Rubik's Cube Solver: Integrating Classical Algorithms with Machine Learning Prediction

Ahmet Burak Çiçek — Advisor: Şefik Şuayb Arslan
CMPE 492, Computer Engineering Department

1. Abstract & Objectives

The Rubik's Cube state space ($|G| \approx 4.3 \times 10^{19}$) presents a unique challenge for heuristic search. This project implements a comprehensive framework integrating **classical solvers** (Thistlethwaite and Kociemba) with an **AI-based predictive system**.

Key Objectives:

- Implementation of Thistlethwaite (Multi-phase) and Kociemba (Two-phase) algorithms.
- Creation of a reproducible benchmarking framework ($N = 5000$ dataset).
- Development of ML models to predict solver performance (runtime, moves, nodes) and enable **adaptive solver selection**.

2. Classical Algorithms: Theory

The solvers reduce the state space via group theory subgroups.

A. Thistlethwaite's Algorithm (Satisfaction) Decomposes the problem into 4 nested subgroups to limit search breadth:

$$G = G_0 \supset G_1 \supset G_2 \supset G_3 \supset \{e\}$$

- Uses 4 phases.
- Guarantee: Predictable, bounded runtime.
- Trade-off: Suboptimal solution lengths (30 moves).

B. Kociemba's Two-Phase Algorithm (Optimization) Optimizes the reduction into two stages using IDA* and Pattern Databases:

- Phase 1: Reach $G_1 = \langle U, D, R2, L2, F2, B2 \rangle$.
- Phase 2: Solve within G_1.
- Trade-off: Near-optimal solutions (19 moves) but heavy-tailed runtime distribution.

3. Classical Benchmark Results

We analyzed the trade-off between **Solution Quality** (Moves) and **Computational Cost** (Time/Nodes) on the midterm dataset ($N = 200$).

Table: Summary Statistics (Mean / Median)

Metric	Thistlethwaite	Kociemba
Runtime (s)	0.016 / 0.015	0.218 / 0.063
Solution Length	25.47 / 30.0	17.92 / 22.0
Nodes Expanded	142 / 149	4,150 / 1,486
Success Rate	100%	100%

Key Findings:

- **Kociemba** provides significantly shorter solutions ($\Delta \approx 7.5$ moves) but exhibits high variance (σ^2 is large).
- **Thistlethwaite** dominates in consistency and speed (Linear scaling).

4. Analysis of Trade-offs

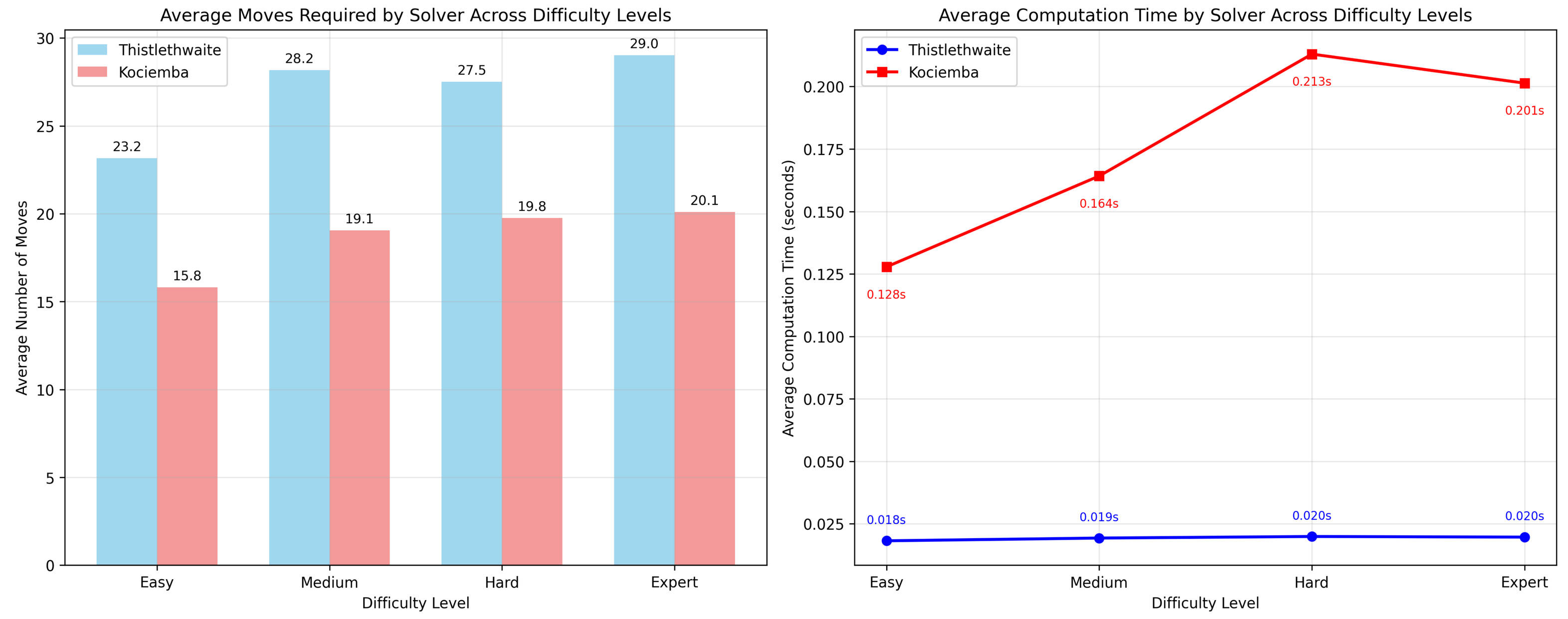


Figure: Mean Runtime and Move Count across difficulty levels. Note Kociemba's runtime spike in 'Expert' categories vs. Thistlethwaite's stability.

- **Kociemba:** produces substantially shorter solutions (18 moves mean / 22 median) but shows *high runtime variance* (median 0.063 s; mean inflated by outliers).
- **Thistlethwaite:** gives longer solutions (25–30 moves) with *predictable, low* runtimes (median 0.015–0.016 s) and tight variance.
- **Practical guideline:** use Thistlethwaite when consistent, real-time performance matters; use Kociemba when minimizing moves is the priority and occasional slow cases are acceptable.

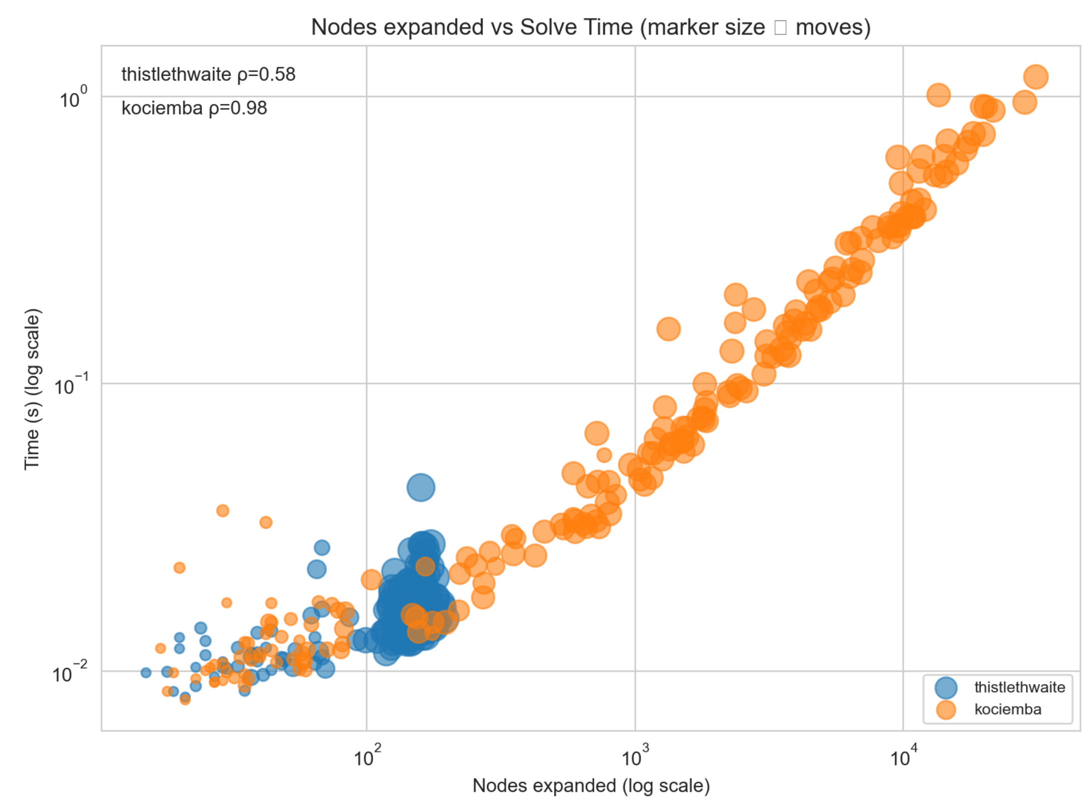


Figure: Nodes vs. Time (Log-Log). Thistlethwaite shows tight linear scaling; Kociemba shows heavy-tailed behavior due to heuristic sensitivity.

- **Thistlethwaite scaling:** points cluster tightly along an almost-linear trend (near $T \propto N_e$); mean/median nodes are low (143 / 150) and runtime variance is small.
- **Kociemba scaling:** shows a heavy-tailed scatter — median cases are moderate but *rare* instances trigger massive node expansion (mean nodes median), driven by IDA*/heuristic sensitivity.
- **Constant-factor trade-off:** Thistlethwaite has slightly higher per-node overhead (table lookups / subgroup checks) but avoids worst-case blowups; Kociemba often has cheaper per-node work yet can require far more nodes on hard instances.

5. ML Prediction Performance

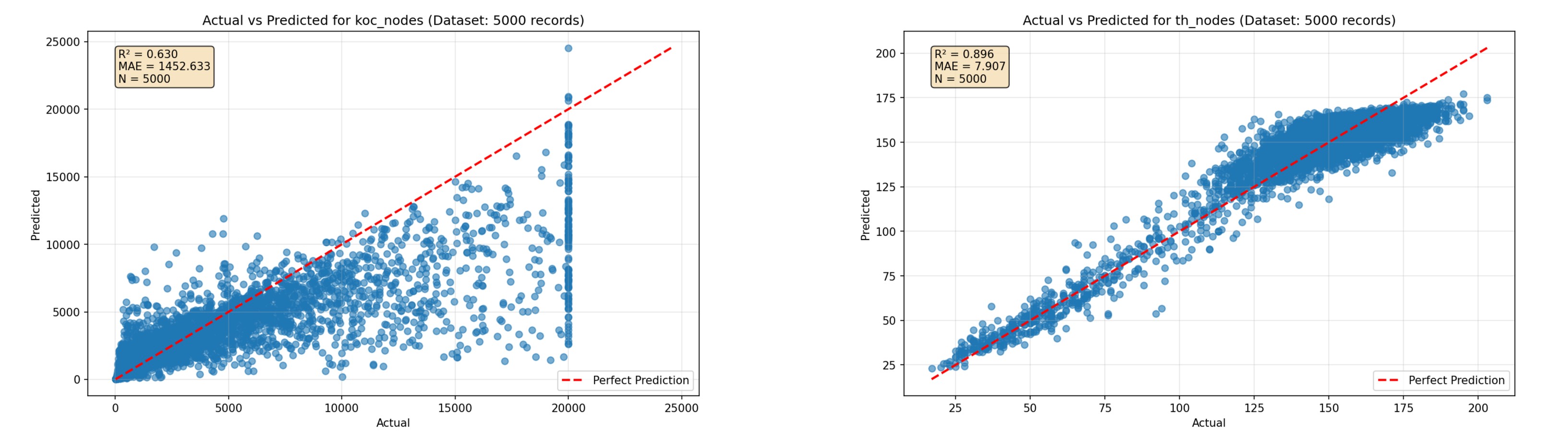
We trained Random Forest models on 25 static cube features (e.g., `solution_score`, `orientation_entropy`) to predict solver behavior ($N = 5000$).

Target	R^2 (Held-out)	CV R^2	Insight
Thistlethwaite Moves	0.998	0.997	Highly Deterministic
Kociemba Moves	0.998	0.997	Heuristics align with reality
Kociemba Nodes	0.630	0.661	Hard to predict (IDA*)
Kociemba Time	0.240	0.266	Implementation overhead

Adaptive Classification (Solver Selection): The system decides: *"Should we use Thistlethwaite to avoid Kociemba's worst-case?"*

- **Accuracy:** 94.2% (Held-out test split).
- **Feature Importance:** `solution_score` and `solution_variance` are the dominant predictors.

6. Prediction Accuracy Analysis Across Metrics



(a) Confusion matrix (raw counts) for 'avoid$_k$ociemba'.

(b) Confusion matrix (normalized by true class) for 'avoid$_k$ociemba'.

Figure: Actual vs. Predicted analysis for node expansion. The model for `Thistlethwaite` (right) shows high linearity due to deterministic phases, while `Kociemba` (left) shows higher variance ($R^2 = 0.630$) due to IDA* heuristic sensitivity.

7. Conclusion & Future Work

Conclusion:

- Established a clear dichotomy: Thistlethwaite for **predictability**, Kociemba for **optimality**.
- Demonstrated that ML can accurately predict solution length ($R^2 \approx 0.99$) but resource usage is inherently chaotic for IDA*.
- The hybrid system effectively selects the optimal solver based on constraints.

Future Work:

- Deep Learning (GNNs) to capture spatial dependencies better than static features.
- Online learning to adapt solver selection during deployment.

References

1. M. Thistlethwaite. Quarter-Turn Metric Rubik's Cube Solving Using Subgroup Reduction.
2. H. Kociemba. Two-Phase Algorithm for Solving the Rubik's Cube.
3. R. E. Korf. Finding Optimal Solutions to Rubik's Cube Using Pattern Databases. AAAI, 1997.
4. F. Agostinelli et al. Solving the Rubik's Cube with Deep Reinforcement Learning. Nature MI, 2019.