

Using an Algorithm Portfolio to Solve Sokoban

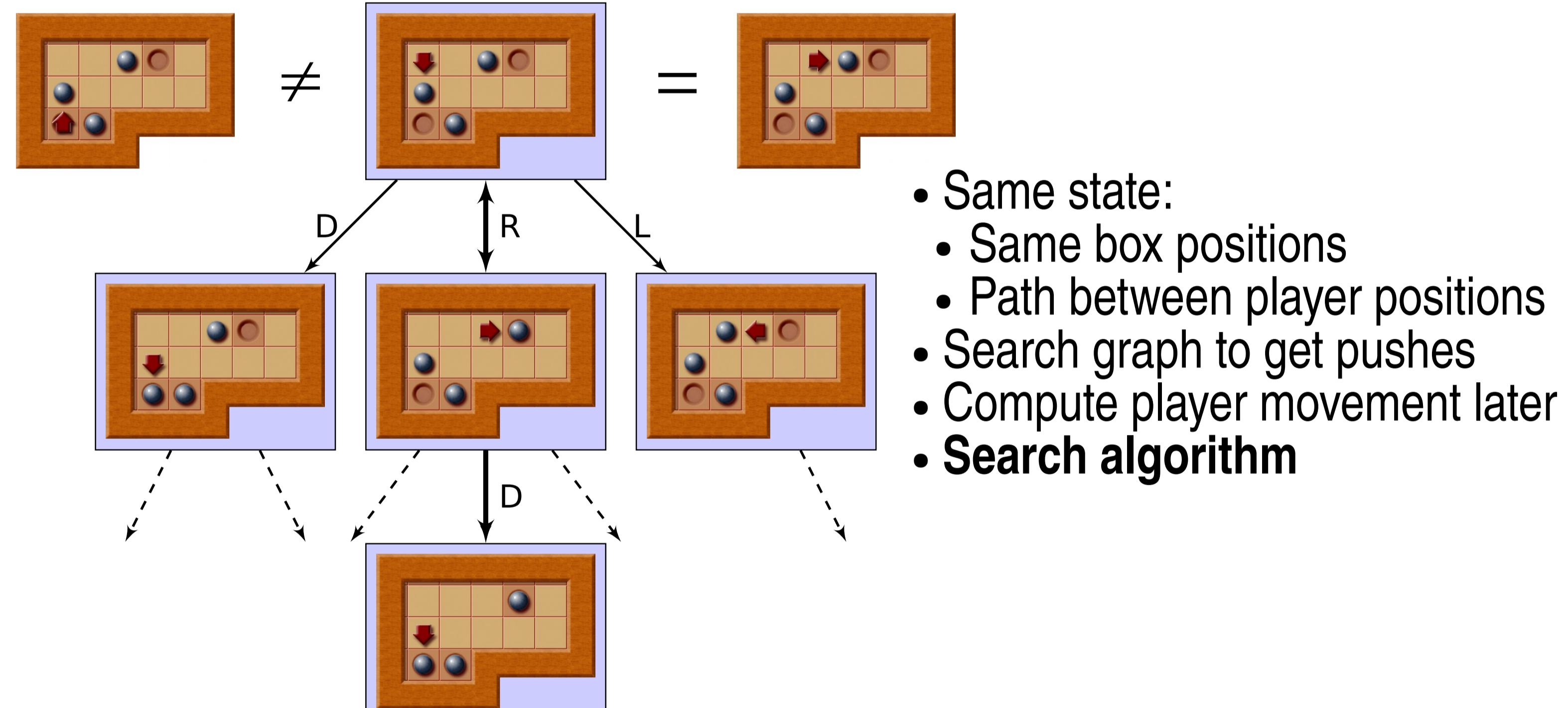
What is Sokoban?

- A PC puzzle game released in 1982 in Japan
- Goal: Push all boxes (one at a time) onto a goal
- Hard: PSPACE-complete, uneven search space, deadlocks

Algorithm Portfolio

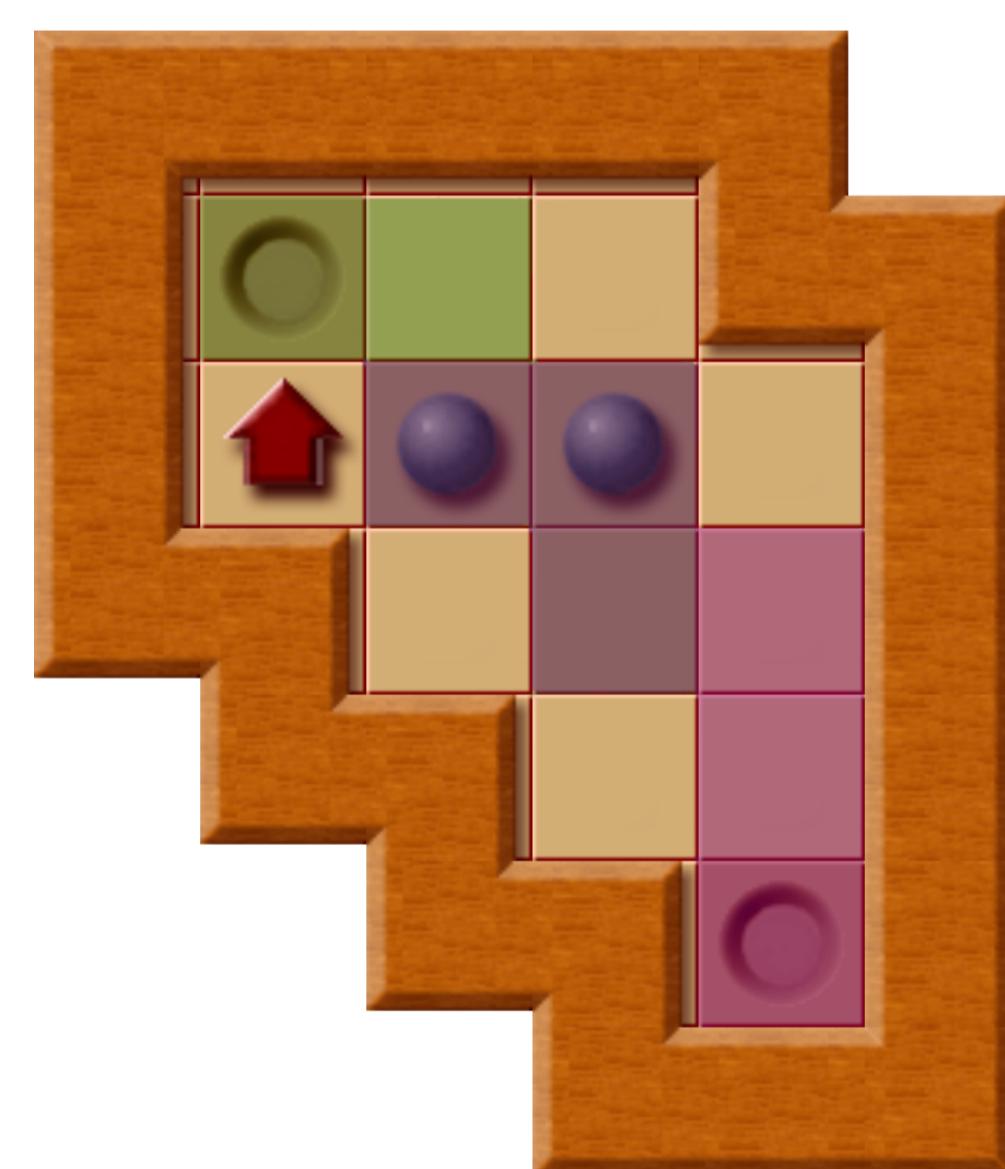
- Useful for NP-hard search problems
- Different algorithms have different run times on different instances
- Can be run in parallel or interleaved on a single processor
- Implement critical parts in multiple ways
- Assemble different solvers from those parts for the portfolio
- Diversity is important

State Space Search

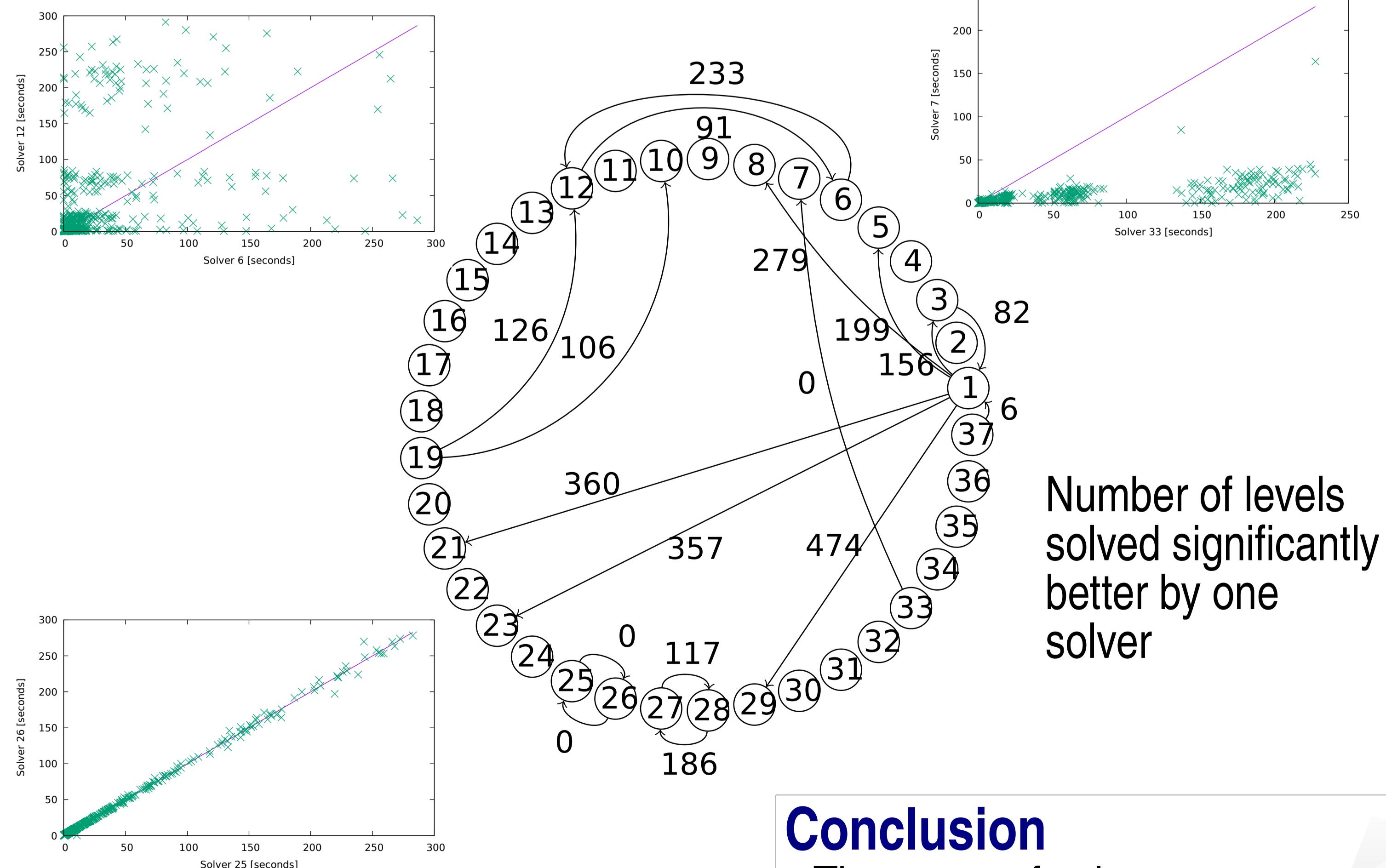


Count Area Detection

- Compute box-reachable area for each goal
- Find connected areas and store the number of reachable goals
- During the search maintain a counter of boxes in each area
- Number of boxes > reachable goals
→ Deadlock



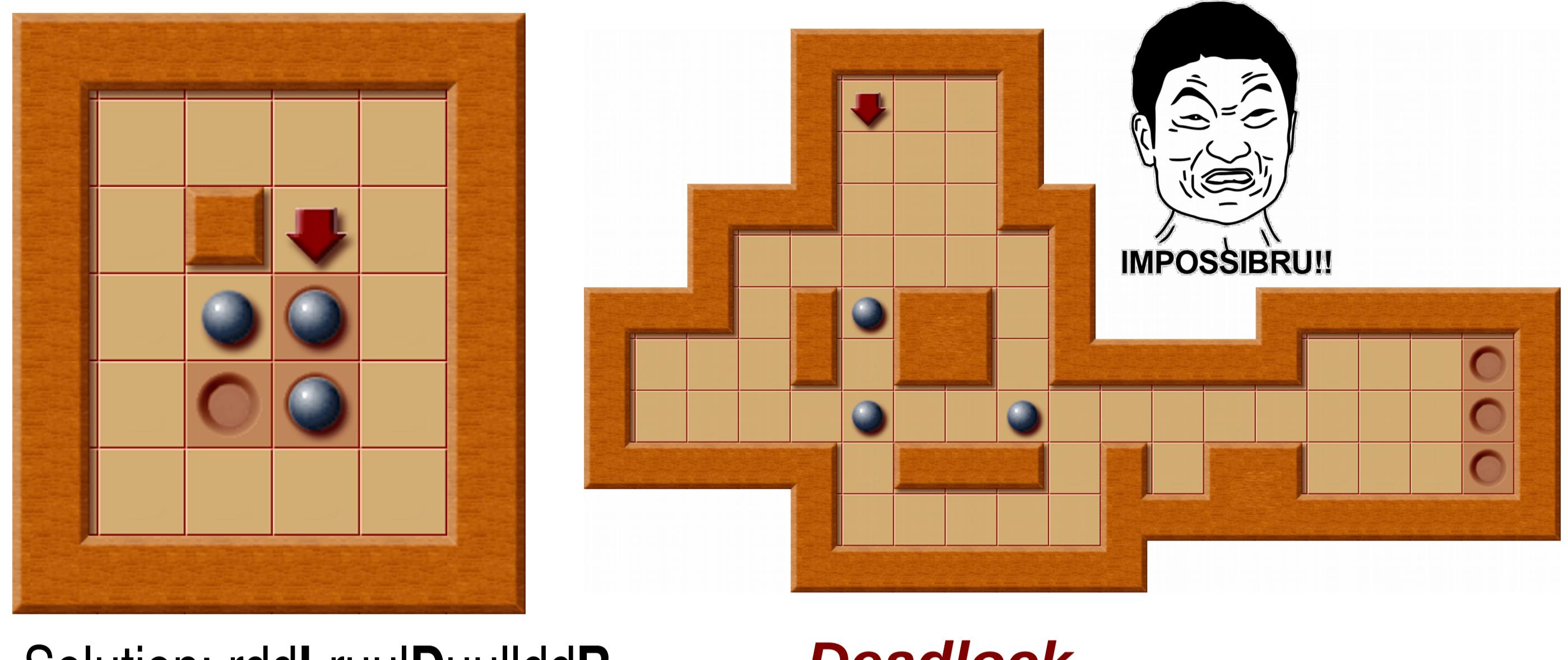
Diversity



Conclusion

- The group of solvers we present is diverse
- Our solvers miss some important features, but with the portfolio we can eventually surpass the others
- The approach of algorithm portfolios can be of value for hard search problems especially when traditional parallelization concepts are not well suited
- Future Work: More aggressive search enhancements (that might fail some times)

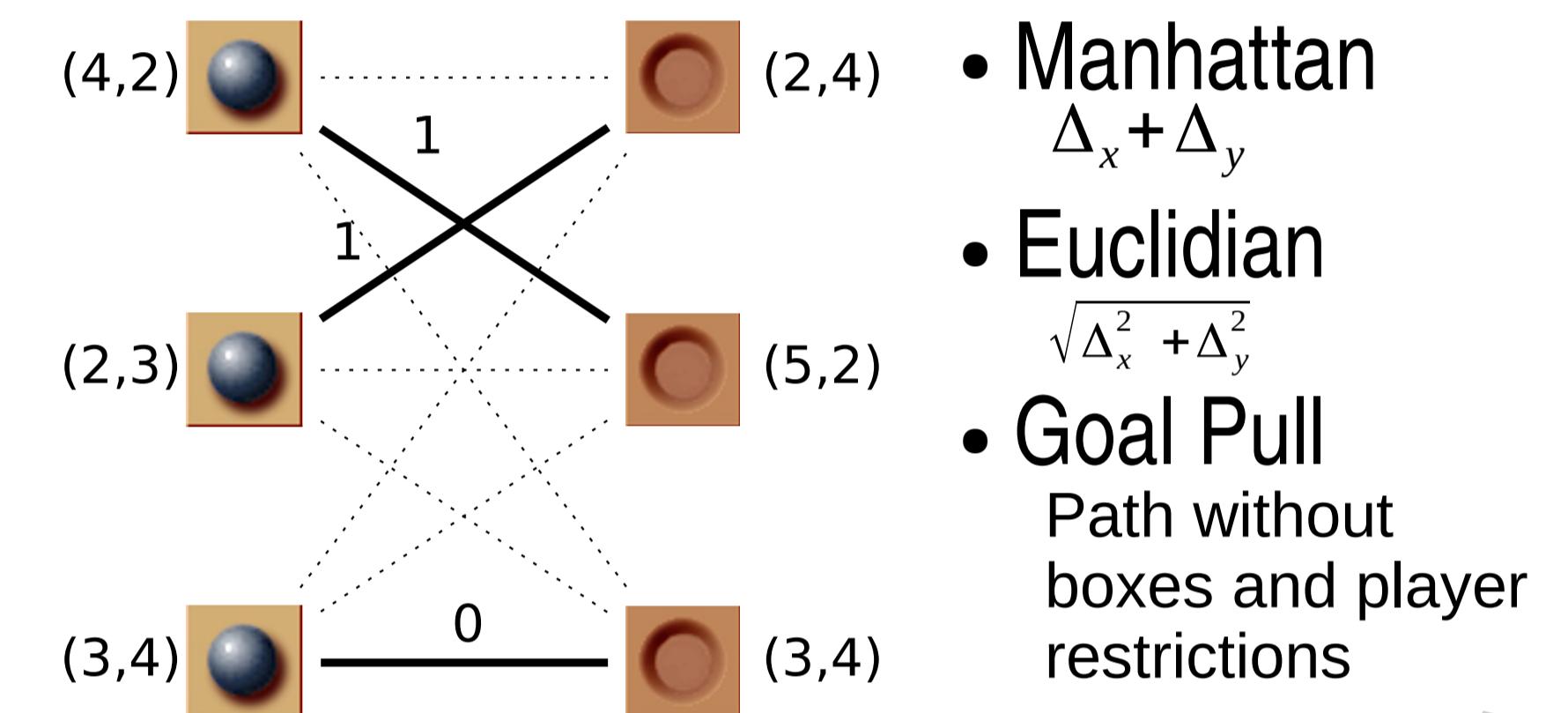
Example: one solvable and one unsolvable puzzle



Lower Bound

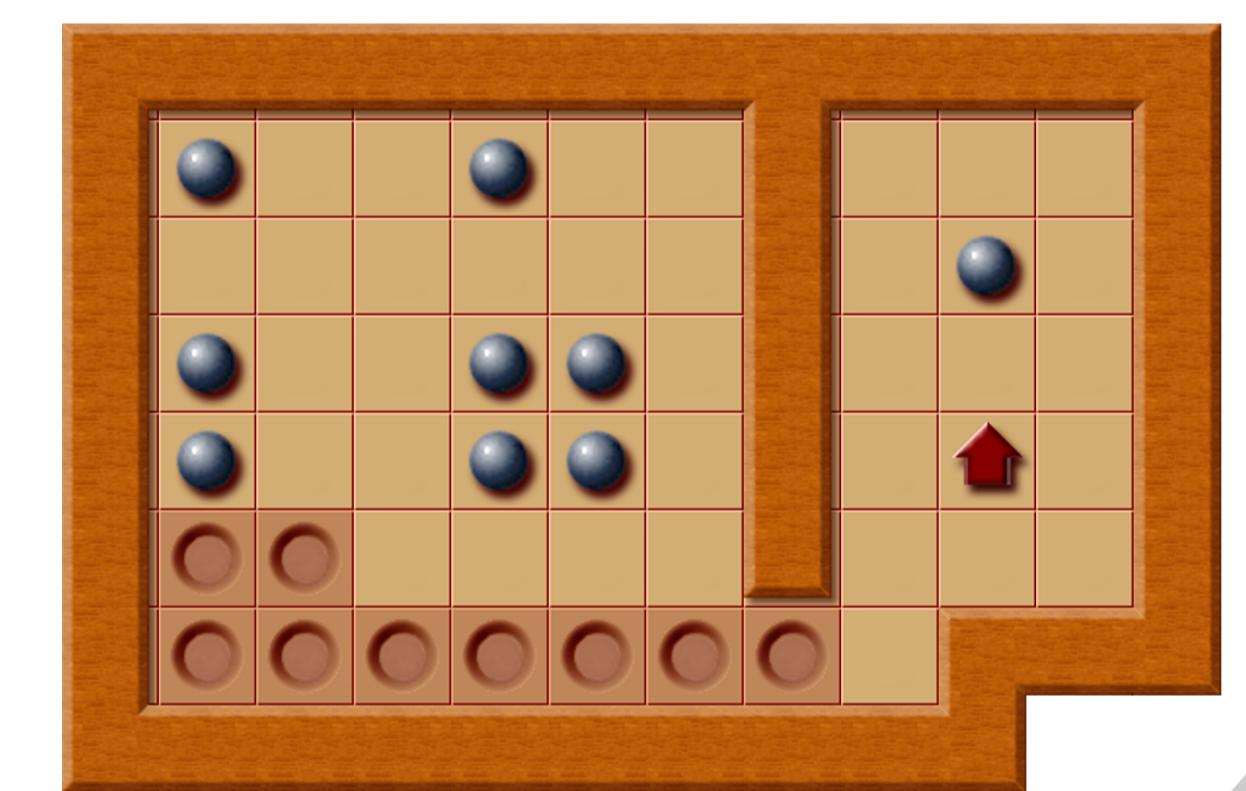
Assign a goal to each box and sum up the **distances**

- Closest assignment
- Hungarian method
- Greedy assignment



Deadlocks

- Make a level unsolvable
- Player position can be relevant
- Important to detect as soon as possible



Experiments

- Test set: 200 levels
- Timeout: 300 seconds
- PC: Two Intel Xeon E5-2650 (2x8 cores) @2.6 GHz and 128 GiB DDR3 RAM
- All other solvers are sequential

Comparison

