to be suitable. Indeed the smallest difference between two entries is 0.0001 (entries $a_{1,2}$ and entry $a_{2,1}$ in the matrix with the means), this means that we need a standard deviation that is at most 0.00002 if we want the outliers to be at most 5% (as discussed in Section 4.1). As Table 1 shows, it is with more than 10 000 runs that such a standard deviation can be achieved. With the mentioned standard deviation rule, this standard deviation will be reached after 10^{11} challenges which is impossible to perform in acceptable time. We decided to make the average after 100 000 challenges, which gives a 0.3% error.

5 Results

In this section the results achieved with the different search methods will be discussed. Each of the graphs in this section are based on three rooms with respective hit probabilities [0.9 0.5 0.7].

5.1 Forecasting procedures

The forecasting procedure is important to analyze, because it show results that were not expected.

In Figure 3 the plot has been made on 20 updates of the beliefs, and based on 100 000 challenges to get one entry in the matrix. It compares the performance of two implementations of the F-1 procedure. One where only the room that has the highest chance to contain $\mathcal{H}$ according to the formula of F-1 is searched, and one where each room gets searched with some positive probability according to the strength of the corresponding beliefs. As one can observe, the method where the best room gets searched, performs better than the other one.

In Figure 4 the plot has been made on 20 updates of the beliefs, and based on 100 000 challenges to get one entry in the matrix. It compares the performance of two implementations of the F-2 procedure. One where only the room that has the highest chance to contain $\mathcal{H}$ according to the formula of F-2 is searched, and one where each room gets searched with some positive probability according to the strength of the corresponding beliefs. Here one can also observe that the method where the best room gets searched, performs better than the other one.

The two figures show that selecting the room that has the higher chance according to the search method to contain $\mathcal{H}$, is better than having a probability distribution over the three rooms for the search. This can be explained because it is an average number of steps, and taking a probability to select a room where the object has a relatively small chance to be, will increase the average number of steps. As the value of the game is directly linked to the average number of steps, a probabilistic procedure of searching will increase the average and therefore the value of the game.

5.2 Value of the game

The performance of the search methods on the value of the game will be analyzed here.

In Figure 5 the plot has been made on 20 updates of the beliefs, and based on 100 000 challenges to get one entry in the matrix. It compares the performance of the three basic forecasting procedures: F-0, F-1 and F-2.

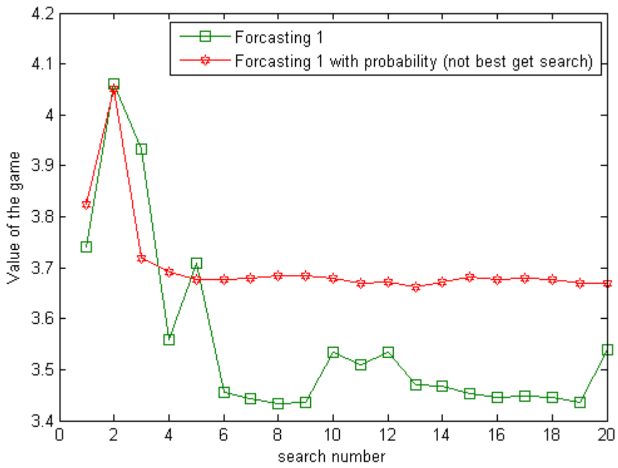

Figure 3: In green (square shape) an F-1 procedure where the room that gives the best expected chance to find the hider is searched. In red (star shape), F-1 also but the rooms get searched with probabilities according to the chance of finding the hider with the current method.

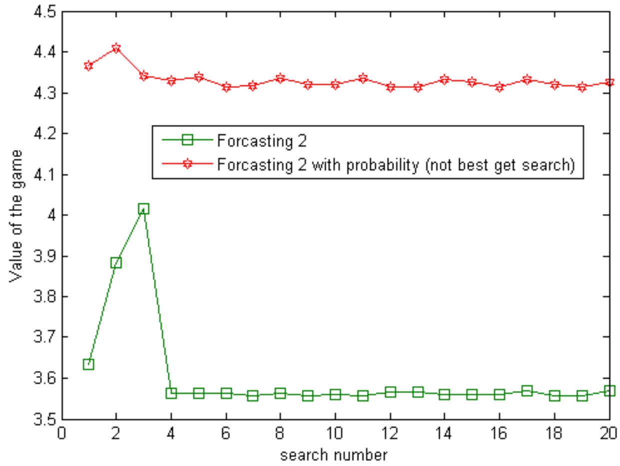

Figure 4: In green (square shape) an F-2 procedure where the room that gives the best expected chance to find the hider is searched. In red (star shape), F-2 also but the rooms get searched with probabilities according to the chance of finding the hider with the current method.

In this graph, only the room that has the highest belief is searched. It can be seen that F-1 performs better than the two other procedures. It is an effect that is not intuitive, because one could expect that F-2 would perform better than F-1.

This fact can, however, be explained. The two step horizon has as objective to search in the room that would, after two searches, give the highest probability to find $\mathcal{H}$. In such a calculation, it does not matter what room to search first and second, just searching the two rooms needs to be done. As shown in Figure 6, if $\mathcal{S}$ chooses room 1 first, he has 3 chances out of 4 to find $\mathcal{H}$ within two steps, otherwise, if $\mathcal{S}$ chooses room 2 first, he has also 3 chances out of 4 to find $\mathcal{H}$ within two steps. So, indeed the order does not matter. As it does not matter, room i and j will both have 50% chance to be searched. This is due to the assumption made in Section 2.5 that, if some rooms have equal chances to contain $\mathcal{H}$, then the choice of the room to search is made randomly. In the F-1 search method, only one of these rooms will be searched: the one that gives the best one step chance to contain $\mathcal{H}$. Moreover, after each selection of a room, the calculation is re-evaluated. It means that room i might have a better one step horizon chance to contain $\mathcal{H}$ but will not especially be searched. As room i and j give the best two step outcome, room j can be chosen to be searched first with probability $\frac{1}{2}$. After the choice, the calculation is re-evaluated, and there is a high probability that room i will re-appear in the best two step outcome, but with another room k. Again, it may appear that room k gets chosen instead of room i, also with probability $\frac{1}{2}$. This is how it can be explained that F-2 does not perform better than F-1. Therefore, another version of F-2 was tried. One where the assumption of Section 2.5 was not taken, but the room that was best according to F-1 that searched. This new version of F-2 has shown a similar outcome to F-1 with a very small variation.

6 Conclusions

In conclusion, we can say that there exists a strategy to minimize the number of steps needed to find a hidden person in a set of rooms when there is a probability that the person is not found even if he is hiding in the room that is searched. The corresponding beliefs for such a strategy can be successfully computed with the help of a matrix game but it is totally dependent of the probability to find the person in each of the rooms. The best beliefs selection is not the same between two different search method. As the beliefs totally determines the search procedure, two different discriminations for the choice of rooms to search will give different results. Which means that different search methods give different results when starting with the same initial conditions. In the case of a strategy that looks ahead in time for the selection of the room to search, a one step ahead short sighted view gives better results than a longer forecasting method. This is because in the implementation of the model described, the process of looking ahead is not a planning for the following steps but just a one step discriminant. In the implementation, a lot of parame-

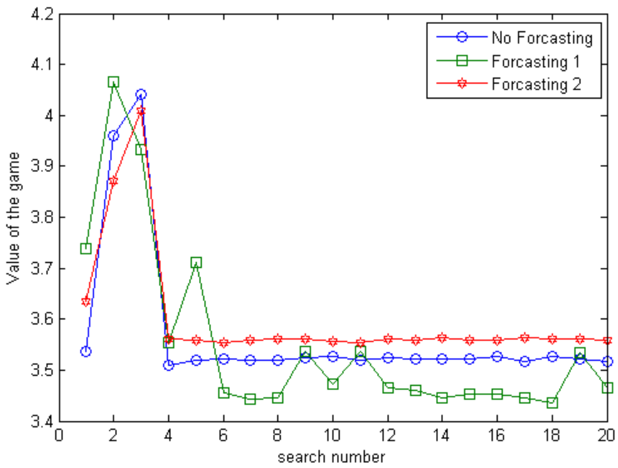

Figure 5: In blue (round shape), F-1, in green (square shape), F-2, and in red (star shape), F-2. In each of the procedure the room that gives the best expected chance to find the hider gets searched.

ters can be tuned or changed which would give different results. That leaves the field open for a lot of future investigations.

7 Further research

Here, some aspects that would be interesting to explore in further research are presented and discussed.

7.1 Forecasting procedure

In the model, F-2 is an overlapping procedure. The calculation of the room to search next is computed, but after every search, the search is re-evaluated, this means that F-2 is not a planning for two following steps but only a discrimination for the one time next search. The impact of an n-forecasting procedure could be studied if it was taken as a plan for the n next steps of the search. This could give a better performance for F-2 and we might see that it outperforms F-1.

In our case, the room that is searched is the one that gives the highest chance to find $\mathcal{H}$ after n step (depending of the search method). It would be interesting to look at the lowest expected remaining number of steps to find $\mathcal{H}$, instead of the highest chance to find $\mathcal{H}$ after n steps. This could be tried as a planning but also as a one time calculation as it is implemented in this model. This could lead to an improvement of the performance of F-2.

7.2 Memory factor

In the implementation of the method, the memory factor has been taken as $1 - \frac{1}{n}$ with $n = 2, 3, ...$ Where n is an increasing counter that increases with games that leads

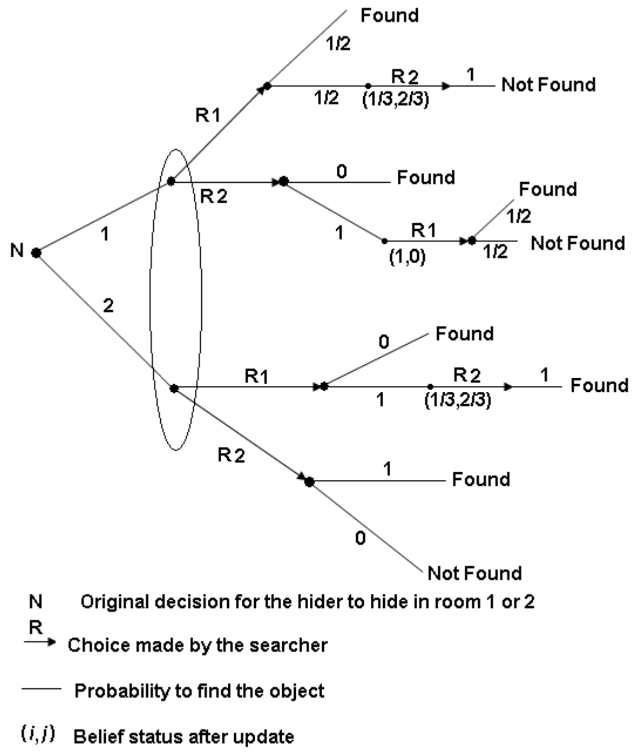

Figure 6: Probability tree for the two step forecasting search method. It concern two room with hit probabilities $\frac{1}{2}$ and 1. The starting beliefs for the rooms is $(\frac{1}{2}, \frac{1}{2})$. The choice of room made by $\mathcal{H}$ is not known by $\mathcal{S}$ and is represented in N

to an increase in the value of the game. The impact of another increase of this memory factor could be studied. For example a linear increase, or an increase corresponding to the percentage of the error, or even an exponential increase. Many possibilities can be seen. This could lead to a better overall performance of the model.

7.3 Accepted steps

In this research, every change in the beliefs was taken into account, one could explore whether or not a step should be taken into account or not and try to see if a process like simulated annealing could be applied in this area.

7.4 Time restricted games

The entire model could be made as a time restricted game where 1 point could be awarded if $\mathcal{H}$ is found within n steps and 0 otherwise. This would change the problem to a maximization problem. The strategy adopted by $\mathcal{S}$ in this situation could be explored. It is a method that could show nice results but that has not been implemented due to time restriction.

Acknowledgments:

A special thanks to Jean Derks from University of Maastricht for the different feedbacks and the really helpful draft of on the subject. Also, thanks to Frank Thuijsman from University of Maastricht for the reflexions and guidance on the field.

References

[1] Chavatal, V. (1983). *Linear Programming*. W.H. Freeman and Company.

[2] Dantzig, G.B., Orden, A., and Wolfe, P. (1955). Generalized simplex method for minimizing a linear form under linear inequality constraints. *Pacific Journal Math.*, Vol. 5, pp. 183–195.

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

[4] Grimmett, G. and Stirzaker, D. (2001). *Probability and Random Processes*. Oxford, third edition.

[5] Walpole, R., Meyers, R., Meyers, S., and Ye, K. (2007). *Probability & Statistics for Engeneers & Scientists*. Pearson Education International, eight edition.