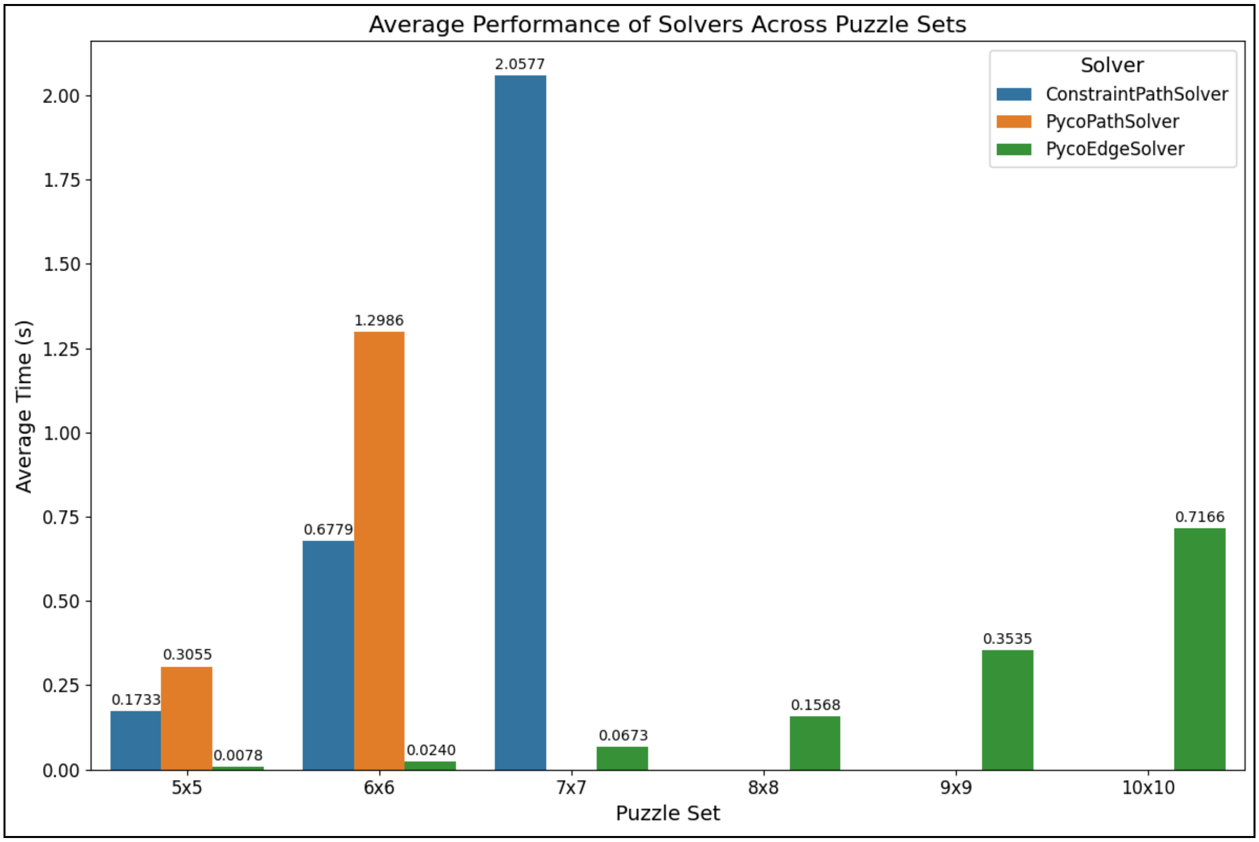


This bar chart compares the average performance times of three solvers—**ConstraintPathSolver**, **PycoPathSolver**, and **PycoEdgeSolver**—across puzzle sets of varying sizes (5x5, 6x6, 7x7, 8x8, 9x9, and 10x10). The following key points summarize the results:

1. **ConstraintPathSolver**:
   - Successfully completed puzzles for 5x5, 6x6, and 7x7 sets, with an increasing average time as the puzzles became larger (0.1733 seconds for 5x5, 0.6779 seconds for 6x6, and 2.0577 seconds for 7x7).
   - Did not complete puzzles for 8x8, 9x9, and 10x10 sets within the 10-second timeout.
2. **PycoPathSolver**:
   - Solved puzzles for 5x5 (0.3055 seconds) and 6x6 (1.2986 seconds) sets but did not solve larger puzzle sets within the timeout.
3. **PycoEdgeSolver**:
   - Demonstrated significantly faster performance for smaller puzzles, solving 5x5 in 0.0078 seconds and 6x6 in 0.0240 seconds.

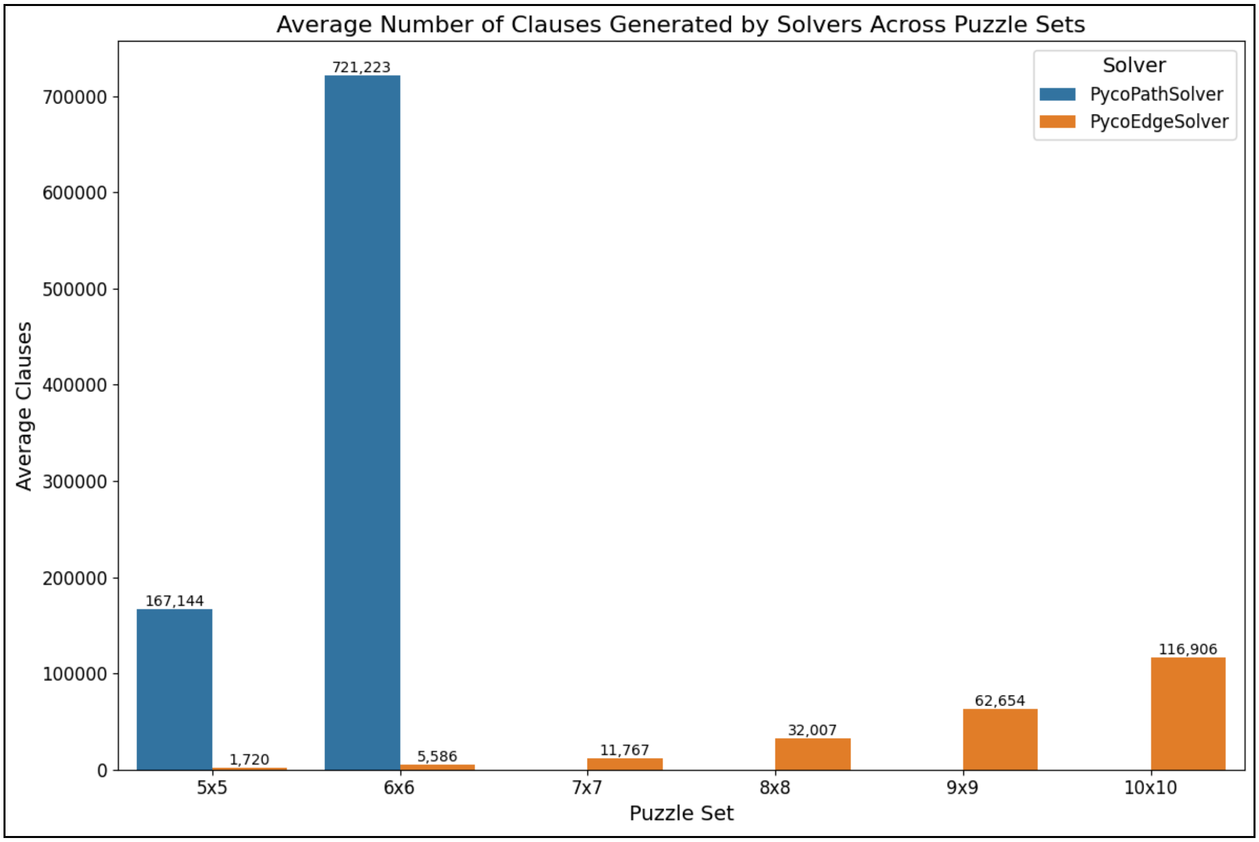
    - Continued to solve larger puzzles (7x7 to 10x10) with increasing times, ranging from 0.0673 seconds for 7x7 to 0.7166 seconds for 10x10.

Observations:

- **Efficiency**: PycoEdgeSolver is the most efficient solver, capable of solving all puzzle sets up to 10x10, with its average times increasing steadily with puzzle size but remaining well below the timeout limit.
- **Scalability**: ConstraintPathSolver and PycoPathSolver struggle with larger puzzles, timing out beyond 7x7 and 6x6 respectively.
- **Performance Trends**: All solvers show increasing average times with puzzle size, reflecting the growing complexity of larger puzzles.

Conclusion:

PycoEdgeSolver is the most robust and scalable solver, consistently solving all puzzles within the time limit. ConstraintPathSolver and PycoPathSolver perform well for smaller puzzles but lack scalability for larger and more complex puzzles.



This bar chart displays the average number of clauses generated by two solvers—**PycoPathSolver** and **PycoEdgeSolver**—across different puzzle sets (5x5 to 10x10). Here are the key takeaways:

PycoPathSolver:

- Generates a significantly higher number of clauses compared to PycoEdgeSolver, indicating a computationally more intensive approach.
- The number of clauses increases dramatically with puzzle size:
    - **167,144 clauses for 5x5** puzzles.
    - **721,223 clauses for 6x6** puzzles.
- Does not show data for puzzle sets larger than 6x6 because it timed out.

PycoEdgeSolver:

- Generates far fewer clauses, showcasing a more efficient approach.
- The number of clauses grows steadily with puzzle size, reflecting its scalability:
    - **1,720 clauses for 5x5** puzzles.
    - **5,586 clauses for 6x6** puzzles.

    - **11,767 clauses for 7x7** puzzles.
    - **32,007 clauses for 8x8** puzzles.
    - **62,654 clauses for 9x9** puzzles.
    - **116,906 clauses for 10x10** puzzles.

Observations:

1. **Efficiency**: PycoEdgeSolver's efficiency is evident in its ability to handle significantly larger puzzles (up to 10x10) with a steady increase in clause generation.
2. **Scalability**: PycoEdgeSolver demonstrates excellent scalability, while PycoPathSolver struggles to handle larger puzzles, as evident by the lack of data for puzzle sets beyond 6x6.
3. **Clause Growth**: The rapid growth in clause generation for PycoPathSolver highlights potential inefficiencies in its algorithm compared to the more controlled growth in PycoEdgeSolver.

Conclusion:

PycoEdgeSolver outperforms PycoPathSolver in terms of clause generation efficiency and scalability. While PycoPathSolver creates exponentially more clauses, it fails to scale beyond 6x6, making PycoEdgeSolver the more reliable choice for larger and more complex puzzles.