Report LINGI2261: Assignment 3

Group N°107

Student1: Roman Hardy

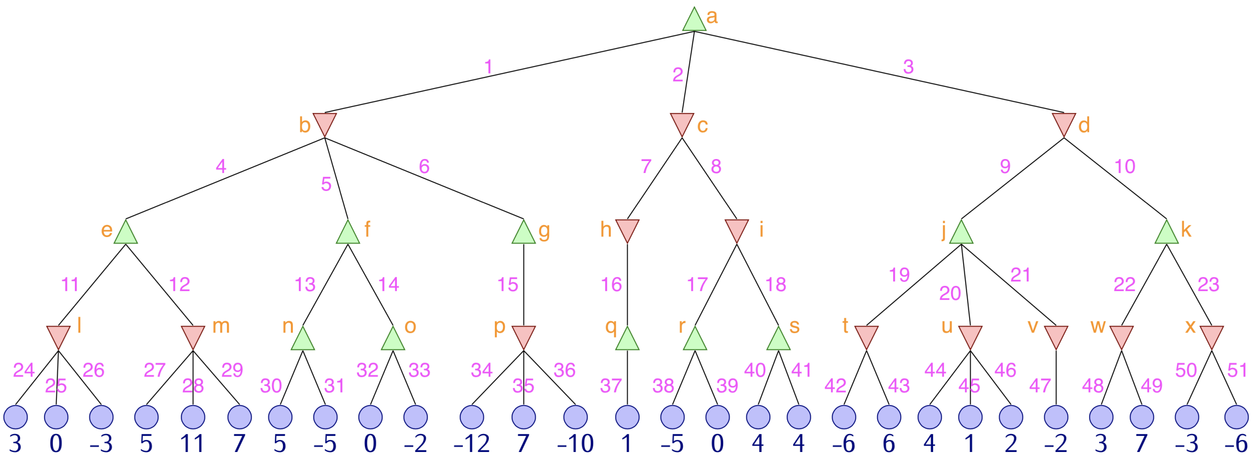
Student2: Héloïse Cordi Claes

April 24, 2022

Answer to the questions by by adding text in the boxes. You may answer in either **French or English**. Do not modify anything else in the template. The size of the boxes indicate the place you *can* use, but you *need not* use all of it (it is not an indication of any expected length of answer). Be as concise as possible! A good short answer is better than a lot of nonsense ;-)

1 Alpha-Beta search (5 pts)

The following figure assigns a unique letter to each node, and a unique number to each branch. Use it to answer the following questions.



1. Perform the MiniMax algorithm on the following tree, i.e. put a value to each node. What move should the root player do? **(1 pt)**

   Assign a numerical value to each node, and indicate the move (i.e. 1, 2, or 3) to perform:

a: 1	f: 5	k: 3	p: -12	u: 1
b: -12	g: 12	l: -3	q: 1	v: -2
c: 0	h: 1	m: 5	r: 0	w: 3
d: 1	i: 0	n: 5	s: 4	x: -3
e: 5	j: 1	o: 0	t: -6	Move: 3

2. Perform the Alpha-Beta algorithm on the same tree. At each non terminal node, put the successive values of α and β. Cross out the arcs reaching non visited nodes. Assume a left-to-right node expansion. **(1 pt)**

   Indicate the successive α and β values of each node in the table below. Separate successive values by a comma (,). Indicate at the bottom the identifiers of the branches that are cut (in increasing order, separated by a comma) (indicate only the branches where the cuts happen, i.e. don't *indicate the branches that are below a cut).*

Node	α values	β values
a	-12, 0, 1	
b		5, -12
c	-12	1, 0
d	0	1
e	-3	
f	5	5
g	-12	5
h	-12	1
i	-12	1, 0
j	0, 1	
k	0, 3	1
l		-3

Node	α values	β values
m	-3	, 5
n	5	5
o	/	/
p		5, -12
q	-12, 1	
r	-12, 0	1
s	-12, 4	0
t	0	-6
u	0	1
v	1	
w	0	1, 3
x	/	/

Cuts: 14, 23, 31, 41, 43

3. Do the same, assuming a right-to-left node expansion instead. **(1 pt)**

Node	α values	β values
a	1	
b	1	
c	1	0
d		3, 1
e	1, 5	
f	1, 5	
g	1	
h	1	0, 1
i	1	4
j	-2, 1	3
k	-6, 3	
l	5	-3

Node	α values	β values
m	1	5
n	1, 5	
o	1	
p	1	-10
q	1	0
r	1	4
s	1, 4	
t	1	3, -6
u	-2	3, 1
v		3, 2
w	-6	3
x		-6

Cuts: 24, 25, 34, 35

4. Can the nodes be ordered in such a way that Alpha-Beta pruning can cut off more branches (in a left-to-right node expansion)? If no, explain why; if yes, give the new ordering and the resulting new pruning. **(1 pt)**

   Insert an image below containing the reordered tree, with successive α/β values indicated next to each node, and where the branches that are cut by the algorithm are crossed out. This may either be an edited version of `minimax.png` *(using paint, gimp, etc.), a photograph of a drawing you made by hand, etc. In any case, the image must be* ***clear*** *in order to be graded.*

We need to put the node with the best value (depending on whether it's min or max turn) at the left of the tree (because the left-to-right DFS will search there first) to maximise the pruning.

5. How does Alpha-Beta need to be modified for games with more than two players? **(1 pt)**

The alpha-beta will store a value for each of it's player (it currently store alpha for one player and beta for the other). The tree is generated like the classic alpha-beta, and every agent try to minimize the others, and maximise itself (considering a 1 vs 1 vs 1 game). The pruning will be more complex and less efficient than with the classic alpha-beta algorithm.

2 Fanorona (35 pts)

2.1 A Basic Alpha-Beta Agent (5 pts on INGInious; nothing to report)

2.2 Evaluation function (5 pts)

5. What are the weak points of the evaluation functions of your basic agent? **(2 pts)**

   Please list your ideas using bullet points, e.g. something of the form:

   - ***Name of your idea***

     Possibly a short explanation describing your idea... Go straight to the point!

- **Only one field is taken into account**

  We only look at the number of opponent pieces the player has managed to capture. We are blind to other important factor, such as our pieces left. Also we have high probability of ending with multiple nodes having the same evaluation at the end of the branch and one is randomly choose.

There is not so much to tell about the first evaluation function, because it was trivial. We could list all the thing the function didn't do, but that would be redundant with the next question.

6. As described in the class, an evaluation function is often a linear combination of some features. Describe new features of a state that can be interesting when evaluating a state of this game. **(2 pts)**

   As for the previous question, list your ideas using bullet points.

- **The number of our pieces left**

  Winning with one or 22 pieces is the same, but because we cannot generate the complete search tree, we would rather go to a direction where we keep our pieces. Also, if we reach 50 boring moves, the winner is the player with the most pieces left. We don't want to loose 3 pieces to capture 1.

- **If the next move is a steal**

  If the next move we made is a steal, we want to go in that direction. It's the main point of the game.

7. Describe precisely your improved evaluation function. **(1 pt)**

Our evaluation function is a linear combination of our pieces left, the opponent's pieces left and if the next move is a steal or not.

2.3 Successors function (4 pts)

8. Give an upper bound on the maximum number of actions that can be performed in any given state. If you can't, explain how you could estimate what is the average number of successors and estimate it. **(1 pt)**

Put your numeric answer here (if you can't give a number, put an "X").

5.5

Explain your answer below.

The game being complex, it is difficult to give a non-trivial upper bound. We choose to estimate it, by printing the number of actions returned by FanoronaRules.get_player_actions and taking it's mean value.

9. What do you loose by ignoring some of the successors? **(1 pt)**

We prune the tree but not with the alpha-beta style, so this affect the final result. This means that the algorithm is no longer complete if the tree is finite. Because the tree is not finite (we stop at a certain depth), we can accept it.

10. Could reordering the successors help the performance of your agent? If so, in what order should the successors be returned in order to help Alpha-Beta prune the search tree? **(1 pt)**

We need to reorder the successor such as the best move is the first visited (on the left side of the tree on a classic DFS). The best move depend on whether it is our turn (maximum) or the opponent's turn (minimum). The problem is that we need the bottom layer to be ordered and we can only make that once the tree is generated. It makes it difficult to implement.

11. Describe you successor function. **(1 pt)**

The successor function return every possible action, with it's new state associated.

2.4 Cut-off function (4 pts)

12. The cutoff method receives an argument called depth. Explain precisely what is called the *depth* in the minimax.py implementation. **(1 pt)**

It's the depth in the search tree, so it is the number of move we predicted. A depth of 4 says that we have predicted 4 moves so far (0 is the root, current state, 1 the next move, and so on).

13. The cutoff function is also the one responsible for identifying which states lead to the end of the game (and so possible victory or defeat). Usually, the end is obtained when one player has lost all of its pawns. However, a situation where each player has a limited number of pawns could lead to an infinite game, where each one is able to escape the other. By convention, the victory is then given to player which has the most pawns. In the implementation, this is dealt with by ending the game when 50 "boring moves" (i.e. no capture of an opponent pawn) have occurred. Why could such an approach be problematic for an Alpha-Beta agent? How could you remedy this? **(1 pt)**

We simply return True when the game reach 50 boring moves. It is therefore treated like a classic ending of the game (and the tree don't look further).

14. In the Fanorona contest your agent will be credited a limited time. How do you manage that time? How can you make sure that you will never timeout? Explain how you can use *iterative deepening* to avoid timeouts. **(1 pt)**

A slow agent was impractical to implement, because testing it locally was really difficult, or impossible in certain cases when the game crashed. Therefore we tried to implement a relatively fast agent, and we do not have to worry about the time credit.
We could use iterative deepening to get a solution under a definite time, and exceeding the maximal depth at every iteration until we reach the credited time. We didn't implement iterative deepening.

15. Describe your cut-off function. **(1 pt)**

The cut-off function return true when the game is ended (50 boring moves or victory) and when the depth exceed a certain limit (6). We also ignore successor when there is too much boring move (to avoid one piece to circle around and do nothing useful). Boring moves are accepted when the game is ending (few pieces left for the opponent), because we might need to walk a little bit to get thoses last pieces.
Also, at depth 0, we return false, otherwise the minimax algorithm returns None instead of an action.

2.5 A Smart Alpha-Beta Agent (6 pts on INGInious)

2.6 Contest (6 pts on INGInious; 5 pts for report)

18. Describe concisely your super tough agent. **(5 pts)**

We tried to sort the list returned by the successor function to optimize the pruning but it didn't change much of the final result, sometimes even leading to worst situation. The idea was to sort the list based on the evaluation function, and sorting it in descending order if it was Max playing and descending order if it was Min playing.

Quick note on why the report is half empty
I (Roman) am aware that this report and assignment is below your expectation. I had to made most of it by myself, my partner helped me with the part 1 (Alpha-Beta search) for the questions 1, 2 and 3. She was supposed to answer the 4th question, hence the lack of graph. I answered to all the questions after that and implement the functions as well. I waited news from her since Wednesday (20/04) to continue and improved the agent, but I received none, so I finished what I could this afternoon.
I don't deny my part of responsibility in this, I should have been more insistent and started the assignment earlier.
I am disappointed because the first two assignments went well. I warned her about this note I left in the end of the report, but I doubt that she will come to remove it.