Adversarial Search

1 Introduction

To evaluate how Minimax scales with increasing game size, we measured the execution time for computing the first move from the initial game state for several (m, n, k) configurations. We then compared these results between setups with and without α–β pruning.

We recorded:

- Execution time for computing the optimal move
- Number of states (nodes) visited during search
- Effectiveness of α–β pruning in reducing computation

2 Results Overview

The following experiments were conducted.

Table 1: Execution times and nodes visited with and without pruning for increasing values of m, n, and k.

$m \times n \times k$	**Without Pruning**		**With Pruning**	
	Time (s)	Nodes Visited	Time (s)	Nodes Visited
3×3×3	3.257	549,946	0.1266	21,731
3×4×3	4751.2218	276,911,233	0.5891	84,467
4×3×3	2688.37	276,911,233	2.0565	284,487
4×4×3	–	–	49.632	5,958,619

As the board grows, the branching factor and depth of the game tree both increase, making Minimax computationally expensive. α–β pruning significantly reduces the number of evaluated states, especially for larger boards, while also reducing execution times.

2.1 Observations

1. 3×3×3

- Minimax without pruning is fast because the search tree is small.
- Pruning yields only a small improvement.

2. 3×4×3 and 4×3×3

- The tree grows substantially.
- α–β pruning cuts off branches early, reducing time and visited nodes drastically.

3. 4×4×3

- Minimax without pruning does not finish in reasonable time due to exponential explosion.
- α–β pruning computes a result, but still takes much longer than smaller boards.

4. 4×4×4 and larger configurations

- Minimax (with or without pruning) becomes infeasible.
- α–β pruning cannot compute a result in reasonable time.

3 Key Findings

- Minimax execution time grows exponentially with board size.
- α–β pruning dramatically reduces:
  - number of states visited
  - execution time
- For large boards (e.g., 4×4×3), Minimax without pruning becomes computationally infeasible.

4 Discussion

4.1 Effect on Time as m, n, k Increase

The time required for Minimax action selection increases significantly as the board size increases, consistent with its theoretical complexity.

- On the 3×3×3 board, Minimax without pruning completes in ∼3.26 s, while the pruned version takes only ∼0.13 s.
- For 3×4×3, non-pruned Minimax jumps to ∼4751 s (1.3 hours), while pruning reduces this to ∼0.59 s.
- The non-pruned Minimax for 4×4×3 is infeasible due to combinatorial explosion.

4.2 Effect on Number of States Visited

The number of states visited shows the same trend as execution time.

- 3×3×3 without pruning visits ∼550k nodes, whereas 3×4×3 visits ∼277 million.
- With pruning, 3×3×3 visits only ∼22k nodes; 3×4×3 visits ∼84k.
- 4×3×3 with pruning visits ∼284k nodes, showing geometry impacts branching factor.

4.3 Effect of α–β Pruning

- Pruning reduces execution time by one to four orders of magnitude.
- It also drastically reduces nodes visited.
- For 3×4×3, pruning reduces time from ∼4751 s to 0.59 s and nodes from ∼277M to ∼84k.
- Without pruning, 4×4×3 is infeasible, but pruning yields a result in ∼50 s.