PRIME: **P**lanning with **R**eflective **I**terative **M**ulti-Agentic **E**xploration

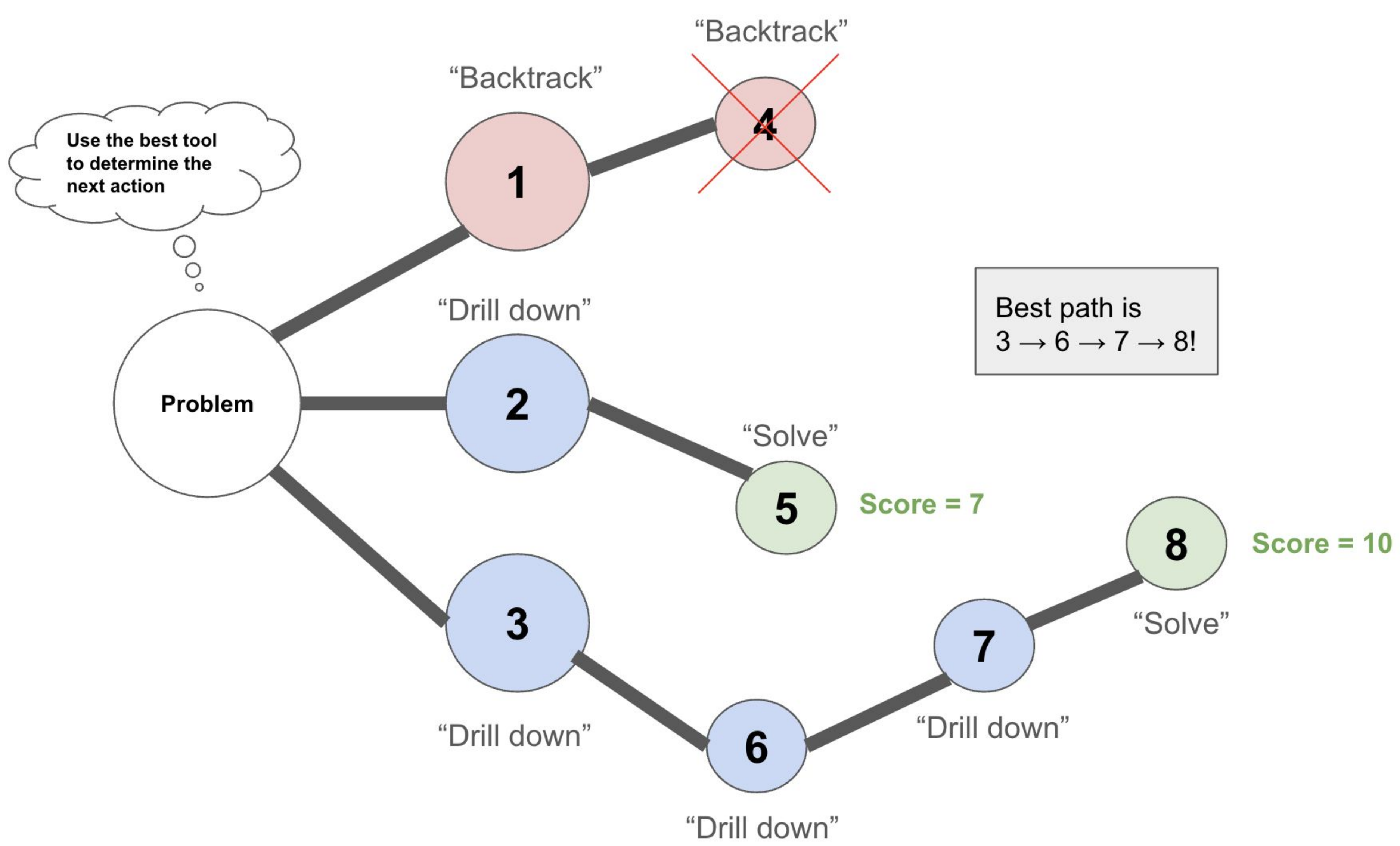
Chelsea Zou, Jui Khankari, Miriam Cheng

Problem

Top-down planning remains unsolved, and existing reasoning techniques are applied in a bottom up ad-hoc manner, lacking a structured framework. We propose PRIME, an MCTS-based algorithm that decomposes tasks, selects optimal reflective reasoning strategies, and dynamically executes specialized subagents

- PRIME integrates the best **self-reflection** technique at each decision step, allowing for **adaptive reasoning** in multi-step tasks
- PRIME systematically learns how to **decompose** complex reasoning problems into **subgoals** and uses self-reflection techniques to choose the best action at each step
- The framework uses an **MCTS**-based approach to generate the best plan, where self-reflective reasoning techniques dynamically generate the next best approach for each subgoal

Methods



Use self-reflective agent to assess which action to take at each node:

- **Drill Down**: Further refines broad subgoals into more specific tasks
- **Solve**: Selects the best reflective agent to directly solve the final subgoal
- **Backtrack**: Discards ineffective subgoals and explores alternative paths

Analysis

Planbench

- Increasing the number of nodes boosts performance, computational costs (for both LATS and PRIME)
- PRIME outperformed LATS because it combines planning with search (top-down subgoal generation with tree search)
- PRIME struggled when the usefulness of individual actions was unclear -> repetitive, undirected tree exploration
- When action outputs are clear, PRIME successfully generates subgoals, plans

Webshop

- PRIME and LATS had similar performance
- Subgoal generation is less beneficial in complex multi-input environments like online shopping
- Subgoals were generic and unhelpful, causing the agent to become stuck in local minima

Game of 24

- PRIME outperformed LATS by dynamically selecting reasoning tools tailored to each puzzle's complexity

Background

PRIME builds upon these works by **integrating MCTS with structured recursive planning**, dynamically selecting different reasoning strategies (e.g., self-reflection, debate, self-refinement) to enable more adaptive problem-solving

- **Option Discovery for Efficient Planning**: Wan & Sutton (2022) introduced **option discovery** in RL to optimize planning efficiency by selecting better subsets of actions at each step, reducing search complexity. PRIME adopts this heuristic to identify the best reflective tools for each subgoal
- **Chain-of-Thought (CoT) Reasoning**: Wei et al. (2022) proposed **CoT prompting**, improving LLMs' reasoning by generating intermediate steps before final answers, enhancing performance in logic and mathematical reasoning
- **ReAct Framework**: Yao et al. (2023) combined **CoT with real-world interactions** to iteratively refine decision-making. However, it lacked structured search mechanisms, limiting adaptability in complex tasks
- **Language Agent Tree Search (LATS)**: Zhou et al. (2024) introduced **MCTS with self-reflection**, enabling LLMs to explore multiple reasoning paths. However, it relied on heuristics and did not explicitly decompose problems into reusable subtasks

Experiments

- **Models:** PRIME, LATS, GPT 4o-mini, GPT-o1, PRIME with an upgraded planner component, and PRIME with an upgraded execution component were tested on 60 randomly selected questions
- **Benchmarks:**
  1. **Game of 24** (mathematical reasoning): math questions to construct 24 using 4 random numbers
  2. **Webshop** (real-world decision-making): navigating online store with 1 million products
  3. **Planbench** (sequential decision-making): real-world questions that require high-level planning
- Counterfactual evaluation was used to test whether structured planning genuinely improves reasoning rather than serving as post-hoc justification.
- A reverse-ablation study was conducted by selectively upgrading PRIME's planner and execution components to GPT-o1 to identify performance bottlenecks.
- Despite being 10x smaller, PRIME sometimes matched GPT-o1's performance while maintaining structured planning
  - PRIME required more API calls but resulted in an estimated 4x reduction in cost

Method	PlanBench	WebShop	Game of 24
PRIME	36.2%	40%	75%
LATS	2.2%	38%	44%
GPT-4o-mini	0%	0%	0%
GPT-o1	100%	100%	100%
PRIME upgraded planner (4o)	95%	94%	100%
PRIME upgraded execution (4o)	96%	92%	100%

Conclusion & Future Work

Our planner outperforms current state-of-the-art (LATS) and introduce a structured method for applying reasoning frameworks, replacing ad-hoc approaches. Future works include:

- **Clustering Problems:** Grouping similar problems to optimize tool selection and improve efficiency.
- **Enhanced Self-Reflection:** Expanding reasoning tools for better goal decomposition and decision-making.
- **Adaptive Value Function:** Dynamically refining evaluation criteria for improved prioritization and planning.