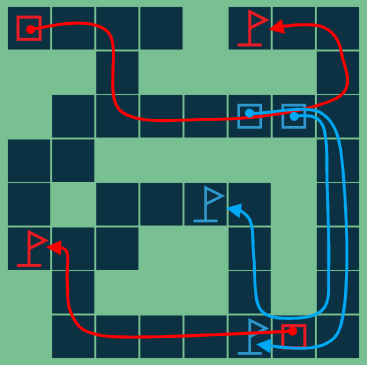
Extending EPEA* to solve Multi-Agent Pathfinding with Matching

[MAPFM]

In Multi-Agent Pathfinding (MAPF), agents move to a goal location without colliding.

In **Multi-Agent Pathfinding with Matching** (MAPFM), agents and goals are divided into teams.

An optimal solution is a path for each agent to a goal of the same team so that the **sum of path lengths** is minimized.



[EPEA*]

Enhanced Partial Expansion A* (EPEA*) is an **optimal** A*-based algorithm for MAPF that improves the memory efficiency and runtime of A* [1].

[Research Question]

How can EPEA* be extended to find MAPFM solutions?

- What are possible extensions and how do they compare?
- Are the solutions optimal?

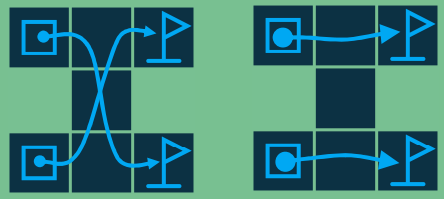
[Heuristic Matching]

The first extension that has been developed for this research is Heuristic Matching.

The A* **heuristic function** is modified to give the sum of distances to the closest goal of the same team for each agent. The problem is solved using one execution of EPEA*.

[Exhaustive Matching]

The second extension that has been developed is Exhaustive Matching. Exhaustive Matching runs EPEA* for **all possible goal assignments** and prunes the A* search when the cost of the best known solution is exceeded.

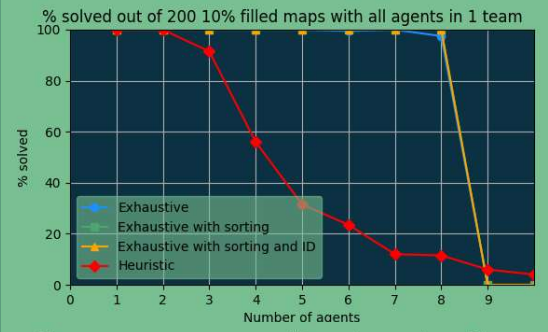


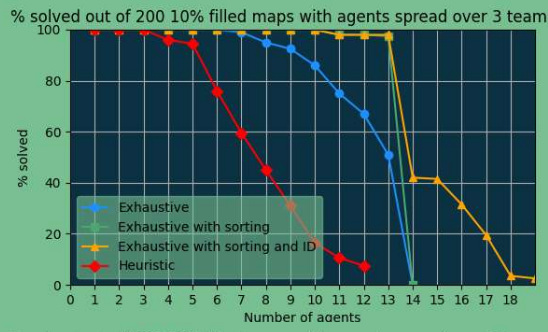
Two optimizations have been implemented for Exhaustive Matching:

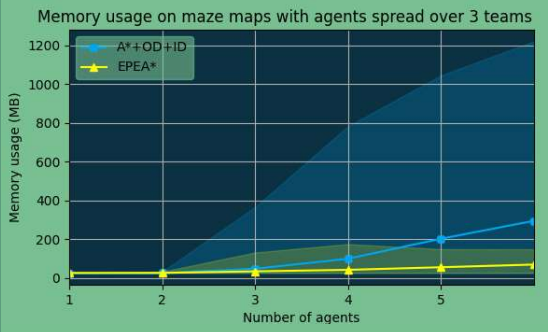
- **Sorting** evaluates the most promising goal assignments first, thus improving the pruning.
- **Independence Detection** tries to plan teams independently, which reduces the number of goal assignments if teams do not collide [2].

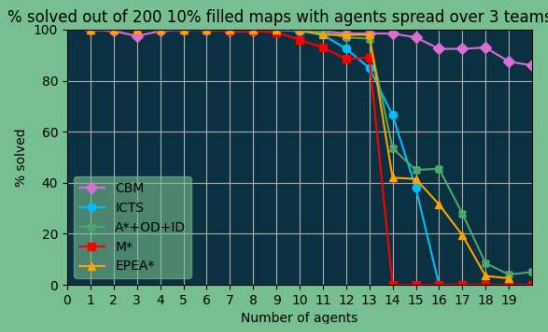
[Results]

For each set of parameters, 200 random 20x20 grids are generated, with random start and goal positions. The results show the percentage of problems that the algorithms were able to solve within 120 seconds and the memory usage of solved and unsolved problems.









[Conclusion]

Even though exhaustive matching seems inefficient, it actually performs quite well. The sorting and ID extensions are useful additions that improve the performance of exhaustive matching. The memory requirements of the algorithm are low compared to other solvers and the solutions are optimal.

The exhaustive matching solvers ICTS, A*+OD+ID, M* and EPEA* perform similar while the CBM-based solver outperforms them in this scenario.
For future work, it would be interesting to assess how the algorithms scale with map size.

TU Delft

Jaap de Jong
j.dejong-18@student.tudelft.nl
Supervisor: Jesse Mulderij
Responsible professor: Mathijs de Weerdt

References:
[1] Meir Goldenberg et al. “Enhanced Partial Expansion A*”. In: *Journal of Artificial Intelligence Research 50* (2014), pp. 141-187.
[2] Ivar de Bruin. “Solving Multi-Agent Pathfinding with Matching using A*+ID+OD”. 2021