University of Puerto Rico
Department of Electrical and Computer Engineering
ICOM5015 Artificial Intelligence

CSP Algorithms

Group C

Manuel Alejandro Umaña Rodriguez -Undergraduate Computer Engineering
Dahyna Martínez Pérez -Undergraduate Computer Engineering
Jan Luis Pérez de Jesús -Undergraduate Computer Engineering
Christopher Hans Mayens Matías -Undergraduate Computer Engineering

For: Profesor José Fernando Vega Riveros
Date: May 5, 2025

Agenda

01 Purpose of experiment

02 Hypothesis

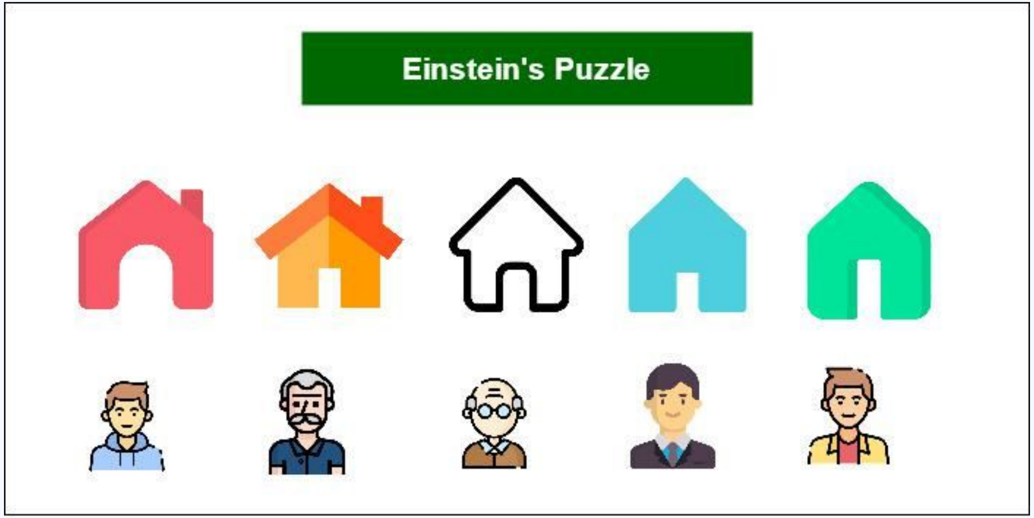
03 Concepts

04 Experiments set up

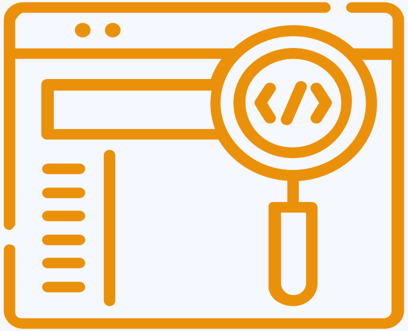
05 Information

06 Conclusion

07 Credits & References



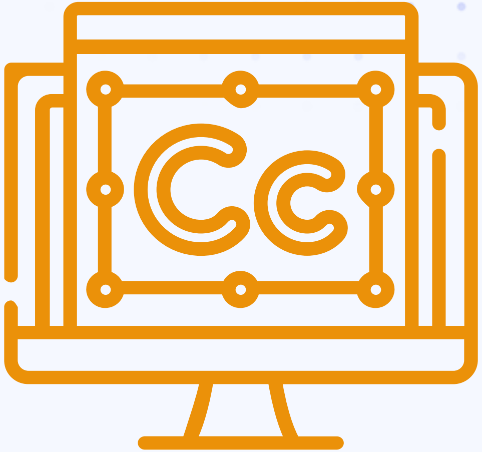
01
Purpose of experiment



- Understand Constraint Satisfaction problems to develop a goal state.

- Study the effectiveness of Backtracking Search, Min-Conflicts, Arc Consistency 3, and Forward Checking algorithms to identify the optimal solution for the zebra puzzle.

- Evaluate AI performance across different measurements such as solution status, execution time, and number of steps taken to solve the problem.

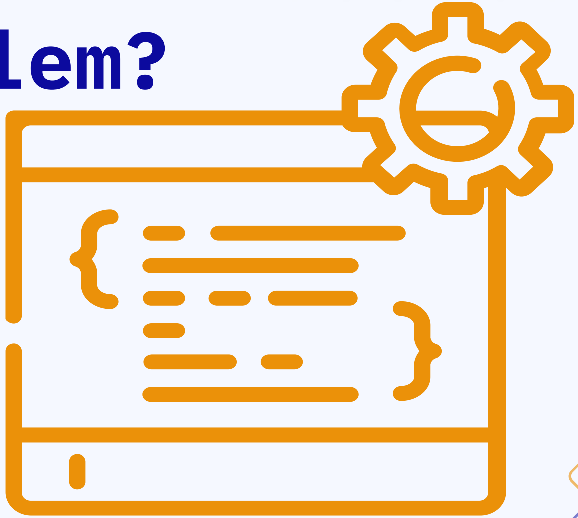


02

Key questions and hypothesis

Can an AI agent using Arc Consistency 3 perform effectively in a Constrained Satisfaction Problem?

- Does the amount of simulations affect the performance of the agent?
- How can efficiency be measured for an AI agent in the zebra puzzle problem?



Hypothesis: The Arc Consistency 3 algorithm combined with backtracking search will have the best overall performance due to its consistency enforcing capabilities and robustness in constraint satisfaction scenarios.

03

Concepts

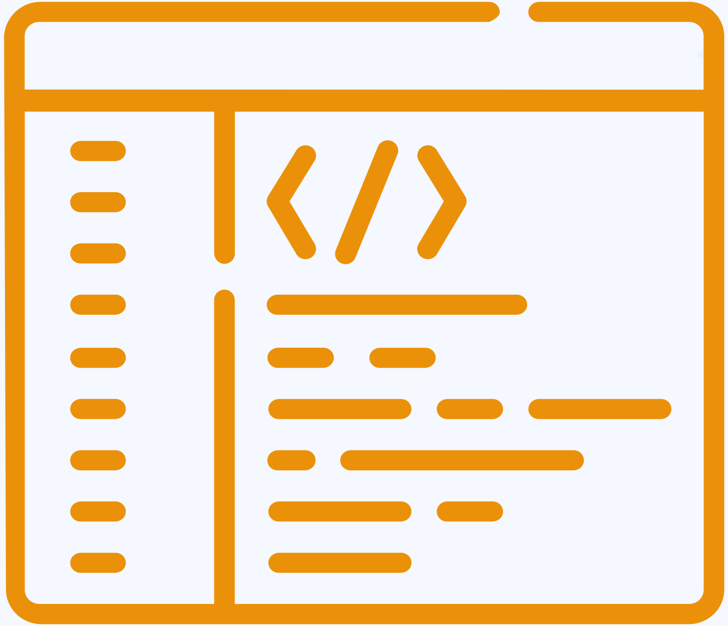
Experimental concepts

Platform

Python: Programming language of high level. Emphasizes code readability with the use of indentation.

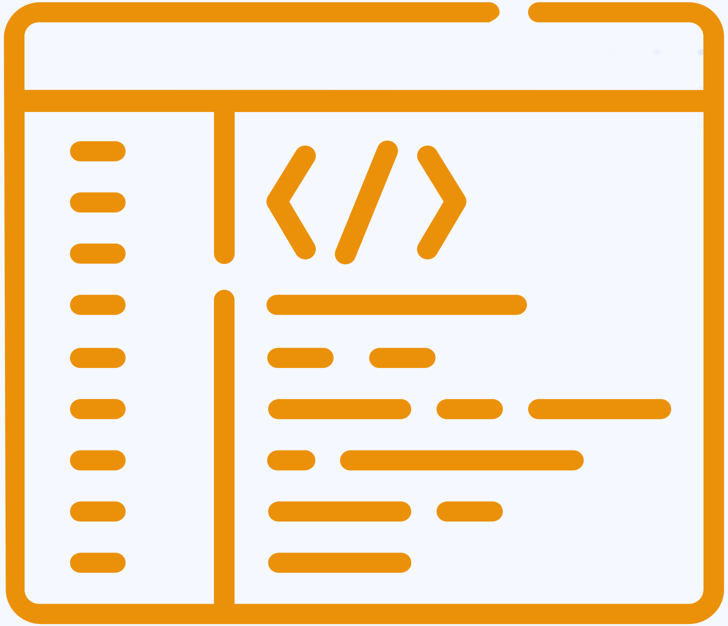
Subjects

- Zebra Puzzle
- Backtracking Algorithm
- Min-conflicts Algorithm
- AC-3 with Backtracking
- Forward Checking Algorithm



Experimental concepts

Zebra Puzzle Solver – A digital logic puzzle solver using CSP techniques like backtracking, min-conflicts, AC-3 with backtracking, and forward checking. Solutions are visualized with a GUI showing house attributes and highlighting the zebra and water locations.



Measure

Performance is being evaluated by checking if a valid solution is found and efficiency by measuring execution time and steps taken. These metrics highlight the strengths of each CSP algorithm.

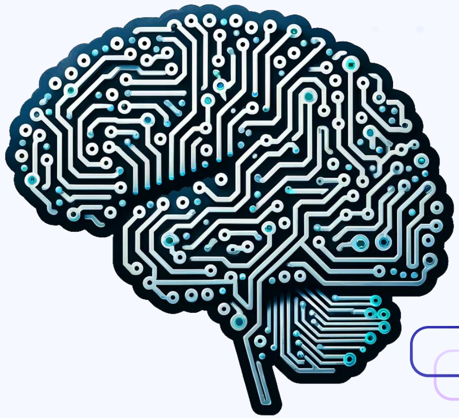
04
Experiments set up.

Tools and Resources Utilized

The following tools and resources were used:

- **Aimacode Repository**: The core repository containing foundational code and algorithms for the Zebra Puzzle.

- **Algorithms used to solve the puzzle**:
  - **Backtracking Search:** A depth-first search algorithm that incrementally builds solutions and backtracks when a constraint is violated.
  - **Min-Conflicts Heuristic:** A local search algorithm that starts with a complete but possibly invalid assignment and iteratively minimizes conflicts by changing variable values.
  - **AC-3 with Backtracking:** A combination of the AC-3 algorithm, which enforces arc consistency by reducing variable domains, and backtracking to find consistent solutions.
  - **Forward Checking:** An enhanced backtracking method that looks ahead by pruning the domain of unassigned variables when a new variable is assigned, preventing future conflicts.



Method for comparing

Criteria Used for Comparison:

- **Performance:** Indicates whether the algorithm successfully found a solution to the Zebra Puzzle.

- **Efficiency:** Measures the number of steps taken and the time required by each algorithm to solve the puzzle, when a solution is found.

Method for graph creation

How the Data for the graph was collected and used.

- Data was collected by running each CSP algorithm once to solve the Zebra Puzzle.

- The graph compares the performance of each algorithm based on the time taken and the number of steps required to reach a solution. If no solution was found, it shows the number of steps completed before the algorithm stopped.

05

Information.

Backtracking

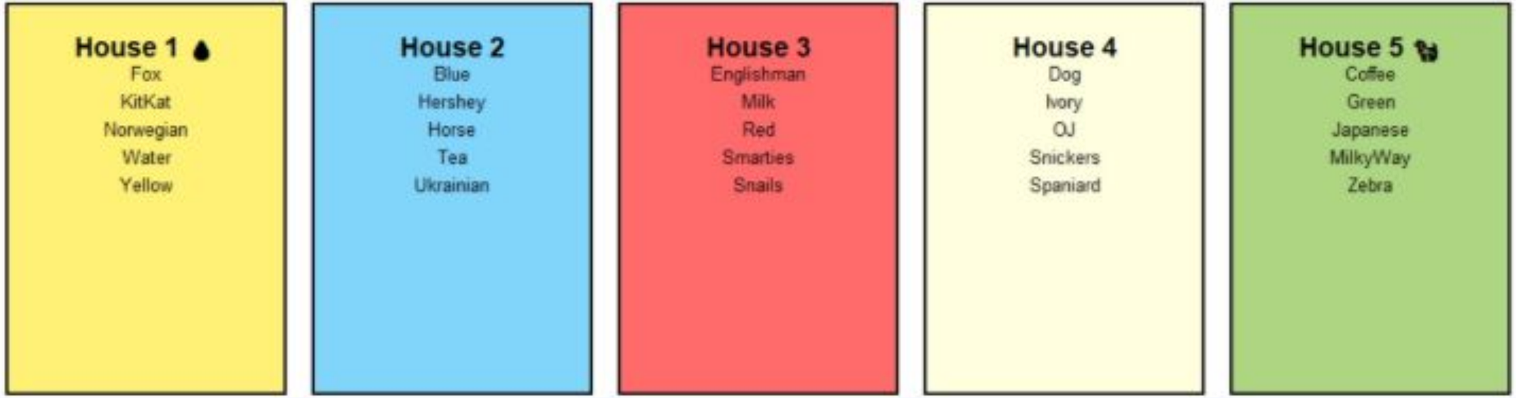


Figure 1. Zebra Puzzle solved state utilizing Backtracking Algorithm

AC-3 with Backtracking

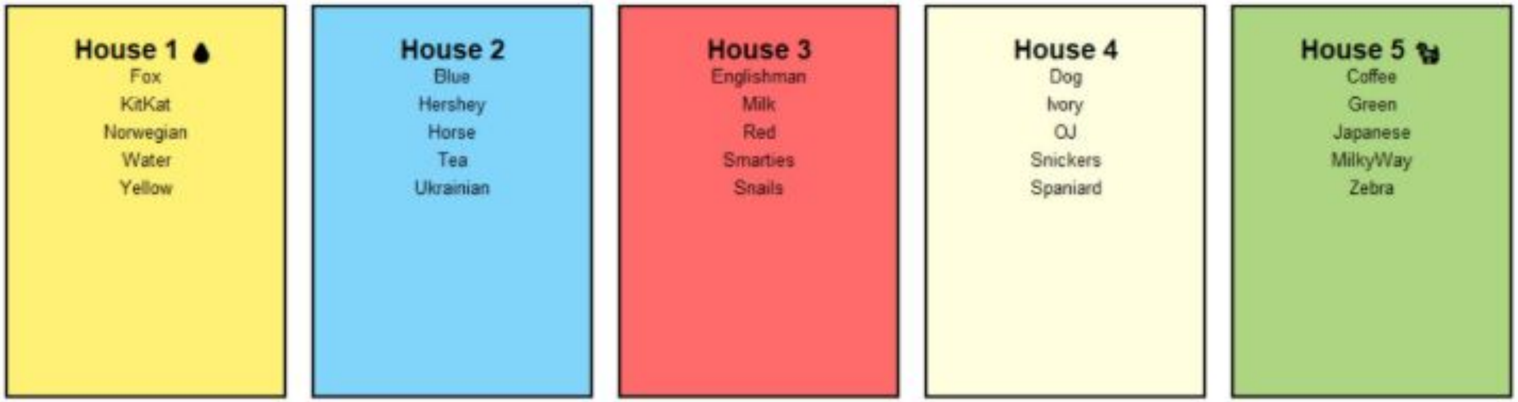


Figure 2. Zebra Puzzle solved state utilizing AC-3 + Backtracking Algorithm

Forward Checking

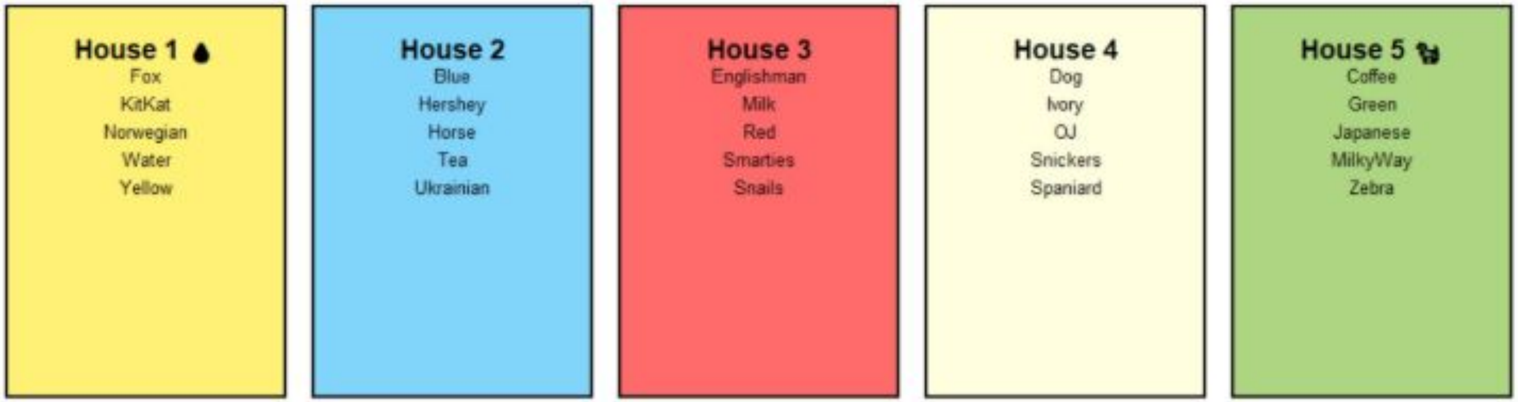


Figure 3. Zebra Puzzle solved state utilizing Forward Checking Algorithm

Information about the Puzzle

Algorithm	Status	Time (seconds)	Steps taken to solve the problem
Backtracking	Success	1.3732	75586
Min Conflicts	Failure	4.2799	10000
AC3 + Backtracking	Success	0.0289	1684
Forward Checking	Success	0.1614	4353

Table 1. Algorithms Performance Results

Information about the Puzzle

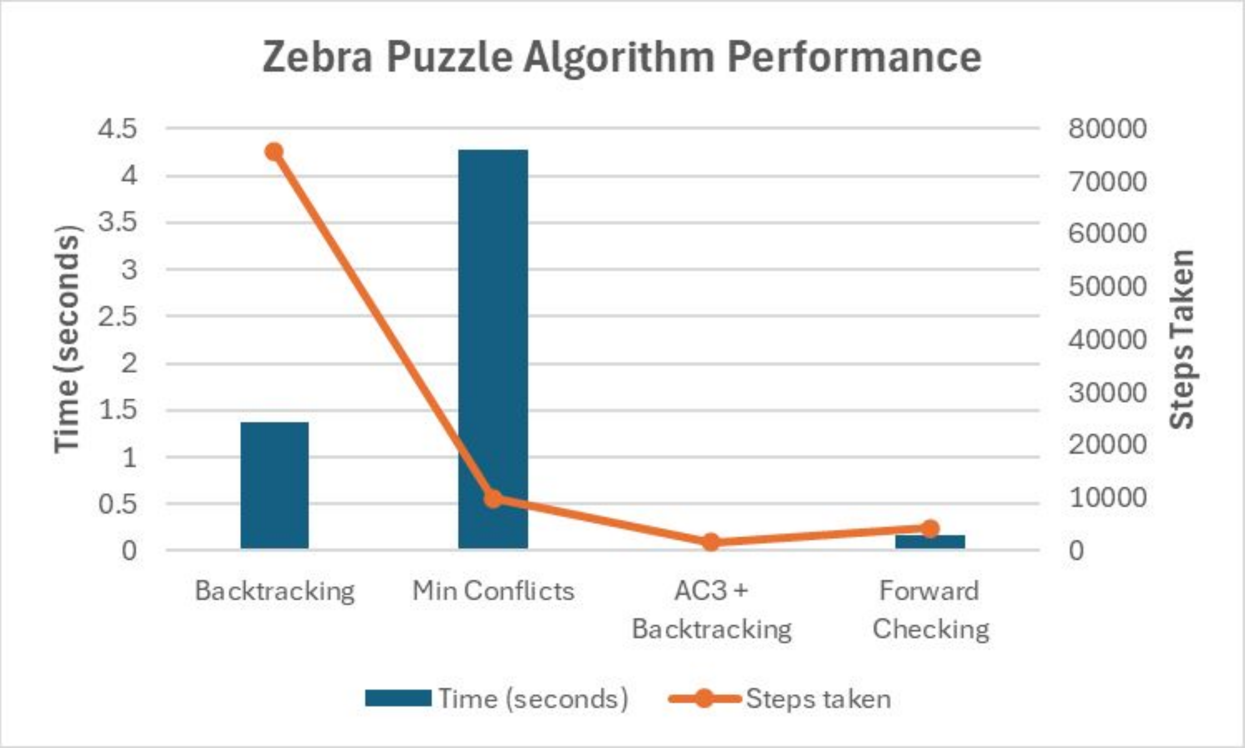


Figure 4. Performance measurements graph of the different algorithms solving the CSP

06

Conclusion and Lessons Learned.

Conclusions and Lessons Learned

01 **CSP Algorithm Performance**

Different CSP algorithms vary in how effectively they solve structured, constraint-heavy problems like the Zebra puzzle. Each method balances completeness, efficiency, and simplicity in distinct ways, revealing the strengths and trade-offs of approaches such as backtracking, forward checking, arc-consistency, and min-conflicts.

02 **Efficiency of AC-3 with Backtracking**

Backtracking with AC-3 offers a highly effective solution by significantly reducing the search space early on. This approach outperforms both standard backtracking and forward checking in terms of speed and consistency, requiring fewer recursive steps to reach a valid solution.

Conclusions and Lessons Learned

03 Limitations of Min-Conflicts

The min-conflicts algorithm struggles with tightly constrained problems due to its reliance on random restarts and local adjustments. Its performance is inconsistent and sensitive to initial conditions, making it less suitable for structured puzzles with complex interdependent constraints.

07

Credits and References.

References

[1] S.Yan, "Solving Real-World Puzzles: Constraint Satisfaction Problems for Data Science & Applied AI (incl. FREE Stanford-Berkeley Resources)," *medium.com*, Apr. 13,2025. [Online]. Available: https://luluyan.medium.com/solving-real-world-puzzles-constraint-satisfaction-problems-for-data-science-applied-ai-incl-44af36cf48c5 .[Accessed May. 3, 2025].

[2] AI Term Glossary, "Constraint Satisfaction Problem (CSP)," *ai-terms-glossary.com*, May 2025, [Online]. Available: https://ai-terms-glossary.com/item/constraint-satisfaction-problem/ .[Accessed May. 3, 2025].

[3] S. Russell, and P. Norvig, "Artificial Intelligence: A Modern Approach," *aima.cs.berkeley.edu*, Aug. 22, 2022. [Online]. Available : https://aima.cs.berkeley.edu . [Accessed May 4, 2025].

[4] Aimacode, "aimpa-python," *github.com*, Feb. 1, 2016. [Online]. Available: https://github.com/aimacode/aima-python . [Accessed May. 4, 2025].

[5] Geeks for Geeks, "Puzzle | The Einstein's Puzzle," *geeksforgeeks.org*, Feb. 22, 2023. [Online]. Available: Puzzle | The Einstein's Puzzle | GeeksforGeeks .[Accessed May. 3, 2025].

References

[6] Geeks for Geeks, “Types of Environments in AI,” *geeksforgeeks.org*, Aug. 5, 2024. [Online]. Available: Types of Environments in AI | GeeksforGeeks .[Accessed May. 4, 2025].

[7] Geeks for Geeks, “Introduction to Backtracking,” *geeksforgeeks.org*, Jun. 24, 2024. [Online]. Available: Introduction to Backtracking | GeeksforGeeks .[Accessed May. 4, 2025].

[8] N. Jeevanandam, “The Min-conflicts algorithm, Incremental learning, and k-means clustering,” *india.gov.in*, Nov 3, 2022. [Online]. Available: The Min-conflicts algorithm, Incremental learning, and k-means clustering.[Accessed May. 5, 2025].

[9] S.Edelkamp and S.schrödl, “Arc Consistency,”, *sciencedirect.com*, 2012. [Online]. Available: Arc Consistency - an overview | ScienceDirect Topics .[Accessed May. 5, 2025].