Policy-Guided Search on Tree-of-Thoughts for Efficient Problem Solving with Bounded Language Model Queries (Extended Abstract)

Sumedh Pendurkar[1], **Guni Sharon**[1]

[1]Department of Computer Science & Engineering, Texas A&M University
{sumedhpendurkar, guni}@tamu.edu

Abstract

Recent studies explored integrating search algorithms with Language Models (LMs) to perform look-ahead on the "Tree-of-Thoughts" (ToT), improving performance on problem-solving tasks. However, these search algorithms overlook the computational costs of LM inference under constrained budgets. We adapt Levin Tree Search (LTS) (Orseau et al. 2018) to ToT (Yao et al. 2023), leveraging LM probabilities as a policy to guide tree exploration without additional evaluation queries. We extend theoretical results of LTS by showing that, for ToT (a pruned tree), LTS guarantees a bound on states expanded and thoughts generated. Empirical evaluation under fixed LM query budgets demonstrates that LTS consistently achieves comparable or higher accuracy than baseline search algorithms across three domains (Blocksworld, PrOntoQA, Array Sorting) and four distinct LMs.

Code — https://github.com/sumedhpendurkar/Search-LLM-inference

Extended version — https://openreview.net/pdf?id=Rlk1bWe2ii

Introduction

Language Models (LMs) show strong performance on natural language based tasks, yet complex reasoning remains challenging. Chain-of-Thought (CoT) (Wei et al. 2022) prompting improves reasoning by eliciting intermediate steps. Tree-of-Thoughts (ToT) (Yao et al. 2023) generalizes this by generating search trees over *thoughts* (coherent token sequences), exploring multiple paths using search algorithms such as depth-first search (DFS) or beam search (BS). RAP (Hao et al. 2023) uses Monte-Carlo Tree Search for the tree search.

These methods require repeated LM queries for generating and evaluating thoughts, which is expensive (Snell et al. 2024). We focus on improving performance under strictly constrained computational budgets. Our key observation: evaluating thoughts requires additional LM queries that can be avoided by using the LM itself as a policy to guide search. We adapt *Levin Tree Search (LTS)* (Orseau et al. 2018) to ToT, extend its theoretical expansion bound to pruned trees, and demonstrate consistent improvements over DFS and beam search across three domains and four LMs.

Background

Tree-of-Thoughts. Given LM p and input $\mathbf{x}$, a thought $\mathbf{t}$ has probability $p(\mathbf{t}|\mathbf{x}) = \prod_i p(t_i|t_{<i}, \mathbf{x})$. ToT (Yao et al. 2023) generates a tree $\mathcal{T}$ where nodes are states (combination of partial thought sequences and input $\mathbf{x}$) and edges are thoughts. Terminal states H end with end-of-sequence tokens; goal states $Z \subseteq H$ are a subset of terminal nodes. The objective of search algorithms is to find a start-to-goal path.

Levin Tree Search. LTS (Orseau et al. 2018) uses a policy π to prioritize states by $cost(s) = g(s)/\pi(s)$, where $g(s)$ is depth and $\pi(s)$ is cumulative path probability. LTS expands in best-first order with expansion bound $\min_{s\in H} g(s)/\pi(s)$.

Initial-Commitment Problem of DFS. DFS commits greedily to one subtree. If the goal lies elsewhere, DFS exhausts the committed subtree before backtracking, because it favors confidence at the current depth over global comparison. LTS compares costs globally, switching subtrees earlier. This manifests empirically: DFS reaches terminal states quickly but with lower accuracy, i.e., does not reach goal nodes.

LTS on Tree-of-Thoughts

We use LM p as the LTS policy π. Adaptations: (1) sample $b_{\max}$ thoughts per expansion due to high vocabulary (Yao et al. 2023; Pendurkar et al. 2024); (2) remove duplicate thoughts (Hao et al. 2023); (3) exclude state-cuts (ToT is cycle-free). Since sampling prunes states, the original bound requires modification:

Proposition 1. *Given tree $\mathcal{T}$ with terminal states H, sampled subtree T, and $H' \subseteq H$ as terminal states in T, LTS ensures $N(T, H') \leq \min_{s\in H'} g(s)/\pi(s)$.*

Corollary 1. *Thoughts generated: $M(T, H') \leq b_{max} \cdot \min_{s\in H'} g(s)/\pi(s)$.*

Temperature Sensitivity. For a sufficiently accurate LM, the bound's sensitivity satisfies $\partial M/\partial\tau \leq b_{\max} \cdot g(s_g)^2/\pi(s_g)^2 \cdot k\Delta^+/\tau^2$, implying higher τ increases required queries, with sensitivity inversely proportional to τ^2.

Detailed analysis and proofs are available in the full version of this work (Pendurkar and Sharon 2025).

Task	Model (Size)	LTS	DFS
BW (step 2)	Llama 3.2 (3B)	15.5	11.1
BW (step 4)	Llama 3.2 (3B)	14.3	9.5
BW (step 2)	Llama 3.1 (8B)	33.3	20.0
BW (step 4)	Llama 3.1 (8B)	28.6	19.0
BW (step 6)	Llama 3.1 (8B)	17.8	9.2
BW (step 8)	Llama 3.1 (8B)	10.6	4.6
BW (step 2)	Qwen 2 (7B)	31.1	26.7
BW (step 6)	Qwen 2 (7B)	6.6	2.0
Sort	Llama 3.2 (3B)	20.0	15.0
Sort	Llama 3.1 (8B)	47.0	47.0
Sort	Qwen 2 (7B)	26.0	22.0
PrOntoQA	Llama 3.2 (3B)	27.8	18.8
PrOntoQA	Llama 3.1 (8B)	70.4	49.0
PrOntoQA	Qwen 2 (7B)	21.6	12.8

Table 1: Accuracy (%) of LTS vs. DFS. Budget tuned for DFS to reach terminal states on all instances.

Experiments

Setup. Built on Hao et al. (2023). Temperature $\tau = 0.8$, top-k=50, $b_{\max}$=3. Budget counts thoughts twice for DFS/BS (evaluation queries) and once for LTS. We use Llama 3.2 Instruct (1B, 3B), Llama 3.1 Instruct (8B), and Qwen 2 (7B).

Domains. *Blocksworld* (Valmeekam et al. 2022; Pendurkar et al. 2022, 2023): block arrangement validated by PDDL planner (depths 2–12). *PrOntoQA* (Saparov and He 2023): multi-hop reasoning (500 instances). *Array Sorting*: pairwise-swap sorting (100 instances).

Results. Table 1 shows LTS matches or outperforms DFS in all settings, with the largest gain of 21.4% on PrOntoQA (Llama 8B). LTS avoids initial-commitment and eliminates evaluation queries. Beam search performs well on easy instances but degrades on harder problems due to sample inefficiency. With increasing budgets, LTS consistently outperforms DFS across both low and high budget regimes. The improvement in low budget regimes stems from two factors: avoiding initial-commitment problem and requiring fewer LM queries per expansion. In high budget regimes, the advantage persists because DFS may still commit to incorrect subtrees regardless of available budget.

Temperature Sensitivity. Evaluating LTS with varying cost-function temperatures on BW step 4 reveals three regimes: low τ ($\approx$0.01) mimics DFS with early commitment and accuracy plateau; intermediate τ (0.5–1.5) achieves the best exploration-exploitation balance and highest final accuracy; high τ ($\approx$2.0) approaches BFS-like uniform exploration with poor budget efficiency. This confirms the theoretical analysis: moderate temperatures allow LTS to interpolate between greedy and balanced exploration of ToT.

Conclusion

We adapted Levin Tree Search to Tree-of-Thoughts, using LM probabilities as a search policy to eliminate evaluation queries. LTS provides theoretical bounds on the number of thought generated for pruned trees and empirically achieves comparable or superior accuracy to DFS and beam search across three domains and four LMs under constrained budgets, making it well-suited for latency-critical and resource-constrained applications.

Acknowledgments

The research was conducted in the PiStar AI and Optimization Lab at Texas A&M University. PiStar is supported in part by the NSF (IIS-2238979).

References

Hao, S.; Gu, Y.; Ma, H.; Hong, J.; Wang, Z.; Wang, D.; and Hu, Z. 2023. Reasoning with Language Model is Planning with World Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 8154–8173.

Orseau, L.; Lelis, L.; Lattimore, T.; and Weber, T. 2018. Single-agent policy tree search with guarantees. *Advances in Neural Information Processing Systems*, 31.

Pendurkar, S.; Huang, T.; Juba, B.; Zhang, J.; Koenig, S.; and Sharon, G. 2023. The (Un) Scalability of Informed Heuristic Function Estimation in NP-Hard Search Problems. *Transactions on Machine Learning Research*.

Pendurkar, S.; Huang, T.; Koenig, S.; and Sharon, G. 2022. A discussion on the scalability of heuristic approximators. In *Proceedings of the International Symposium on Combinatorial Search*, volume 15, 311–313.

Pendurkar, S.; Lelis, L. H.; Sturtevant, N. R.; and Sharon, G. 2024. Curriculum Generation for Learning Guiding Functions in State-Space Search Algorithms. In *Proceedings of the International Symposium on Combinatorial Search*, volume 17, 91–99.

Pendurkar, S.; and Sharon, G. 2025. Policy-Guided Search on Tree-of-Thoughts for Efficient Problem Solving with Bounded Language Model Queries. *Transactions on Machine Learning Research*.

Saparov, A.; and He, H. 2023. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. In *The Eleventh International Conference on Learning Representations*.

Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Valmeekam, K.; Olmo, A.; Sreedharan, S.; and Kambhampati, S. 2022. Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.