

Title: Optimized Multispace Evolutionary Search for Large-Scale Problems

Group: 73

Members:

- A. Praneeth - S2023001007 - Sec01
- K. Chidwipak - S20230010131 - Sec02
- K. Jeevan Kumar - S20230010121 - Sec03

Abstract:

We're building a better Multispace Evolutionary Search (MSES) based on a research paper. The paper uses a movie-picking system that's too tough for us to explain or code—it's super complicated. So, we're using 8-puzzle from class because it's easy to show, but our MSES can also handle huge problems! MSES fixes things by trying a full version and a small one, sharing good moves. We'll code it in C++ to solve 8-puzzle faster with smarter small puzzles and less sharing. A screen will show tiles moving. We'll test if ours beats the paper's by counting moves and time. Coding the screen might be hard, and settings could be tricky, but we'll test tons and start small. This project mixes class stuff with big ideas, and we hope you let us present it!

What We'll Do:

- **Why 8-Puzzle:**
 - The paper's MSES picks movies (10 from 1000s) with tons of details like genres—way too hard and slow to explain.
 - We picked 8-puzzle, a class game where you slide tiles to get 1-8, blank last. It's simple to code and show, but lets us test MSES for big problems too.
- **Normal Way vs. MSES:**
 - Normal searches like A* use one board. They try moves but get slow when puzzles are messy or huge.
 - MSES uses two boards—full and small—trying moves on both, sharing the best to solve fast. For 8-puzzle, it's fixing the game and a mini one together, and it scales to giant problems.
- **Paper's MSES for 8-Puzzle:**
 - Uses all 9 tiles on the full board.

- Makes a small board with 4 tiles, like 2,8,3,1.
- Ties moves: small board move helps full board.
- Tests lots of moves, keeps top ones, mixes them for better shots.
- Shares moves all the time to hit 1-8, blank last.
- Example: Board [2,8,3,1,6,4,7,0,5]. Full and small boards share moves to fix tiles quick.
- Paper uses MSES for big stuff like movies—it rocks for tons of choices.
- **Our Better MSES:**
 - Pick tiles by the blank for small board—they're key, so it's faster for 8-puzzle and big problems.
 - Share moves only when puzzle's close, like 2 tiles wrong—cuts wasted time.
 - Match moves simple: same move in both boards, no math—works for any size.
 - Example: Same board, small board uses 7,5,8. Share when nearly done, solve with less effort.
 - Test: Check if we're quicker, use fewer moves than paper's MSES.
- **Coding It:**
 - Code both MSES in C++ with puzzle as a grid.
 - Paper's: Two boards, share tons, solve.
 - Ours: Smart small board, share little, simple match—scales big.
 - Use SFML to show tiles slide on screen.
 - Test: Scramble puzzle, run both, count moves, time.

What It's Good For & Issues:

- **Good For:** Puzzles like 8-puzzle, huge problems like movie picks, schedules, or song lists—handles small and giant tasks.
- **Issues:** The paper's movie example was super hard to follow, with tons of details we couldn't get. It had too much math, like big equations, that confused us. The graphs showing results were tough to understand, not clear for us.
- **Fixes:** Start with easy 8-puzzle to learn MSES, use online C++ guides to code it, test step-by-step to make it clear.

Reference Paper:

L. Feng, Q. Shang, Y. Hou, K. C. Tan, and Y.-S. Ong, "Multispace Evolutionary Search for Large-Scale Optimization With Applications to Recommender Systems," in IEEE Transactions on Artificial Intelligence, vol. 4, no. 1, pp. 107-120, Feb. 2023.