Università di Pisa

Symbolic AI vs. Large Language Models
A comparative study on Game Theory and Optimization

Luca Gallone
l.gallone@studenti.unipi.it

Lorenzo Blanco
l.blanco2@studenti.unipi.it

Goal

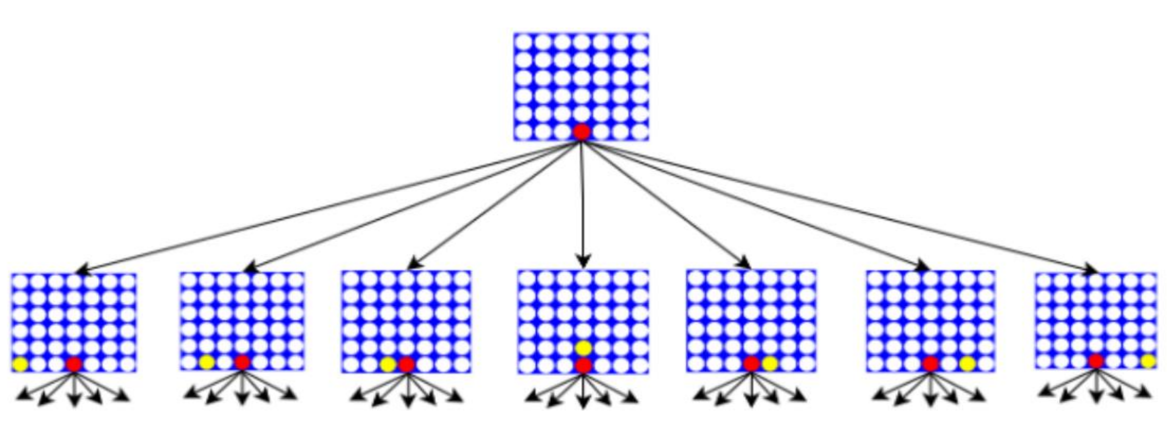
We aim to implement an intelligent agent designed to address two different types of problems: adversarial game strategy in Connect Four and combinatorial optimization in the Knapsack problem. First, we implement from scratch symbolic AI techniques, using different approaches for each problem. Finally, we compare their performance and reasoning capabilities against the latest Large Language Models.
Can the latest Large Language Models compete with traditional Symbolic AI techniques in logic reasoning tasks?

Connect Four

- Board of 6 rows by 7 columns
- Players alternate in turn and drop a token
- *Objective*: connect 4 tokens in any direction
- Deterministic, zero-sum and fully observable game

Minimax + α-β pruning + Depth Limited d + Heuristic

- *Optimal-play search*: recursively evaluates the game tree
- *Heuristic*: Positional Weighting + Pattern-based
- *Complexity*: exponential $O(7^{42}) \rightarrow O(7^{21})$ with α-β pruning



Monte Carlo Tree Search

- Simulations rather than exploring the entire state space
- Selection ($UCB1$) → Expansion → Rollout → Back-propagation
- *Best move*: child with the most visits (win-rate for tie-break)
- *Complexity*: linear in N (# rollouts) or t (time limit)

Results Connect Four

Minimax vs. Monte Carlo Tree Search

- 10 games, alternating which player moved first
- Same computational time: Minimax $d = 6$ (best time / move quality trade off) and MCTS $t = 10$

Metric	$t=1$	$t=5$	$t=10$	$t=30$	$t=60$	$t=180$	$t=240$
Minimax winrate (%)	100	100	100	100	100	100	100
avg num. plies	12.3	15.5	20.3	21.10	22.7	23.1	25

- Minimax with just $d = 6$ outperforms MCTS, even with higher t → the power of the heuristic vs. simulations!
- Time required for Monte Carlo Tree Search easily becomes unfeasible for a real game.

Minimax vs. Large Language Models

- 5 games against each LLM, alternating which player moved first

Metric	o4 mini high	Claude 3.7 thinking	Gemini 3 pro
Minimax winrate (%)	100	100	80
avg move time (s)	74.41	74,86	104.46
avg move cost ($)	0.034	0,106	0.117
avg total cost ($)	0.30	1.04	1.70
avg num. plies	18.3	16.6	25
invalid moves	0	0	0

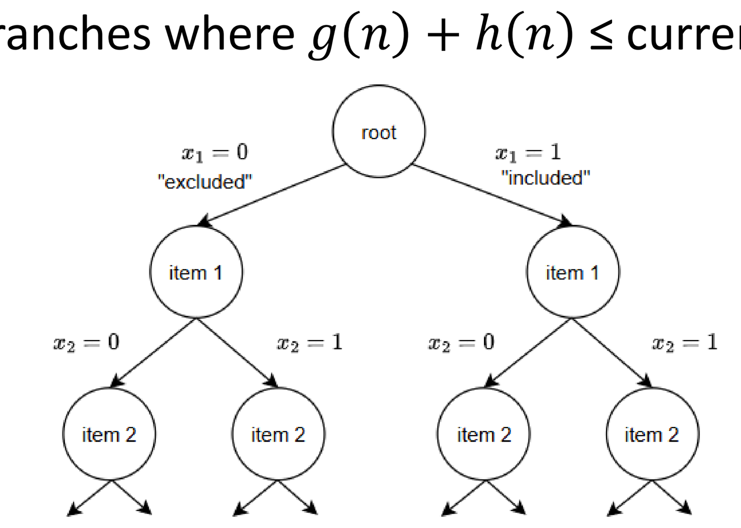
- All LLMs, despite requiring a lot of reasoning time, still lose out to minimax, which reasons in avg. 10s
- Only **Gemini 3 Pro** was able to win a game (close game with 38 plies!), being also the LLM that reasoned the longest and the most expensive.

Knapsack Problem

- *Given*: n items with (weight w_i, value v_i), and max capacity W
- *Objective*: maximize total value $\Sigma_i^n v_i x_i$, where $x_i \in \{0,1\}$
- *Constraint*: $\Sigma_i^n w_i x_i \leq W$
- *Type*: Combinatorial Optimization, NP-Hard
- *Complexity*: 2^n possible combinations of items

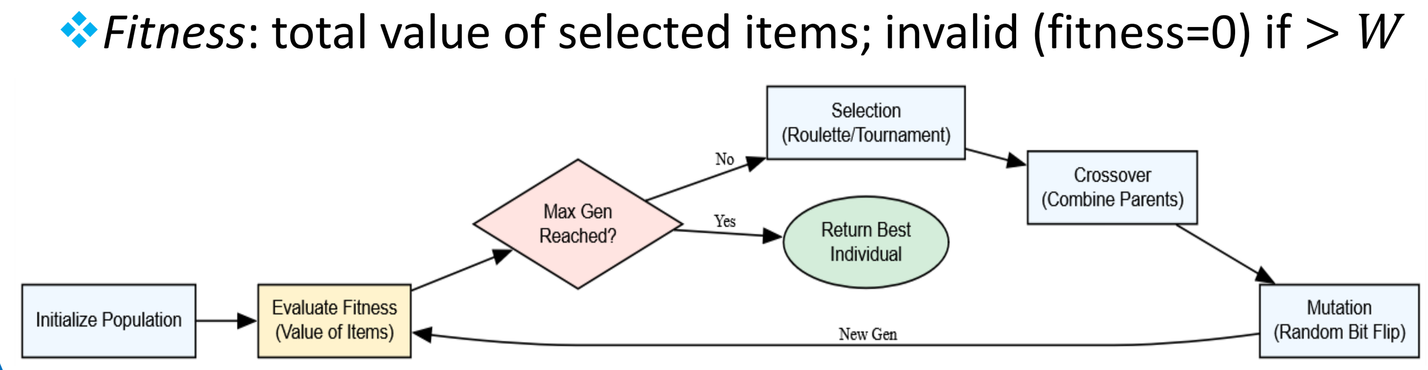
A* (Branch & Bound)

- Explores the decision tree to find the global optimum
- *Heuristic* $h(n)$: Fractional Knapsack
  (Relaxed problem → allows "cheating" with fractions)
- *Pruning*: branches where $g(n) + h(n) \leq$ current best found



Genetic Algorithm

- Population-based stochastic optimization
- *Encoding*: Chromosome string of n bits [1,0,0,1,…]
- *Fitness*: total value of selected items; invalid (fitness=0) if $> W$



Results Knapsack Problem

- Data: 500 items, 7117 optimal target, 2517 max capacity
- Results avg. on 5 runs:

Metric	o4 mini high	Claude 3.7 thinking	Gemini 3 pro	Genetic Alg.	A*
avg solution	7077	5767	7117	6694.4	7117
avg time (s)	131.29	248.2	157.94	19.49	7.22
avg cost ($)	0.08	0.34	0.22	-	-
invalid solutions	0	1	0	0	0

- **A*** is the winner, finding the global optimum very efficiently
- **Genetic Algorithm** achieved competitive results despite being a generic meta-heuristic, demonstrating that problem-agnostic approaches can efficiently approximate solutions without the domain-specific tailoring required by A*
- **o4-mini high** achieved near-optimal results (99.4% accuracy) and was the fastest among LLMs
- **Claude 3.7 thinking** struggled with strict constraints, resulting in invalid solutions and the lowest avg. solution despite the longest computation time
- **Gemini 3 Pro** successfully reached the optimal solution, demonstrating strong reasoning, but was ~22× slower than the symbolic approaches

Conclusion

- Traditional **symbolic AI** techniques continue to have a major impact in **well structured domains**, where explicit state representation and systematic exploration of the solution space enable high performance, short timescales, and fully deterministic behaviour.
- Although **LLMs** still underperform vs. symbolic AI techniques in strict decision-making tasks, the rapid progress of **new models** such as Gemini 3 Pro, which was release one week ago, suggests that this gap could potentially be reduced in the future as the technology continues to advance.