

Inteligencia Artificial  
Proyecto II  
**Othello**

Stefano De Colli  
09-10203

Oskar González  
09-10351

Karen Troiano  
09-10855

Mayo 29, 2014

El presente proyecto se basó en implementar los algoritmos de búsqueda: Minimax,  $\alpha$ - $\beta$  Pruning, Negamax, Negamax con poda  $\alpha$ - $\beta$ , Scout y Negascout para conseguir el valor de juego de diferentes estados del conocido juego de estrategia Othello (específicamente Othello 6x6).

Las pruebas consistieron en correr los seis algoritmos sobre los estados correspondientes a la *Variación Principal* del juego (cuyo valor de juego es  $-4$ ) y establecer la profundidad de búsqueda máxima que cada uno podía resolver en un período de tiempo prudente.

Los valores obtenidos fueron:

Algoritmo de búsqueda	Profundidad máxima.
Minimax	17
Minimax con poda $\alpha$ - $\beta$ *	19
Negamax	17
Negamax con poda $\alpha$ - $\beta$ *	19
Scout	21
Negascout	22

Tabla en función de segundos:

Pasos prev.	miniMax	miniMaxAB	negamax	negamaxAB	scout	negaScout
<b>23</b>	0,01	0	0	0	0	0
<b>22</b>	0,02	0	0,02	0	0	0
<b>21</b>	0,07	0	0,08	0	0,01	0
<b>20</b>	0,4	0,03	0,4	0,04	0,03	0,02
<b>19</b>	1,52	0,04	1,5	0,05	0,04	0,04
<b>18</b>	9,02	0,13	9,21	0,13	0,09	0,08

<b>17</b>	60,68	0,51	62,83	0,6	0,29	0,27
<b>16</b>	INF	0,91	INF	1,07	0,52	0,51
<b>15</b>	INF	7,94	INF	8,66	3,5	3,16
<b>14</b>	INF	31,47	INF	26,9	13,59	11,56
<b>13</b>	INF	51,53	INF	56,66	33,518	28,001
<b>12</b>	INF	INF	INF	INF	118,11	105,06

### Conclusiones:

Se observa que, Minimax puede resolver instancias de Othello 6x6 de profundidad entre 16 y 17. Vemos una mejora sustancial al correr  $\alpha$ - $\beta$  Pruning, dado que en el mismo período de tiempo, este algoritmo puede resolver instancias de hasta 20 niveles de profundidad. El mismo caso ocurre al comparar Negamax con Negamax  $\alpha$ - $\beta$  Pruning. Esto se debe a que gracias a la poda  $\alpha$ - $\beta$  que hay en estos algoritmos la cantidad de nodos visitados se reduce sustancialmente.

Para el caso de Minimax y Negamax, se nota una diferencia notable en relación de tiempo en las profundidades visitadas, esto se debe a que Negamax aprovecha las bondades de la propiedad  $\max(a,b) = -\min(-a,-b)$  haciendo el algoritmo Negamax mucho más eficiente ante Minimax. Esta relación de tiempos y desempeño también es notoria con  $\alpha$ - $\beta$  Pruning y Negamax con  $\alpha$ - $\beta$  Pruning.

Luego los resultados obtenidos con Scout fueron mejores que aquellos obtenidos con los algoritmos de  $\alpha$ - $\beta$  Pruning, aun cuando el primero no se puede decir que es más fuerte que los últimos, es posible que este resultado se deba al recorrido que se obtiene de la implementación, por último el NegaScout mejora aún más los tiempos de los algoritmos anteriores, logrando resolver a partir de la profundidad 12 en el menor tiempo entre los algoritmos implementados.