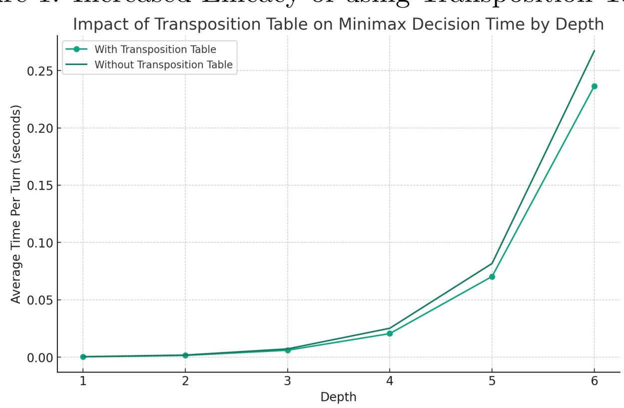
U2106764: Adversarial Search Evaluation

The Minimax algorithm, crucial for AI decision-making in two-player games, faces challenges with computational load as game complexity rises. Alpha-Beta pruning, refining Minimax, curtails this by cutting extraneous search branches, thus enhancing efficiency. This study quantifies such improvements, with metrics including move time and node expansion, demonstrating the optimisation's practical advantages in intricate game scenarios and its pivotal role in sophisticated AI strategies.

1 Evaluation of Minimax without pruning

Initially, the Minimax algorithm's performance was hindered by slow computation, particularly with large decision trees. Integrating transposition tables, which store and recall past calculations, mitigated this by avoiding redundant processing[1]. Rigorous testing against a random AI showed a 15% reduction in average move time, confirming the transposition tables' role in streamlining Minimax, enhancing speed without strategic compromise.

Figure 1: Increased Efficacy of using Transposition Tables



Furthermore, by varying board sizes and layouts, the analysis explores how spatial changes affect computational efficiency and decision-making accuracy (refer to figure 2 below). The analysis reveals that adding rows to the game space minimally affects performance, as it does not increase move options. In contrast, adding columns exponentially raises the average time per move due to a greater choice of moves for the AI. Additionally, changing the number of pieces required to win shows an initial spike in processing time, then a decline. This is because fewer potential winning configurations are needed to be evaluated: for instance, from 120 possible ways with a win condition of 3 pieces to just 16 when the requirement is 7 pieces, reducing the computational effort.

Figure 2: Investigating effects of varying board size

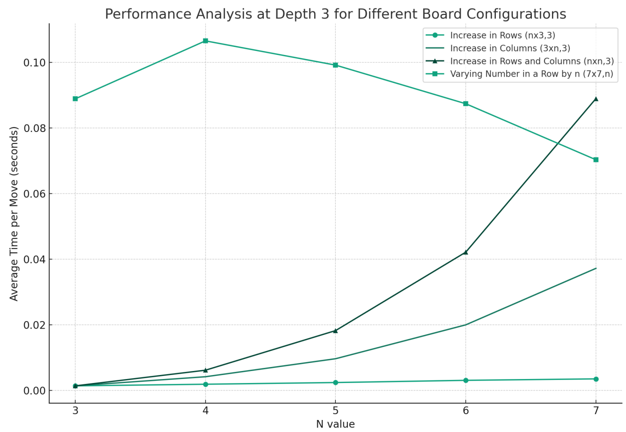


Table 1 presents an analysis where the AI competes against a random opponent on a 6x7 board with a victory condition of aligning four in a row. This study specifically examines how escalating the search depth affects AI performance. The results demonstrate a symbiotic relationship: as the depth increases, there is a corresponding rise in both the average time per move and the average number of nodes expanded. This correlation highlights the complexity added with each incremental depth level, impacting the AI's decision-making process.

Depth	Average Time per Move (seconds)	Average Nodes Expanded	Win Rate (percentage)
1	0.00446	2273.85	99
2	0.03455	19700.99	100
3	0.22467	125323.23	100
4	1.44672	793968.39	100
5	9.76164	5369450.44	100

Table 1: Experiment for evaluating how varying depth affects Minimax performance.

2 Evaluation of Minimax with alpha-beta pruning

Table 2 replicates the experiment from Table 1, this time implementing Alpha-Beta pruning in the AI's algorithm. The results affirm a consistent symbiotic relationship between increased depth and both the average time per move and the number of nodes expanded. Notably, while the relationship remains, the actual count of nodes expanded is substantially lower compared to the scenario without pruning. This reduction is attributed to the efficiency of Alpha-Beta pruning, which effectively trims the search space, thereby enhancing overall computational efficiency.

This conclusion is further corroborated by the findings presented in Figure 3, which visually depicts the impact of pruning on the algorithm's performance. The application of Alpha-Beta pruning to the Minimax algorithm markedly enhanced performance, with modest gains at shallow depths

Depth	Average Time per Move (seconds)	Average Nodes Expanded	Average Nodes Pruned	Win Rate (percentage)
1	0.00197	2159.8	0	100
2	0.00860	11574.56	1613.25	100
3	0.03652	53065.4	6318.43	100
4	0.12315	166281.67	31204.01	100
5	0.52786	777714.81	121436.11	100

Table 2: Experiment for evaluating how varying depth affects Minimax Alpha-Beta Pruning performance.

transitioning to significant improvements at deeper levels. The efficiency boost reached approximately 21.83% at a depth of 2 and soared to around 90.35% by depth 8, underlining the pruning method's effectiveness in optimizing computational demands for complex strategic evaluations.

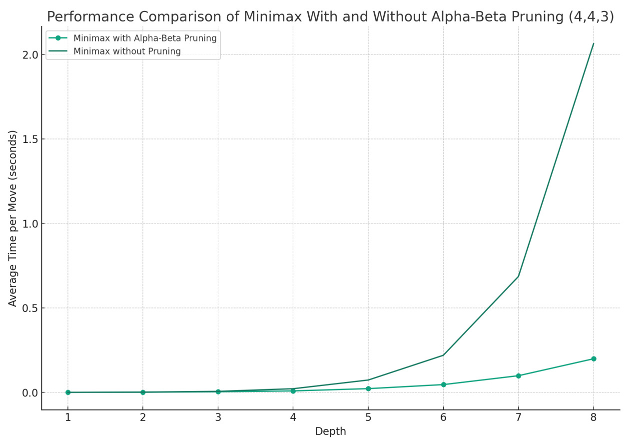


Figure 3: Investigating efficiency at greater depths

3 Discussion and Conclusions

The efficacy of the Minimax algorithm, augmented with Alpha-Beta pruning, is particularly prominent at deeper levels, where pruning markedly boosts efficiency [2]. This method adeptly harmonises computational resources with strategic depth, proving highly beneficial in intricate decision-making scenarios. Enhancements could include the generation of more streamlined Zobrist keys [3] and transitioning the game board array to a string format, thereby reducing the complexities associated with array manipulation.

Prospective enhancements to this study should concentrate on refining the evaluation function. The existing implementation favours middle column placements but fails to recognise the tactical advantage of numerous empty rows. Acknowledging these empty rows as a pathway to expedite victories could further enhance the algorithm's effectiveness, yielding a more sophisticated strategy for move selection and game progression.

References

[1] K. Yoshizoe, A. Kishimoto, and T. Ishida. Effective use of transposition tables in stochastic game tree search. In *2007 IEEE Symposium on Computational Intelligence and Games*. IEEE, 2007. [Accessed 01/01/2024].

[2] Samineh Bagheri, Markus Thill, Patrick Koch, and Wolfgang Konen. Online adaptable learning rates for the game connect-4. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(1):33–42, 2016.

[3] D.R. Mason, T.S. Hudson, and A.P. Sutton. Fast recall of state-history in kinetic monte carlo simulations utilizing the zobrist key. *Computer Physics Communications*, 165(1):37–48, 2005.