
# Multi Agent Pathfinding in Asprilo

Marius Wawerek

March 30,2021

University of Potsdam

# Table of Contents

# Table of Contents

# The MAPF Problem - An Example

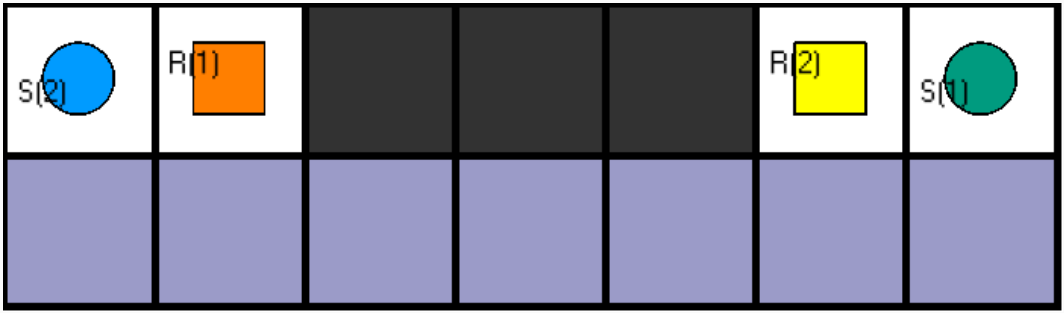
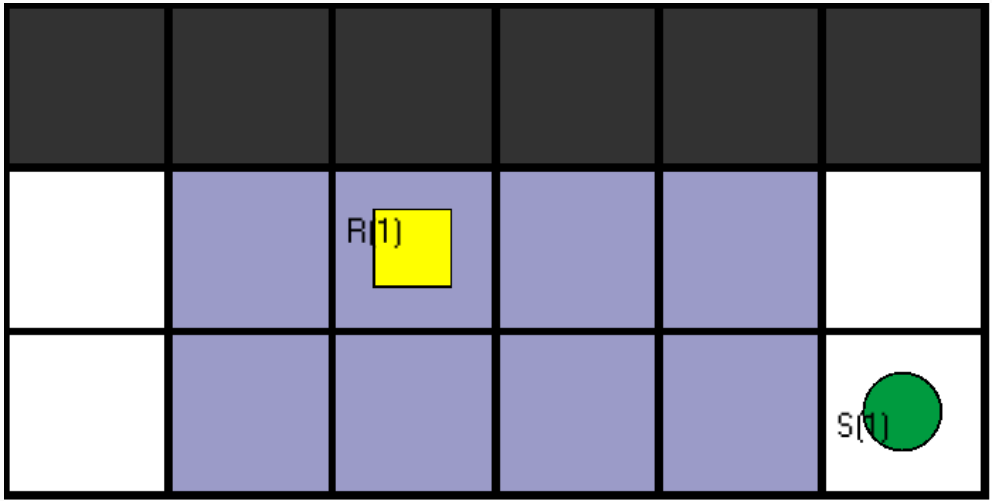



**Figure 1:** A first example of MAPF

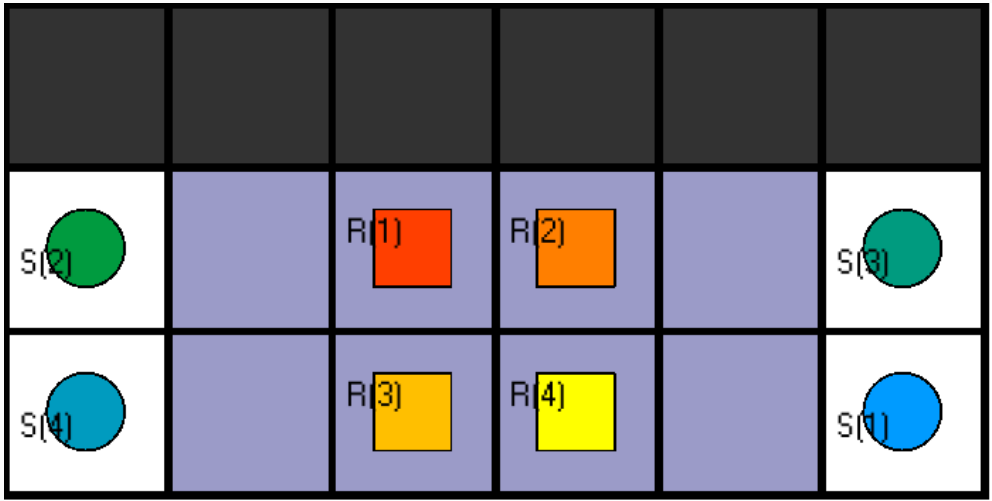
Each square wants to go the circle with the same number.

# Table of Contents

We want to move from a starting position to a goal.



We want to move from a starting position to a goal.



# MAPF in particular

Given

- A graph
- Starting positions for robots
- Goal positions for robots

Find path from start to goal for each robot.

# MAPF in particular

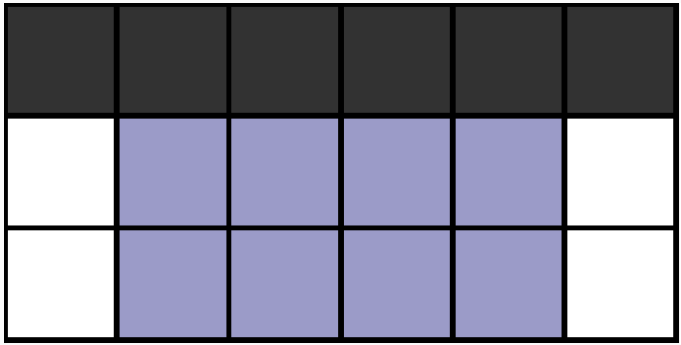
Given

- A graph
- Starting positions for robots
- Goal positions for robots

Find path from start to goal for each robot.

# Graphs

A graph is determined by vertices and edges.



In Asprilo Vertices are Nodes.

# MAPF in particular

Given

- A graph
- Starting positions for robots
- Goal positions for robots

Find path from start to goal for each robot.

# MAPF in particular

Given

- A graph
- Starting positions for robots
- Goal positions for robots

Find path from start to goal for each robot.

# Paths

Each robot has to move to a goal.
It moves step by step $\rightarrow$ we call these time steps
Each time step each robot takes an action.

## MAPF in particular

Given

- A graph
- Starting positions for robots
- Goal positions for robots

Find path from start to goal for each robot.

Constraints:

- No robots may share a position
- No robots may switch position

# Table of Contents

# How do we find a path?

Naive:
Try out every possible move for every robot.

# How do we find a path?

Naive:
Try out every possible move for every robot.

Result:
We have a solution.

# How do we find a path?

Naive:
Try out every possible move for every robot.

Result:
We have a solution.

Disadvantage:
It does not scale well.
It takes a exponential time to find it.

# How do we find a path?

Instead:

$k$ - Individual agent merger

# How do we find a path?

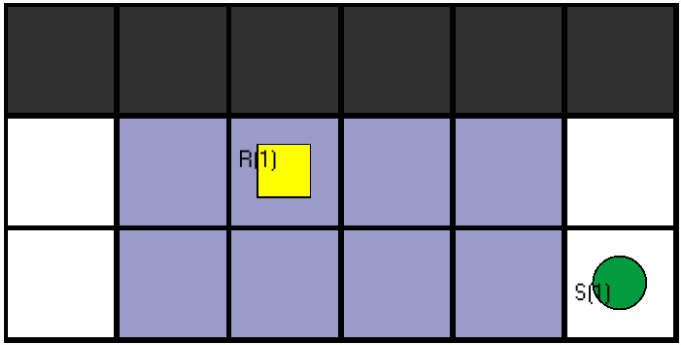
First generate individual plan per robot
Then “merge” these plans.

# How do we find a path?

First generate individual plan per robot
Then “merge” these plans.

# How do we find a path? - individual plans

Generate individual plan per robot



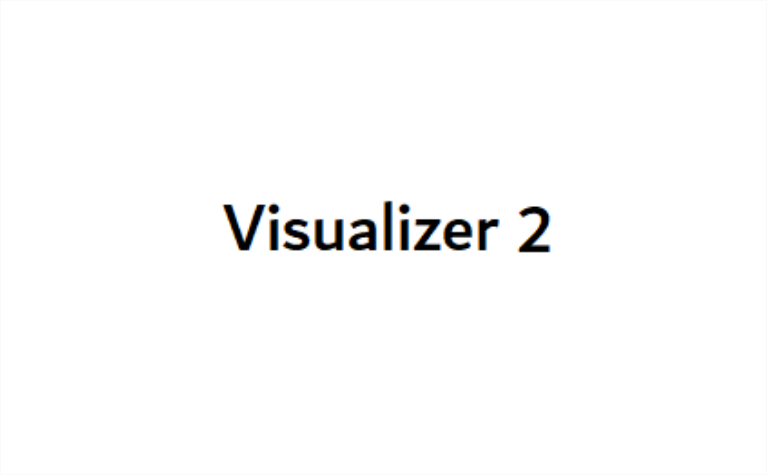

Find shortest path from start to goal.

# How do we find a path?

First generate individual plan per robot
Then “merge” these plans.

# How do we find a path? - Merging

Is ‘merging’ these plans difficult?



# How do we find a path? - Merging

Is ‘merging’ these plans difficult?

YES

Conflicts may arise.

# Table of Contents

# MAPF Mergers

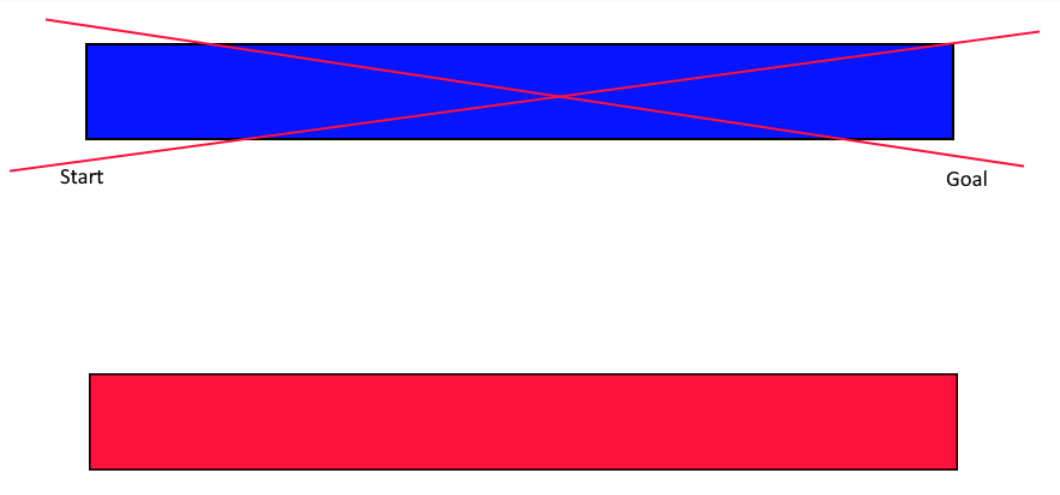
We need to modify the initial plans.
We created various plan mergers.

# Clean Plans

Here you can see the original plans

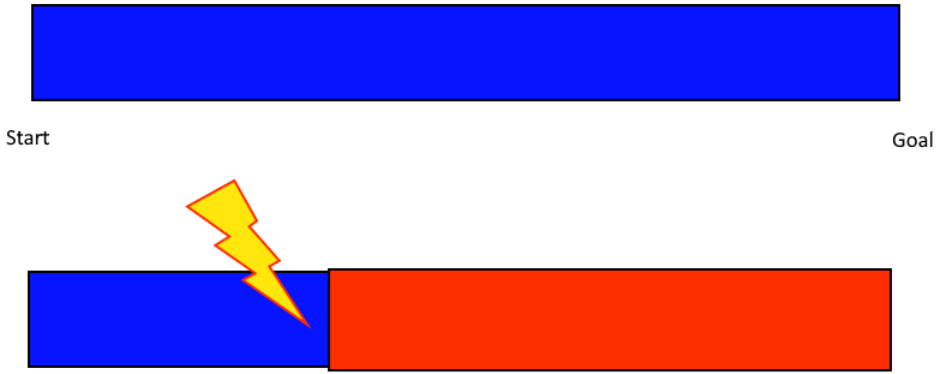
# Random Moves

Here you can see the ‘Random Moves’ merger
It throws away every original plan.



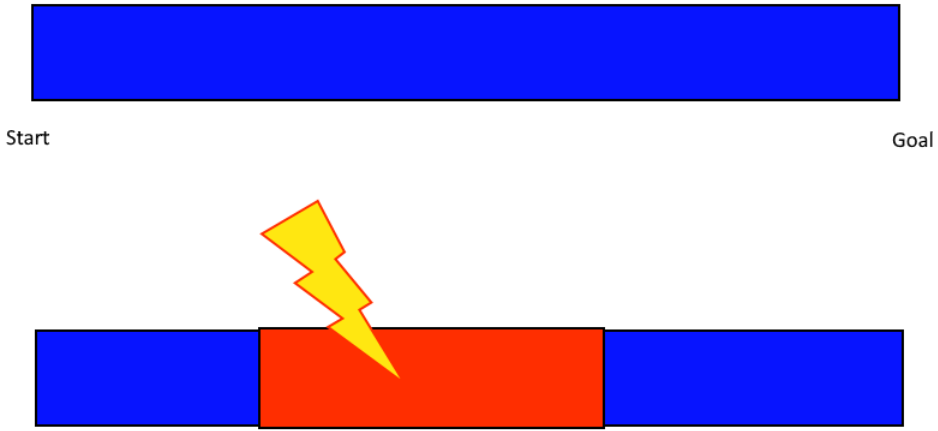
# Specific conflict

Here you can see the ‘Specific Conflict’ merger
It keeps the original plans until the first conflict



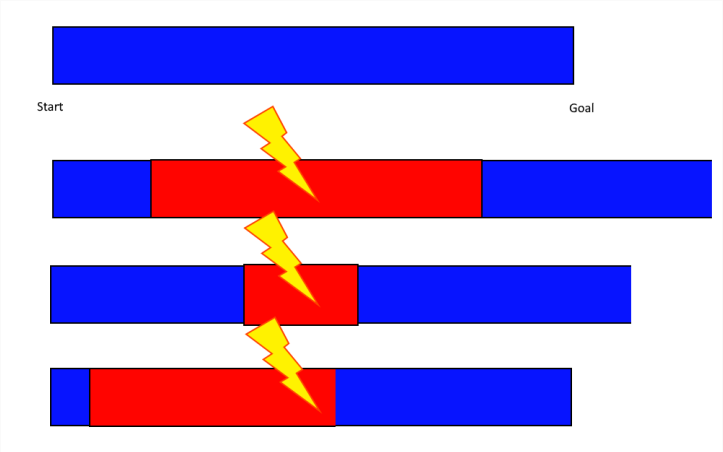
# Change Time

Here you can see the ‘Change Time’ merger
It cuts out the original plans around a conflict.



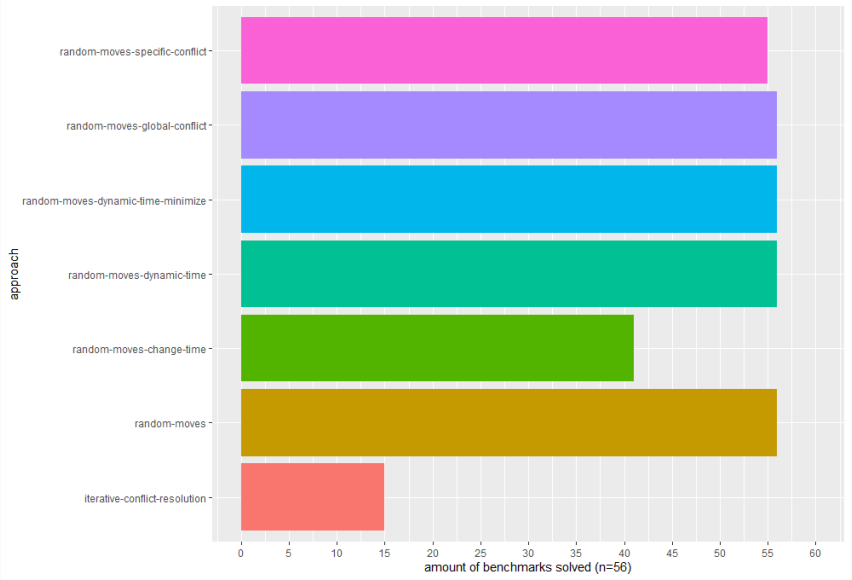
# Dynamic Time

Here you can see the ‘Dynnamic Time’ merger
It cuts the original plans before a conflict.
However it keeps the moves after a conflict. It may delay these moves.

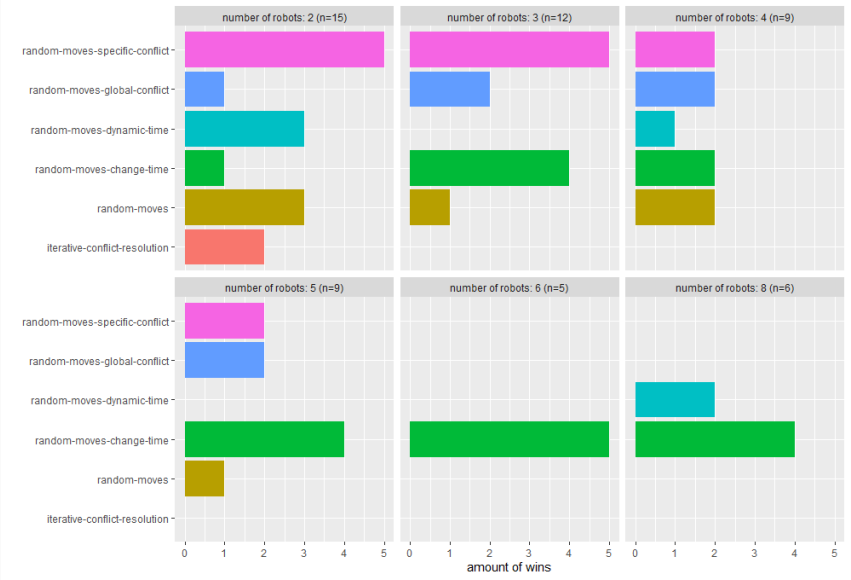


# Table of Contents

# Amount of Solvable Benchmarks



# Amount of Wins per Approach grouped by the number of robots



# Table of Contents

# Conclusion

There is no perfect approach.
It is a trade off between performance and solving ‘power’.

Thank you for your attention

Any Questions?