

VILNius gediminas technical university

faculty of fundamental sciences

department of information technologies

Elmira Sultanova

Using Data-Oriented Design (DOD) for Algorithm Optimization

Final Bachelor Thesis

Information Technologies Study Programme, State code 6121BX033

Finance specialization

Study Field of Informatics

Vilnius, 2025

**Table of Contents**

[1. Introduction 3](#_Toc186296988)

[1.1. Relevance 3](#_Toc186296989)

[1.2. Aim and Objectives 4](#_Toc186296990)

[1.3. Methodology Overview 4](#_Toc186296991)

[1.4. Practical Uses 4](#_Toc186296992)

[References 5](#_Toc186296993)

1. Introduction

As systems, users, and data grow, there is a need to optimize algorithms to increase efficiency and speed. Pathfinding algorithms are no exception, as they are crucial in fields such as game development, navigation systems, robotics, and artificial intelligence where efficiency is essential (Patras et al., 2021; Rafiq et al., 2020). One of the most important aspects to consider when optimizing an algorithm is data management. Data Oriented Design (DOP) addresses this issue by prioritizing data while programming, by allowing to effectively handle memory access, cache, and data (Dang, 2023).

A\* (A star) is one of the most used pathfinding algorithms. It is faster and more effective than other algorithms because it uses a heuristic function to 'predict' the path (GeeksforGeeks, 2024). Despite its effectiveness as maps become larger, it requires a large amount of processing power and memory (Foead et al., 2021). This research paper will focus on optimizing the A\* algorithm to display the use of DOP in algorithm optimization.

* 1. Relevance

In modern systems, efficient use of computational resources is very important as it has a direct influence on performance, energy usage, and scalability. Optimizing algorithms may help to reduce energy consumption, improve performance, speed, and enable scalability. However, it is difficult to apply optimization as it might compromise the accuracy of the algorithm. Additionally, the increased robustness of the algorithm can lead to a larger use of computational resources (Penev et al., 2024; Sacthesw, 2024). Henceforth, applying DOD principles might help to balance performance with the usage of memory by prioritizing the data handling and management.

The A\* algorithm, its use, and different techniques of optimization have been widely researched. Existing research focuses on improving the pathfinding, eliminating the number of nodes, smoothing the path, and improving heuristics by adding new factors (L. Liu et al., 2022; Tang et al., 2021; Sun & Li, 2016). These approaches are largely focused on introducing new and more effective functions, however, there is lack of focus on handling data and managing memory. Although DOD has been researched in different contexts, such as multi-threaded video games and optimizing deep networks, its uses for algorithm optimization remain unexplored (Shah et al., 2020; Wingqvist et al., 2022).

* 1. Aim and Objectives

The aim of this research is to analyse the DOD approaches in optimizing algorithms by applying these principles to the A\* algorithm and measuring its performance.

1. Implementing a map and A\* algorithm in C# with SFML (Simple and Fast Multimedia Library): Create a visual map with obstacles and the classical A\* algorithm that can navigate through the map
2. Identifying and applying DOD in C#: Analyzing the ways of data reconstruction and access based on DOD
3. Applying the identified principles to the A\* algorithm: Creating variants of the A\* algorithm with modifications according DOD
4. Comparing the performance of different A\* algorithms: using BenchmarkDotNet to compare the performance
5. Analysing the results: Comparing how each version of the A\* algorithm performed and evaluating the use of DOD in optimizing algorithms
   1. Methodology Overview

The methodology consists of implementation and benchmarking. Firstly, a classic A\* algorithm is implemented with visuals to demonstrate the path that was found. Second, DOD … principles are used to optimize the A\* algorithm, focusing of data handling, memory access, time and cache efficiency. BenchmarkDotNet will be used to evaluate different variants of the algorithm. The results will be compared to evaluate the effectiveness of DOD in optimizing algorithms.

* 1. Practical Uses

The methodologies and findings in this research have many practical uses in computer science and engineering. In game development, the A\* algorithm is used to create non-playable characters (NPC), optimizing the algorithm can increase the performance of video games with many NPCs (Candra et al., 2021). Furthermore, in robotics, the A\* algorithm is used to help autonomous robots move, the enhanced algorithm will enable them to navigate through dynamic environments more efficiently and allow further advancements in automation (Ducho et al., 2014). Furthermore, the findings of this research can be applied to improve navigation applications in which the A \* algorithm is used to find paths through the city (Veisi et al., 2023). Moreover, optimizations focusing on the data can be very useful in modern datasets that keep growing and complexity for systems where performance and efficient use of computational power are important.

References

Candra, A., Budiman, M. A., & Pohan, R. I. (2021). Application of A-Star algorithm on Pathfinding game. *Journal of Physics Conference Series*, *1898*(1), 012047. https://doi.org/10.1088/1742-6596/1898/1/012047

Dang, T. (2023, October 3). *Revolutionize your code: the magic of Data-oriented Design (DOD) programming*. Orient Software. https://www.orientsoftware.com/blog/dod-programming/

Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering*, *96*, 59–69. https://doi.org/10.1016/j.proeng.2014.12.098

Foead, D., Ghifari, A., Kusuma, M. B., Hanafiah, N., & Gunawan, E. (2021). A Systematic Literature review of A\* pathfinding. *Procedia Computer Science*, *179*, 507–514. https://doi.org/10.1016/j.procs.2021.01.034

GeeksforGeeks. (2024, July 30). *A\* search algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/a-search-algorithm/

Liu, L., Wang, B., & Xu, H. (2022). Research on Path-Planning algorithm integrating optimization A-Star Algorithm and Artificial Potential Field method. *Electronics*, *11*(22), 3660. https://doi.org/10.3390/electronics11223660

Penev, K., Gegov, A., Isiaq, O., & Jafari, R. (2024). Energy efficiency evaluation of artificial intelligence algorithms. *Electronics*, *13*(19), 3836. https://doi.org/10.3390/electronics13193836

Sacthesw, F. (2024, August 24). *Impact of computational resources on algorithm selection*. Future Machine Learning. https://futuremachinelearning.org/impact-of-computational-resources-on-algorithm-selection/

Shah, A., Wu, C., Mohan, J., Chidambaram, V., & Krähenbühl, P. (2020). Memory optimization for deep networks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2010.14501

Sun, D., & Li. (2016). Evaluation function optimization of A-Star algorithm in optimal path selection. *Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia*. https://doi.org/10.21311/001.39.4.14

Tang, G., Tang, C., Claramunt, C., Hu, X., & Zhou, P. (2021). Geometric A-Star Algorithm: An improved A-Star algorithm for AGV path planning in a port environment. *IEEE Access*, *9*, 59196–59210. https://doi.org/10.1109/access.2021.3070054

Veisi, O., Du, D., Moradi, M. A., Guasselli, F. C., Athanasoulias, S., Syed, H. A., Müller, C., & Stevens, G. (2023). *Designing SafeMap Based on City Infrastructure and Empirical Approach: Modified A-Star Algorithm for Earthquake Navigation Application*. https://doi.org/10.1145/3615900.3628788

Wingqvist, D., Wickstrom, F., & Memeti, S. (2022). *Evaluating the performance of object-oriented and data-oriented design with multi-threading in game development*. https://doi.org/10.1109/gem56474.2022.10017610