Chain Reaction AI Heuristics Performance Report

Nafis Nahian - 2105007

June 16, 2025

1 Project Overview

This report analyzes the performance of five different heuristic functions implemented for the Chain Reaction AI player. Each heuristic was tested at search depths 1-5, measuring the average time taken for the first 5 AI moves in a game.

2 Heuristic Descriptions

2.1 Weighted Combined (weighted_combined)

Purpose: A balanced approach combining multiple strategic elements.

Algorithm:

- Combines chain reaction opportunity detection with strategic evaluation
- Formula: $1 \times \text{chain\_reaction\_score} + 2 \times \text{strategic\_score}$
- Emphasizes strategic positioning while considering immediate threats

Key Features:

- Balances offensive and defensive play
- Considers both immediate tactical gains and long-term positioning
- Weighted toward strategic evaluation for more thoughtful play

2.2 Orb Count Difference (orb_count)

Purpose: Simple material-based evaluation focusing on orb quantity.

Algorithm:

- Counts total orbs for the player vs opponent
- Formula: $\text{player\_orbs} - \text{opponent\_orbs}$
- Maximizes orb advantage

Key Features:

- Fastest to compute due to simplicity
- Focuses purely on material advantage
- May miss strategic positioning opportunities
- Good baseline heuristic for comparison

2.3 Edge Corner Control (edge_corner)

Purpose: Prioritizes control of strategically important board positions.

Algorithm:

- Assigns different point values based on cell position:
  - Corner cells: +3 points
  - Edge cells: +2 points
  - Center cells: +1 point
- Sums points for all player-controlled cells

Key Features:

- Emphasizes positional advantage
- Corners and edges are safer (lower critical mass)
- Promotes defensive, territory-control gameplay
- May sacrifice immediate tactical gains for position

2.4 Strategic Evaluation (strategic)

Purpose: Comprehensive evaluation considering multiple game factors.

Algorithm:

- **Terminal states**: ±10000 for win/loss
- **Vulnerability analysis**: $-5 + \text{critical\_mass}$ penalty for orbs near critical enemies
- **Safety bonuses**: +2 (edge), +3 (corner), +2 (critical) for safe orbs
- **Material count**: +1 per orb
- **Critical blocks**: $+2 \times \text{block\_size}$ for contiguous critical formations

Key Features:

- Most comprehensive analysis
- Balances offense, defense, and positioning
- Considers immediate threats and opportunities
- Computationally intensive but strategically sound

2.5 Chain Reaction Opportunity (chain_reaction)

Purpose: Focuses on creating and exploiting chain reaction opportunities.

Algorithm:

- Identifies enemy cells that are one orb away from critical mass
- Awards +3 points for each such vulnerable enemy cell adjacent to player orbs
- Emphasizes aggressive, explosive gameplay

Key Features:

- Highly offensive strategy
- Seeks to trigger cascading explosions
- May sacrifice long-term position for immediate gains
- Effective in mid-to-late game scenarios

3 Performance Analysis

3.1 Timing Results Summary

Heuristic	Depth	Avg Time (s)	Min (s)	Max (s)
weighted_combined	1	0.000	0.000	0.000
weighted_combined	2	0.018	0.000	0.053
weighted_combined	3	0.079	0.000	0.227
weighted_combined	4	2.124	0.003	6.363
weighted_combined	5	9.895	0.002	14.967
orb_count	1	0.000	0.000	0.000
orb_count	2	0.011	0.000	0.030
orb_count	3	0.031	0.002	0.076
orb_count	4	0.601	0.000	1.796
orb_count	5	3.850	0.007	6.384
edge_corner	1	0.000	0.000	0.000
edge_corner	2	0.008	0.002	0.020
edge_corner	3	0.081	0.000	0.129
edge_corner	4	1.202	0.000	3.603
edge_corner	5	6.558	0.013	10.999
strategic	1	0.001	0.001	0.001
strategic	2	0.016	0.001	0.043
strategic	3	0.067	0.000	0.195
strategic	4	1.738	0.002	5.205
strategic	5	8.036	0.003	12.367

Heuristic	Depth	Avg Time (s)	Min (s)	Max (s)
chain_reaction	1	0.006	0.000	0.016
chain_reaction	2	0.019	0.000	0.034
chain_reaction	3	0.077	0.003	0.116
chain_reaction	4	1.078	0.002	1.663
chain_reaction	5	2.092	0.000	6.275

3.2 Performance by Depth

3.2.1 Depth 1-2 (Shallow Search)

- **Winner**: orb_count and edge_corner (fastest)
- All heuristics perform similarly at shallow depths
- Computation time is minimal (¡ 0.02s average)

3.2.2 Depth 3 (Medium Search)

- **Winner**: orb_count (0.031s average)
- Performance ranking: orb_count ¡ strategic ≈ chain_reaction ¡ edge_corner ¡ weighted_combined

3.2.3 Depth 4 (Deep Search)

- **Winner**: orb_count (0.601s average)
- Significant performance gap emerges
- Performance ranking: orb_count ¡ chain_reaction ¡ edge_corner ¡ strategic ¡ weighted_combined

3.2.4 Depth 5 (Very Deep Search)

- **Winner**: chain_reaction (2.092s average)
- orb_count becomes second fastest (3.850s)
- Performance ranking: chain_reaction ¡ orb_count ¡ strategic ¡ edge_corner ¡ weighted_combined

3.3 Overall Performance Rankings

3.3.1 By Speed (Fastest to Slowest at Depth 4-5)

1. **Orb Count**: Consistently fastest due to simple calculation
2. **Chain Reaction**: Good balance of speed and tactical awareness
3. **Edge Corner**: Moderate complexity, reasonable performance
4. **Strategic**: Comprehensive but computationally expensive
5. **Weighted Combined**: Slowest due to combining multiple heuristics

3.3.2 By Complexity vs Performance Trade-off

1. **Orb Count**: Simple, fast, but may miss strategic nuances
2. **Chain Reaction**: Good tactical focus with reasonable speed
3. **Edge Corner**: Balanced approach with positional awareness
4. **Strategic**: Most comprehensive but requires more computation time
5. **Weighted Combined**: Most sophisticated but slowest execution

3.4 Performance Optimization Insights

1. **Depth scaling**: All heuristics show exponential time growth with depth
2. **Heuristic complexity**: Simpler heuristics scale better to deeper searches
3. **Game phase adaptation**: Different heuristics may be optimal for different game phases

4 Conclusion

The performance analysis reveals clear trade-offs between computational speed and strategic sophistication. The orb_count heuristic provides excellent speed for time-critical applications, while strategic and weighted_combined offer superior move quality at the cost of computation time. The chain_reaction heuristic emerges as an excellent middle-ground option, providing tactical awareness with reasonable performance characteristics.

For practical implementations, consider using adaptive strategies that select heuristics based on available computation time, game phase, and position complexity. This approach can maximize both move quality and responsiveness across different gameplay scenarios.