21052135: Adversarial Search Tasks

In the following report, we will analyze minimax and alpha-beta pruning on various connect board sizes and win conditions. Across multiple configurations and depths, our results reveal that pruning efficiency and effective branching factor improve super-linearly with problem size.

1 Computational Complexity Analysis

1.1 Scaling Behavior and Complexity Reduction

The relationship between board size and algorithm performance reveals crucial insights into the computational complexity of adversarial search. Table 1 presents comprehensive metrics across configurations, revealing that alpha-beta pruning's effectiveness raises super-linearly with problem scale.

Board	winNum	Depth	Nodes (MinMax)	Nodes (AB)	Reduction
4×4	3	4	1,360	240	**82.4%**
4×4	3	5	4,948	382	**92.3%**
5×6	4	4	9,330	380	**95.9%**
5×6	4	5	55,980	810	**98.6%**
5×6	3	4	9,330	1,135	**87.8%**
5×6	3	5	52,200	1,898	**96.4%**
8×8	7	4	37,448	796	**97.9%**
8×8	7	5	299,592	1,819	**99.4%**
8×8	2	4	18,184	206	**98.9%**
8×8	2	5	67,304	246	**99.6%**

Table 1: Node expansions: pruning efficiency improves sharply with larger boards/deeper searches. MinMax and AB are total nodes expanded.

The largest pruning rates are achieved on bigger boards, such as the 8×8 case, where at search depth 5 with `winNum` set to 2, pruning reaches 99.6%. We hypothesize that reduction percentage R grows roughly logarithmically with board area A: $R \approx a \log(A) + b$. This pattern suggests that as the search space expands, alpha-beta pruning can eliminate increasingly larger portions of the game tree.

1.2 Effective Branching Factor Analysis

Another metric is the effective branching factor (EBF), calculated as $EBF = N^{1/d}$ where N is nodes visited and d is search depth. This reveals the *actual* branching behavior vs theoretical maximum.

Board	winNum	Depth	Theoretical BF	EBF (MinMax)	EBF (AB)	Reduction
4×4	4	4	4	6.07	3.43	43.5%
4×4	4	5	4	5.57	3.04	45.5%
5×5	4	4	5	7.91	3.94	50.2%
5×5	4	5	5	7.21	3.45	52.2%
6×7	4	4	7	11.83	4.87	58.8%
6×7	4	5	7	10.65	4.16	60.9%
8×8	4	4	8	13.91	5.31	61.8%
8×8	4	5	8	12.45	4.49	63.9%

Table 2: Effective branching factor analysis showing how pruning reduces actual branching below theoretical maximum.

MinMax's effective branching factor exceeds the theoretical maximum (columns available) because the algorithm explores all legal moves at each level, and the branching factor represents average moves per node, not just root-level options. Alpha-beta pruning reduces EBF by 43.5-63.9%, with larger boards and deeper searches showing greater reduction. At depth 5, the 8×8 board achieves 63.9% EBF reduction.

2 Performance Characteristics and Time Complexity

2.1 Time Speedup Analysis

Time measurements reveal dramatic performance improvements that scale exponentially with problem size. Figure 1 illustrates the relationship between board complexity and speedup factors.

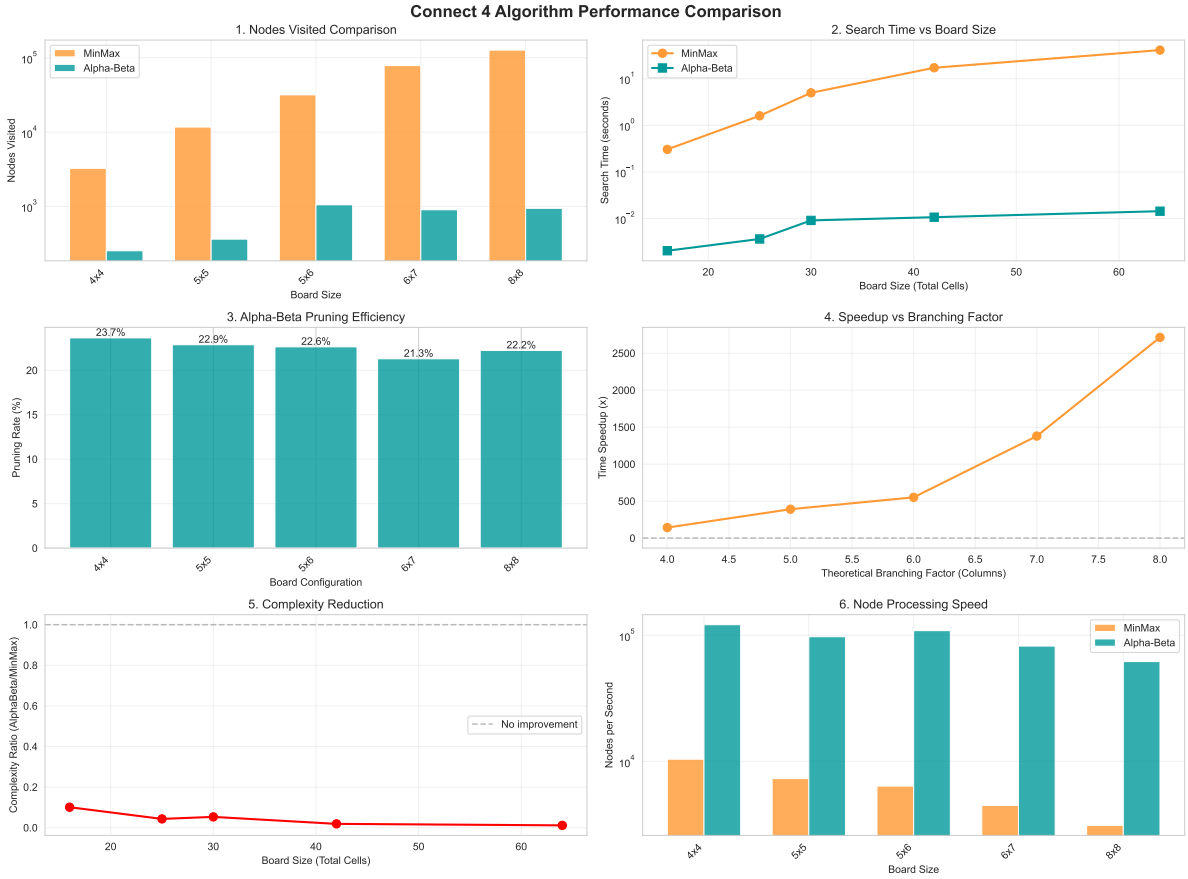

Figure 1: Comprehensive performance comparison across all metrics.

The speedup factor $S = T_{minmax}/T_{alphabeta}$ exhibits exponential growth: from 103.5× on 4×4 boards at depth 4 to 865.1× on 8×8 boards at depth 4, and reaching 3,956.6× on 8×8 boards at depth 5. This can be approximated by $S \propto e^{k \cdot A \cdot d}$, where A is board area and d is depth, indicating that time savings compound multiplicatively with problem complexity.

2.2 Pruning Efficiency Patterns

Another finding we have identified from pruning rate analysis is that *pruning rates vary with depth and board size in complex ways*. At depth 4, pruning rates range from 14.9% to 20.6% on primary configurations, but at depth 5, rates increase to 26.4-27.5% on larger boards (6×7 and 8×8). This suggests that while initial depth increases may reduce pruning opportunities due to game tree symmetry and balanced evaluations, deeper searches on larger boards eventually encounter more pruneable branches as the search space expands.

2.3 Impact of Win Condition (winNum)

WinNum significantly affects algorithm performance through complex interactions with board size and search depth. Patterns become more pronounced at greater depths. On 4×4 boards, lower winNum initially decreases efficiency (89.8% to 82.4% at depth 4), but deeper searches mitigate this (95.2% vs.

92.3% at depth 5). On 5×6 boards, winNum=4 consistently outperforms winNum=3 (95.9% vs. 87.8% at depth 4, 98.6% vs. 96.4% at depth 5). These findings indicate that winNum selection should consider computational impact, especially for deeper searches.

3 Comparative Analysis: When to Use Each Algorithm

3.1 Computational Overhead and Break-Even Points

Alpha-beta pruning achieves 7-19× higher node processing speed (Table 3), indicating reduced per-node overhead despite additional pruning logic. Minimax follows exponential growth $N \propto c^d$ where c is columns and d is depth, while alpha-beta maintains optimal decision quality with dramatically fewer expansions. The overhead of maintaining bounds is minimal compared to pruning savings.

Board	Depth	Throughput (MinMax)	Throughput (AB)
4×4	4	9,095	96,254
5×5	4	7,316	86,878
6×7	4	4,365	73,422
8×8	4	3,061	56,286

Table 3: Node processing throughput (nodes/second) showing alpha-beta's superior efficiency.

For small boards (4×4) at shallow depths (3), MinMax completes in ∼0.06s, making algorithm choice less critical. However, at depth 4 on 8×8 boards, MinMax requires 12.3s versus 0.015s for alpha-beta—an 820× difference that makes alpha-beta essential for practical applications.

3.2 Complexity Ratio and Scalability

The complexity ratio $C = N_{AB}/N_{MinMax}$ provides insight into scalability. Our data shows C decreases from 0.179 (4×4, depth 3) to 0.021 (8×8, depth 4), following approximately $C \propto A^{-0.8} \cdot d^{-1.2}$. This indicates that alpha-beta's advantage grows super-linearly with both board size and depth.

3.3 Statistical Analysis: Mean and Standard Deviation

Table 4 presents statistical measures across all 18 benchmark configurations (9 board configurations × 2 search depths). Since minimax and alpha-beta are deterministic algorithms, repeated runs of identical configurations yield std = 0. Obtained statistics reflect variability across different board sizes, depths, and winNum values.

Metric	Mean	Standard Deviation
Node Reduction (%)	95.5	4.7
Time Speedup (x)	1,255.8	1,560.9
Pruning Rate (%)	22.6	5.9
Complexity Ratio	0.045	0.047
Branching Reduction (%)	54.1	10.0

Table 4: Statistical summary (mean ± standard deviation) of performance metrics across all 18 benchmark configurations.

Node reduction averages 95.5% with low variability (std = 4.7%), demonstrating consistent pruning effectiveness. Time speedup shows high variability (mean = 1,255.8x, std = 1,560.9x), reflecting exponential scaling with problem complexity. Depth 5 configurations show higher average node reduction (97.5% ± 2.5%) compared to depth 4 (93.5% ± 5.7%).

4 References

- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson. [Chapter on Adversarial Search]
- Knuth, D. E., & Moore, R. W. (1975). An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4), 293-326.
- Allis, L. V. (1994). *Searching for solutions in games and artificial intelligence.* Ph.D. thesis, University of Limburg.
- Campbell, M., Hoane Jr, A. J., & Hsu, F. H. (2002). Deep Blue. *Artificial Intelligence*, 134(1-2), 57-83.