CSci 4511 Writing 2

Baanee Singh

February 2025

Adaptive A*

In the paper, "S. Koenig and M. Likhachev, Adaptive A*" in Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, we are introduced to the incremental version of the A* search algorithm called Adaptive A*. The A* search algorithm calculates the shortest path for each search problem. Because the algorithm computes paths from scratch for each search problem, it expands many nodes, consuming a significant amount of memory [1]. Since the traditional approach to A* has issues with spatial complexity, the authors introduce a variation, Adaptive A*, to improve efficiency. This is accomplished by updating heuristic values based on previous searches to make more informed searches [2]. With this iterative approach, the algorithm aims to improve efficiency and reduce computational costs when finding cost-minimal paths.

Adaptive A* enhances the A* search algorithm by iteratively updating heuristic values (h-values). The process starts with updating action costs if they have increased since the last run of the algorithm, ensuring all information is up-to-date and accurate. The algorithm then picks a start state and runs a forward A* search to find the least cost path, denoted g^*. After completing a search, the algorithm updates the h-value of each expanded state s within a *CLOSED* set with the following formula [2]:

$$h[s] = g* - g[s]$$

In this formula, g^* is the cost of the cost-minimal path, *g[s]* is the g-value of the state s after search, and *h[s]* is the h-value of the state s after search. Updating the h-value makes the heuristic more consistent and informed over time, leading to more informed and focused searches.

The Adaptive A* algorithm is well-suited for real-time pathfinding in dynamic and partially known environments, such as those found in real-time computer games like Total Annihilation and Warcraft [2]. It is effective when agents must repeatedly navigate an environment where the terrain information is unknown, but can be incrementally discovered and remembered. This makes it ideal for game intelligence, where paths may become blocked and the agent must adapt without re-expanding unnecessary paths,which would happen in the traditional A* search algorithm.

The authors provide a theoretical and experimental validation of the Adaptive A* algorithm. Theoretically, they state that the h-values of the same state are monotonically nondecreasing over time, which means that they never decrease as the algorithm progresses. Therefore, the algorithm becomes more informed. They also prove that the h values remain consistent, so the algorithm will continue to find cost-minimal paths over time. These proofs establish the algorithm as an optimal and efficient search method, demonstrating how leveraging previous searches improves future ones. Additionally, to supplement the theoretical proofs, the authors performed experiments on randomly generated four-connected mazes. Here, they ran this experiment on three different search algorithms, A* search algorithm, Adaptive A*, and D* lite. The key performance metrics are the number of expanded cells, reflecting the efficiency, and the total planning time until the agent reaches the goal, assessing the speed. The table below reflects the performance metrics.

Algorithm	Expanded Cells ($\mu \pm \sigma$)	Planning Time (μsec) ($\mu \pm \sigma$)
A* with Binary Heap	53,084.27 ± 1,229.42	16,514.75 ± 357.13
Adaptive A* with Binary Heap	41,593.55 ± 735.96	14,355.81 ± 262.40
D* Lite with Binary Heap	11,416.37 ± 116.94	8,505.72 ± 88.01
Adaptive A* with Buckets	41,063.69 ± 541.11	7,051.12 ± 96.86

Table 1: Experiments in Random Mazes [2]

The experimental results demonstrate that Adaptive A* is more efficient than the A* search algorithm by expanding fewer cells during the search while still maintaining optimality. However, it takes slightly longer due to heuristic updates, though it still achieves an overall speedup. While D Lite* performs better in the tested mazes, the authors suggest optimizations that could further improve Adaptive A*'s efficiency [2].

The authors provide a clear and focused explanation to the Adaptive A* algorithm. They highlight the issues with the traditional A* search algorithm and how those issues are solved with the improved algorithm. Providing theoretical and experimental validation furthers the effectiveness of the algorithm. However, the experimental results in Table 1 are not well explained or well structured, and adding graphical comparisons would enhance understanding. Furthermore, at times, the writing felt dense and difficult to follow. Explanations regarding how Adaptive A* updates its heuristics is somewhat unclear, requiring further examples and clarification. The theoretical proofs simply told us that it works, rather than providing a clear explanation of how it works. In general, the paper would be more effective with a clearer structure, a better context for its claims, and stronger explanations.

References

[1] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson series in artificial intelligence, Hoboken: Pearson, fourth edition ed., 2021.

[2] S. Koenig and M. Likhachev, "Adaptive A*," in *In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, (Utrecht, Netherlands), pp. 1311–1312, July 2005.

[3] ChatGPT, "ChatGPT - Latex Equation Formatting."

Note: Used ChatGPT for formatting LaTeX tables, symbols, and syntaxes [3]. Used Grammarly for grammer purposes and sentence restructuring.

Link: ChatGPT Conversation 1

Images from ChatGPT:

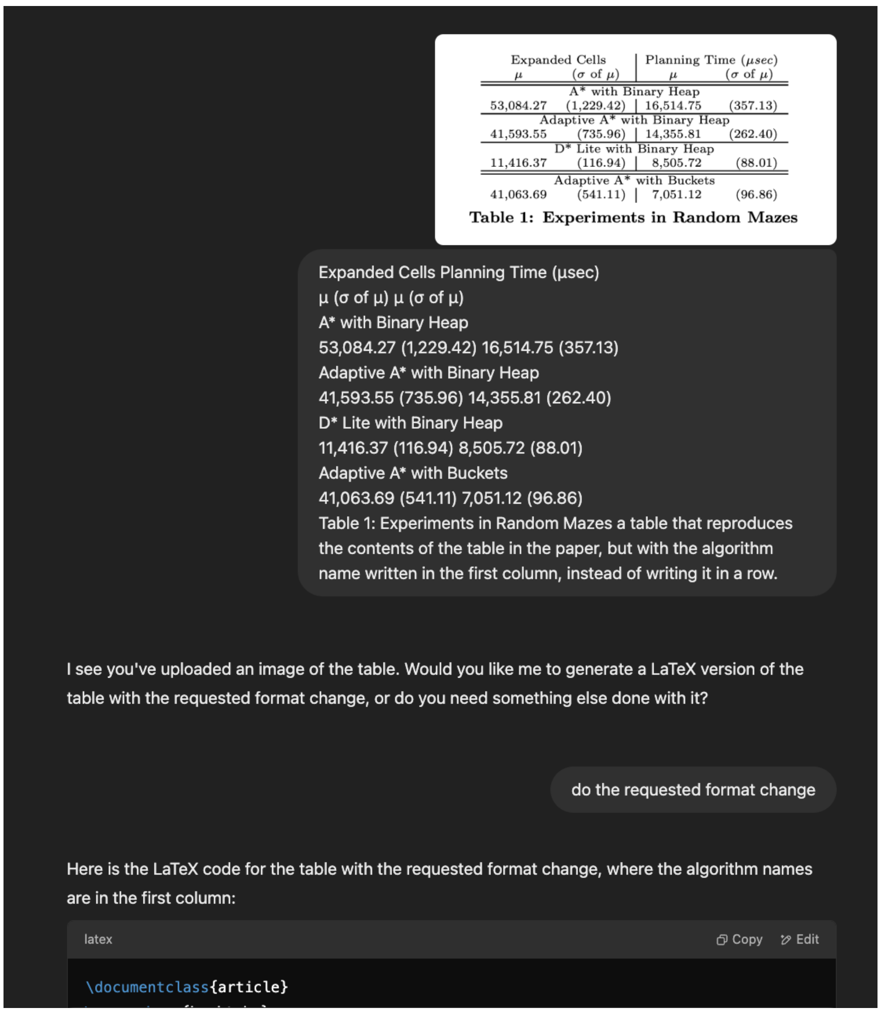
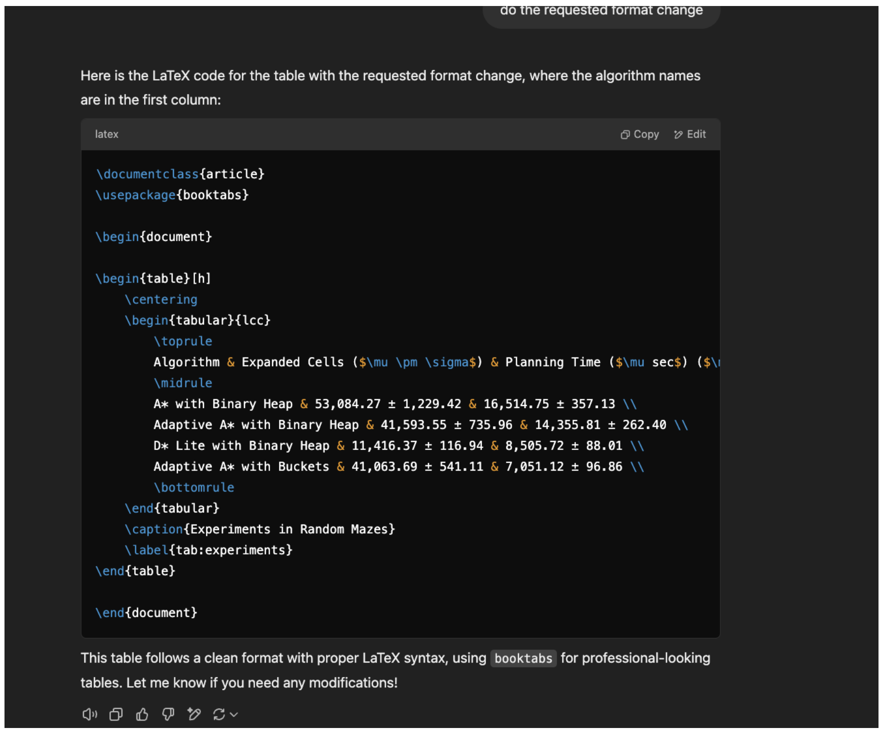


Figure 1: Chat Conversation.



do the requested format change

Here is the LaTeX code for the table with the requested format change, where the algorithm names are in the first column:

latex Copy Edit

```latex
\documentclass{article}
\usepackage{booktabs}

\begin{document}

\begin{table}[h]
    \centering
    \begin{tabular}{lcc}
        \toprule
        Algorithm & Expanded Cells ($\mu \pm \sigma$) & Planning Time ($\mu sec$) ($\
        \midrule
        A* with Binary Heap & 53,084.27 ± 1,229.42 & 16,514.75 ± 357.13 \\
        Adaptive A* with Binary Heap & 41,593.55 ± 735.96 & 14,355.81 ± 262.40 \\
        D* Lite with Binary Heap & 11,416.37 ± 116.94 & 8,505.72 ± 88.01 \\
        Adaptive A* with Buckets & 41,063.69 ± 541.11 & 7,051.12 ± 96.86 \\
        \bottomrule
    \end{tabular}
    \caption{Experiments in Random Mazes}
    \label{tab:experiments}
\end{table}

\end{document}
```

This table follows a clean format with proper LaTeX syntax, using `booktabs` for professional-looking tables. Let me know if you need any modifications!

Figure 2: Chat Conversation.