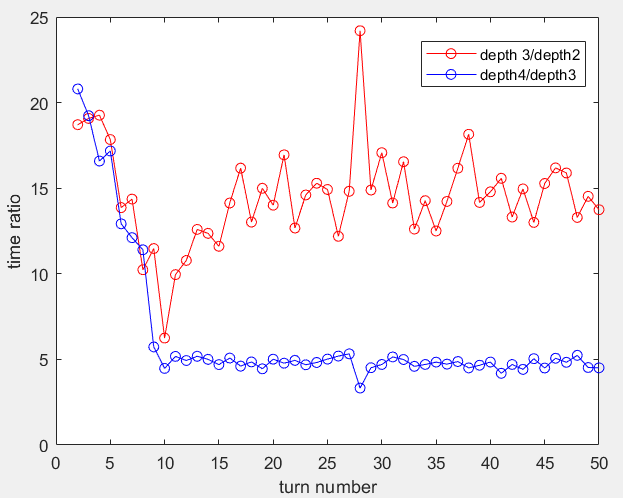
**Global Player Strategy:**

* We get the global time and calculation average time for single move, assuming N=35 moves:
* We want to take into account the fact that for higher turns we need less time. so we   
  are using a *turn factor function* to factorize the move time.
* Factor function:
  + We want a monotonically descending function
  + at we want to get *turn\_factor=1*
  + *at turn=N (the average number of moves) we want to remain with 1 second for move  
    i.e. turn\_factor(N)=1/avg\_time*
  + *So we used an exponent*
  + *from the above conditions, easy to see that:*
* With *move\_time* calculated every turn we use it to set a time limit for a single turn.

**One move time limit Strategy:**

* We performed a lot of experiments and measure the time it takes for every depth and turn.
* We saw that as the turn number is higher most of the time it takes less time to perform a move. That was accounted in a Global Time strategy.
* In addition we saw that the difference between two adjacent depth is around ~20.   
  This fact isn’t so strange cause we know that asymptotically the time of DFS is where   
  b is the branching factor and d is the depth, so asymptotically the ratio between two adjoint depth is :
* This result we sawed in our experiments, as we can see below, the time ratio of two adj. depths is bound by 25.



* Our Code Strategy:
  + Every depth we measure the time it took to perform the Minimax or Alpha-beta loop (*iter\_time)*
  + We continue to the next depth if the
  + We took the factor of 40 (to be on the safe side)
  + In addition we took safety factors of few seconds to be sure that we have enough time to send our move to the game\_wrapper (:

**Part E – Experements:**

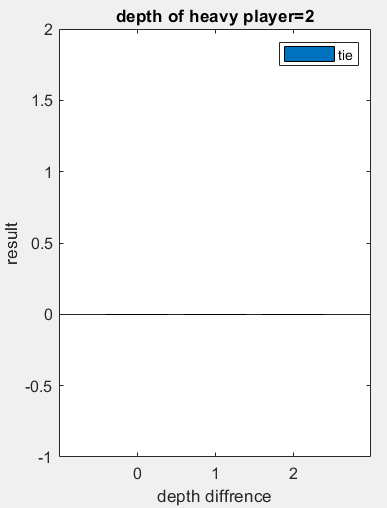
Question 5:

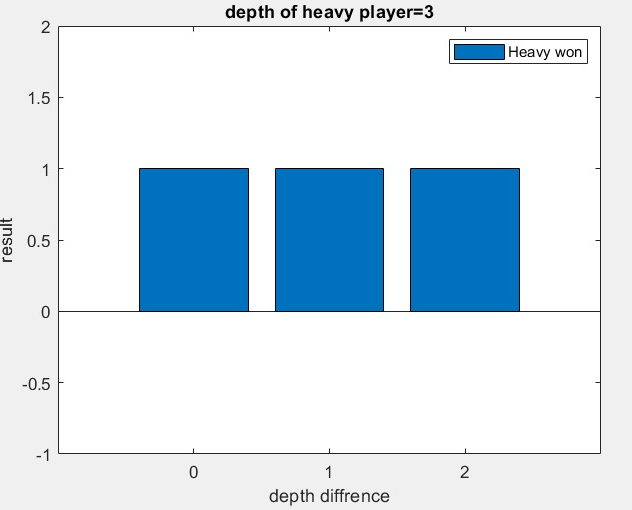
**part a:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Player 1** | **Player 2** | **1.1 sec** | **3 sec** | **5 sec** | **10 sec** | **20 sec** |
| **Alphabeta** | **Minimax** | **0:1** | **0:1** | **1:0** | **1:0** | **tie** |
| **Minimax** | **Alphabeta** | **0:1** | **0:1** | **0:1** | **0:1** | **tie** |
|  |  | **Black is winning**  **(player 2)** | | **Alpha Beta is better** | |  |

* For high enough time limit per move the game should end in a tie- cause the game is quite drawish and both of the player with the same heuristic and if they both reach at least a depth of 3-4 they just not allow the other player to create a mill and the game will end in a tie. This can explain the 20 sec result
* In a low time per move, both of the player can’t get high depth and most of the time they stay in the same one (1.1 sec case). We assume that our heuristic is better for Black – better defending and stopping the rival than attacking (But this is only an assumption).
* For middle time game (5 sec and 10 sec) Alpha-beta is winning. It is enough time for alpha-beta to get more depth than the minimax player and winning the game.

**part b:**

Results:



* For depth=3 we won with our Heavy Player in all the depth diffrences.
* For depth=2 all the games ended in a tie (Program Paused)

Discussion:

1. Depth 3 experiments are understandable – the light heuristic wasn’t informed enough- it doesn’t have parameters such as – the difference between number of blocked soldiers, double mills, 3 piece configurations. So even the light player get higher depth, he is so far away from the endgame/terminal positions and without good evaluation of the positions he is loosing to the more informed player.
2. Depth 2 is strange, especially after the discussion of the depth 3 results. In all the games we stoped the game cause it’s just gets in a loop where no one want to loose and no one going for a new mill. The more informed player is usually in a better position but he can’t win. The explanation is that for winning he need to move one of the soldiers in a mill to construct a mill in the next move. **But** our heavy player is a two depth player so it only calculate his move and the rival’s one, so he can’t reach the mill (which constructed only in depth=3).   
   **The conclusion is that depth=2 isn’t enough for our players and heuristic.**