

Literature Review and Suggestions for AAAI-26 Submission

1. Introduction

This document provides a literature review of recent and relevant papers from top-tier AI conferences (AAAI, NeurIPS, ICML, and IJCAI) concerning optimal search trees, dynamic programming, and pruning techniques. It also offers suggestions for the user's paper, "A Sub-Cubic Algorithm for Constructing Optimal Two-Way Comparison Search Trees under Refined Heaviest-Key Pruning," to enhance its eligibility for publication in the AAAI-26 conference.

2. Analysis of the User's Paper

The user's paper, "A Sub-Cubic Algorithm for Constructing Optimal Two-Way Comparison Search Trees under Refined Heaviest-Key Pruning," presents a novel $O(n^3 \log n)$ -time algorithm for computing optimal two-way comparison search trees. This significantly improves upon the previous best $O(n^4)$ result by Chrobak et al. (2022). The core contributions include the introduction of the Refined Dynamic Programming Recurrence (RDPR) framework, which utilizes segment pruning and balanced partitioning to avoid redundant state exploration. The paper also extends the Refined Most-Likely-Key (RMLK) property through a generalized heaviest-key pruning strategy. The algorithm is applicable to both full and successful-queries variants, marking the first sub-quartic solution for this problem. Experimental evaluations show 2-5x speedups on realistic datasets while maintaining theoretical optimality guarantees.

Key aspects of the paper:

- **Problem:** Constructing optimal two-way comparison search trees.
- **Previous State-of-the-Art:** $O(n^4)$ algorithm by Chrobak et al. (2022) and $O(n^4)$ by Anderson et al. (2002).
- **Proposed Solution:** $O(n^3 \log n)$ -time algorithm.

- **Methodology:** Refined Dynamic Programming Recurrence (RDPR) framework, segment pruning, balanced partitioning, generalized heaviest-key pruning.
- **Impact:** First sub-quartic solution, 2-5x speedup on realistic datasets.

3. Literature Review

3.1. Foundational Work

The problem of constructing optimal search trees has a long history in computer science. Knuth's seminal $O(n^2)$ algorithm (1971) addressed optimal three-way comparison search trees. However, two-way comparison variants, which are more relevant to modern programming languages, have proven more challenging due to the

increased algorithmic complexity arising from "intervals with holes."

Anderson et al. (2002) [1] provided the first polynomial-time algorithm for optimal two-way comparison search trees, achieving $O(n^4)$ time complexity for the successful-queries variant. Their work introduced the Most-Likely-Key (MLK) property, which restricts optimal solutions to trees where equality tests target the heaviest available key.

Chrobak et al. (2021) [2] significantly simplified this approach and extended it to the full variant (allowing unsuccessful queries). They introduced the Refined Most-Likely-Key (RMLK) property, providing a deterministic tie-breaking mechanism. Their $O(n^4)$ algorithm represented the state-of-the-art prior to the user's work.

3.2. Recent Advances in Optimal Decision Trees and Pruning

While direct recent work on optimal *two-way comparison search trees* is limited, the broader field of *optimal decision trees* and *pruning techniques* has seen significant activity in top-tier AI conferences. Many of these papers leverage dynamic programming and branch-and-bound approaches, which are highly relevant to the user's work.

Recent AAAI papers in this area include:

- **Optimal Classification Trees for Continuous Feature Data Using Dynamic Programming with Branch-and-Bound** (AAAI 2025) [3]: This paper proposes a novel

algorithm that optimizes trees directly on continuous feature data using dynamic programming with branch-and-bound. This is highly relevant as it demonstrates recent advancements in applying DP and branch-and-bound for optimal tree construction, albeit for classification trees rather than search trees.

- **Separator-Based Pruned Dynamic Programming for Steiner Tree** (AAAI 2019) [4]: This work presents a novel separator-based pruning technique to speed up a theoretically fast dynamic programming algorithm for the Steiner tree problem. This highlights the ongoing research into effective pruning strategies within DP frameworks, which is central to the user's

paper.

- **Memory-Efficient Dynamic Programming for Learning Optimal Decision Trees** (AAAI 2020) [5]: This paper focuses on improving the memory efficiency of dynamic programming approaches for learning optimal decision trees. This is relevant as it addresses practical challenges in implementing DP algorithms for tree optimization.

Recent NeurIPS papers in this area include:

- **Fair and Optimal Decision Trees: A Dynamic Programming Approach** (NeurIPS 2022) [6]: This paper demonstrates the effectiveness of dynamic programming for optimizing decision trees, particularly in the context of fairness. It shows how DP can exploit tree structure for efficient optimization.
- **Necessary and Sufficient Conditions for Optimal Decision Trees** (NeurIPS 2023) [7]: This work explores the theoretical underpinnings of optimal decision trees, providing conditions for their construction using dynamic programming. This contributes to the fundamental understanding of optimal tree algorithms.
- **Randomized Pruning: Efficiently Calculating Expectations in Large Dynamic Programs** (NeurIPS 2016) [8]: While older, this paper is relevant for its focus on randomized pruning techniques to accelerate dynamic programming, a core concept in the user's paper.

Recent ICML papers in this area include:

- **Optimal Decision Tree Pruning Revisited: Algorithms and Complexity** (ICML 2025) [9]: This very recent paper provides a comprehensive theoretical analysis of the complexity of different pruning strategies for decision trees. This is directly relevant to the user's work on refined heaviest-key pruning.
- **Near-Optimal Decision Trees in a SPLIT Second** (ICML 2025) [10]: This paper discusses dynamic programming approaches for building trees recursively and efficiently. It highlights the ongoing research into fast and near-optimal tree construction.
- **Born-Again Tree Ensembles** (ICML 2020) [11]: This paper proposes an exact algorithm to transform a tree ensemble into an optimal decision tree, focusing on size optimization. This showcases advanced techniques in tree optimization.

Recent IJCAI papers in this area include:

- **Comparing Best-First Search and Dynamic Programming for Optimal Sequence Alignment** (IJCAI 2003) [12]: While not directly on search trees, this paper compares dynamic programming with best-first search for optimal alignment, demonstrating the applicability and efficiency of DP in related combinatorial optimization problems.
- *Learning Optimal Bayesian Networks Using A Search** (IJCAI 2011) [13]: This paper formulates learning optimal Bayesian networks as a shortest path problem and uses an A* search algorithm. This shows the use of search algorithms combined with optimality criteria.
- **Faster Optimal Coalition Structure Generation via Offline Dynamic Programming** (IJCAI 2024) [14]: This recent paper combines dynamic programming with graph search for faster optimal structure generation, demonstrating advanced DP applications.

3.3. Specific Mentions of Two-Way Comparison Search Trees

Beyond the foundational work, recent direct mentions of two-way comparison search trees are less frequent in these top-tier conferences, suggesting that the user's work addresses a niche but important area. The most relevant recent work is still primarily from Chrobak et al. and related authors, often published in more theoretical computer science venues or as preprints.

- **Structural Properties of Search Trees with 2-way Comparisons** (arXiv 2023) [15]: This preprint studies structural properties of two-way comparison search trees, indicating ongoing theoretical interest in the area.
- **A Tight Threshold Bound for Search Trees with 2-way Comparisons** (arXiv 2023, published in a book chapter in 2024) [16]: This work further explores the theoretical bounds for two-way comparison search trees.

These papers confirm that the problem of two-way comparison search trees remains an active area of theoretical research, even if less frequently appearing in the main tracks of general AI conferences compared to broader topics like optimal decision trees.

4. Suggestions for AAI-26 Conference Submission

To maximize the chances of publication at AAI-26, consider the following suggestions:

4.1. Emphasize AI Relevance and Impact

While your paper is strong in theoretical computer science and algorithms, AAI is a general AI conference. Clearly articulate the AI relevance and broader impact of your work.

Consider:

- **Applications in AI:** Explicitly discuss how optimal two-way comparison search trees can be applied in various AI domains. For example, in knowledge representation, expert systems, efficient data retrieval for AI systems, or even in optimizing decision-making processes where two-way comparisons are natural.
- **Connection to Machine Learning:** If applicable, draw connections to machine learning. For instance, could these optimized search trees be used as a component in interpretable AI models, or for efficient feature selection/data partitioning in certain learning algorithms? Even if not directly, consider how the algorithmic advancements could indirectly benefit ML.
- **Generalizability:** While your focus is on two-way comparisons, discuss how the underlying principles of RDPR, segment pruning, and generalized heaviest-key pruning could be adapted or inspire solutions for other complex optimization problems in AI.

4.2. Strengthen Experimental Evaluation and Reproducibility

AAAI places a strong emphasis on empirical validation and reproducibility. Your paper mentions 2-5x speedups, which is excellent. To further strengthen this:

- **Broader Benchmarking:** If possible, test your algorithm on a wider range of publicly available datasets, especially those that might be relevant to AI applications. Compare against more baselines if they exist, beyond just the $O(n^4)$ algorithms.
- **Detailed Experimental Setup:** Provide comprehensive details of your experimental setup, including hardware, software versions, and specific configurations used. This aids reproducibility.
- **Open-Source Implementation:** Strongly consider making your code publicly available. AAAI requires a reproducibility checklist, and providing open-source code is a significant plus. Mention this in your paper and provide a link (e.g., to a GitHub repository) in the supplementary material.
- **Ablation Studies:** If feasible, conduct ablation studies to demonstrate the individual contributions of RDPR, segment pruning, and generalized heaviest-key pruning to the overall performance improvement. This helps reviewers understand the novelty and effectiveness of each component.

4.3. Clarity and Presentation

- **Introduction and Motivation:** Ensure the introduction clearly states the problem, its significance in AI, your contributions, and how they advance the state-of-the-art. Make it accessible to a broader AI audience, not just theoretical computer scientists.
- **Related Work:** Expand the related work section to include more of the recent optimal decision tree and pruning papers from AAAI, NeurIPS, ICML, and IJCAI that were identified in this review. Clearly articulate how your work differs from and builds upon these, even if they are not directly on two-way comparison search trees. This demonstrates a comprehensive understanding of the broader landscape of tree optimization.
- **Technical Details:** While your paper is theoretically rigorous, ensure that the technical details are presented as clearly as possible. Use figures and diagrams to illustrate

complex concepts like RDPR, segment pruning, and heaviest-key pruning. A well-structured algorithm pseudocode can also be very helpful.

- **Page Limit:** Adhere strictly to the 7-page technical content limit, plus references. Use supplementary material for proofs, additional experimental results, and code.

4.4. Address Reviewer Concerns (Anticipated)

Reviewers at general AI conferences might have questions regarding:

- **Practicality vs. Theory:** While your work is theoretically significant, be prepared to discuss its practical implications and potential real-world applications in AI systems. The experimental speedups are a good start, but elaborate on *why* this speedup matters in an AI context.
- **Comparison to Heuristics:** Briefly discuss why an optimal algorithm is preferred over heuristic approaches for this problem, especially if heuristics exist that offer faster (though sub-optimal) solutions. Emphasize scenarios where optimality is crucial.
- **Novelty in AI Context:** Reiterate the novelty of your approach within the broader AI landscape, especially concerning the sub-cubic complexity for two-way comparison search trees.

5. AAI-26 Submission Checklist

Based on the AAI-26 Main Technical Track Call for Papers, ensure you meet the following:

- **Abstract Deadline:** July 25, 2025 (UTC-12)
- **Full Paper Deadline:** August 1, 2025 (UTC-12)
- **Supplementary Material Deadline:** August 4, 2025 (UTC-12)
- **Page Limit:** Up to 7 pages for technical content + unlimited pages for references.
- **Formatting:** Use the AAI-26 Author Kit (LaTeX or Word templates will be provided by AAI Press).
- **Reproducibility Checklist:** Complete the required checklist.

- **Authorship:** Ensure all authors meet AAAI authorship criteria. No AI system can be listed as an author.
- **Generative AI Usage:** If generative AI tools are used for text editing, ensure full responsibility for content and compliance with AAAI ethics policies. Do not present LLM-generated text as experimental analysis unless it is part of the analysis itself. Do not cite LLMs as sources.

6. References

- [1] Anderson, R., Kannan, S., Karloff, H., & Ladner, R. E. (2002). Thresholds and optimal binary comparison search trees. *Journal of Algorithms*, 44(2), 338-358.
<https://www.sciencedirect.com/science/article/abs/pii/S0196677402002031>
- [2] Chrobak, M., Golin, M., Munro, J. I., & Young, N. E. (2021). A Simple Algorithm for Optimal Search Trees with Two-way Comparisons. *ACM Transactions on Algorithms*, 18(1), 1-11.
<https://dl.acm.org/doi/10.1145/3477910>
- [3] Optimal Classification Trees for Continuous Feature Data Using Dynamic Programming with Branch-and-Bound. (2025). *AAAI Conference on Artificial Intelligence*.
<https://ojs.aaai.org/index.php/AAAI/article/view/33210>
- [4] Separator-Based Pruned Dynamic Programming for Steiner Tree. (2019). *AAAI Conference on Artificial Intelligence*. <https://ojs.aaai.org/index.php/AAAI/article/view/3965>
- [5] Memory-Efficient Dynamic Programming for Learning Optimal Decision Trees. (2020). *AAAI Conference on Artificial Intelligence*. <https://cdn.aaai.org/ojs/8024/8024-13-11552-1-2-20201228.pdf>
- [6] Fair and Optimal Decision Trees: A Dynamic Programming Approach. (2022). *NeurIPS*.
https://proceedings.neurips.cc/paper_files/paper/2022/file/fe248e22b241ae5a9adf11493c8c12bc-Paper-Conference.pdf
- [7] Necessary and Sufficient Conditions for Optimal Decision Trees. (2023). *NeurIPS*.
https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5fce9627e15c84db572a66e029b1fc-Paper-Conference.pdf

- [8] Randomized Pruning: Efficiently Calculating Expectations in Large Dynamic Programs. (2016). *NeurIPS*. <http://papers.neurips.cc/paper/3710-randomized-pruning-efficiently-calculating-expectations-in-large-dynamic-programs.pdf>
- [9] Optimal Decision Tree Pruning Revisited: Algorithms and Complexity. (2025). *ICML*. <https://icml.cc/virtual/2025/poster/44353>
- [10] Near-Optimal Decision Trees in a SPLIT Second. (2025). *ICML*. <https://icml.cc/virtual/2025/poster/46182>
- [11] Born-Again Tree Ensembles. (2020). *ICML*. [https://icml.cc/media/Slides/icml/2020/virtual\(no-parent\)-16-13-00UTC-6682-born-again_tree.pdf](https://icml.cc/media/Slides/icml/2020/virtual(no-parent)-16-13-00UTC-6682-born-again_tree.pdf)
- [12] Comparing Best-First Search and Dynamic Programming for Optimal Sequence Alignment. (2003). *IJCAI*. <https://www.ijcai.org/Proceedings/03/Papers/177.pdf>
- [13] Learning Optimal Bayesian Networks Using A* Search. (2011). *IJCAI*. <https://www.ijcai.org/Proceedings/11/Papers/364.pdf>
- [14] Faster Optimal Coalition Structure Generation via Offline Dynamic Programming. (2024). *IJCAI*. <https://www.ijcai.org/proceedings/2024/0027.pdf>
- [15] Structural Properties of Search Trees with 2-way Comparisons. (2023). *arXiv*. <https://arxiv.org/abs/2311.02224>
- [16] A Tight Threshold Bound for Search Trees with 2-way Comparisons. (2024). In *Algorithms and Complexity* (pp. 119-131). Springer, Cham. https://link.springer.com/chapter/10.1007/978-981-97-2340-9_9