>_ Extending A* to solve multi-agent path-finding problems with waypoints (MAPFW)

Stef Siekman

Supervisors
M. de Weerdt, J Mulderij

-= background =-

MAPFW

- Multiple agents: start & goal
- Move or wait each timestep
- No conflicts
- Waypoints for each agent to visit

Goal: lowest cost (sum of path lengths)

A* + OD + ID

A* • Pathfinding with heuristic
• Adapted for mutliple agents

OD • Expand a single move per agents per expansion
• Limits the size of A* search tree

ID • Solve for agents individually, where possible

-= research question =-

What is an effective way to extend A* with operator decomposition (OD) and indepence dectection (ID) to MAPFW?

- Best approach for heuristic calculation?
- Optimisations?
- Evaluation and comparison with alternatives

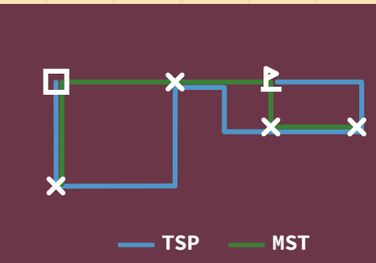
-= algorithm extension =-

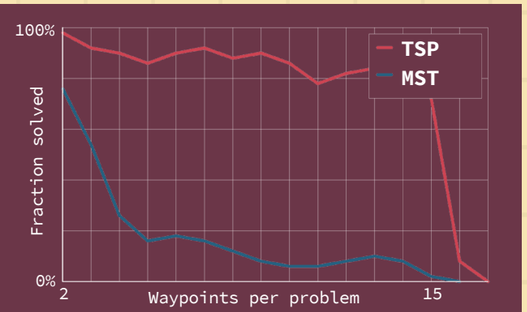
Approaches heuristic

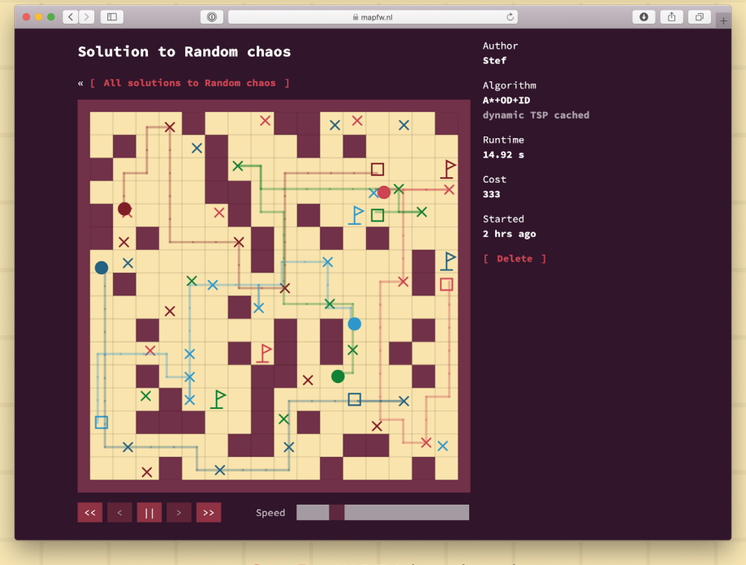
TSP + Optimal with regards to a single agent
- Expensive to compute

MST + Easy to compute
- Less accurate

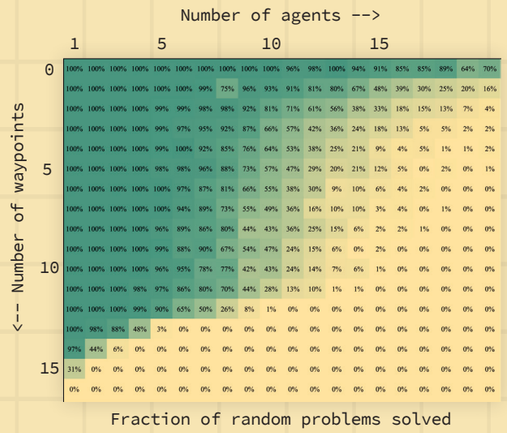
TSP: Traveling Salesperson Problem
MST: Minimal Spanning tree





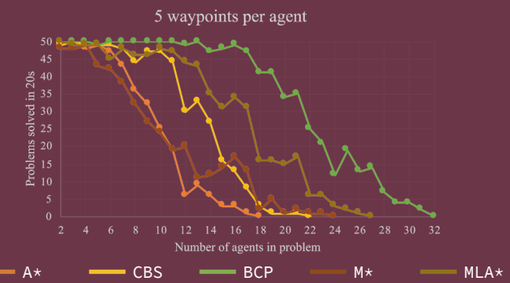


mapfw.nl: MAPFW benchmarks



-= experiments =-

Based on	Optimal	Language
A*	Yes	Python 3
CBS	Yes	Python 3
BCP	Yes	C++
M*	No (~0.5%)	Python 3
MLA*	No (~20%)	Python 3



Performance A* for MAPFW

- 100 random problems per cell
- How many were solved in 100s?

-= conclusion =-

A* + OD + ID	+ Easy to implement + Proven optimal - Small problems (12 agents/waypoints)
CBSW	+ Faster than A* + Optimisations
BCP	+ Fast for few waypoints
M*	+ Good performance for estimation
MLA*	+ Scales with more waypoints

-= limitations =-

Different implementations, languages