Multi Agent Pathfinding in Asprilo

Marius Wawerek Niklas Kämmer

March 30,2021

University of Potsdam

Table of Contents

General Problem Setting

How to calculate solution?

How to solve conflicts?

Experimental Results

Conclusion

Table of Contents

General Problem Setting

How to calculate solution?

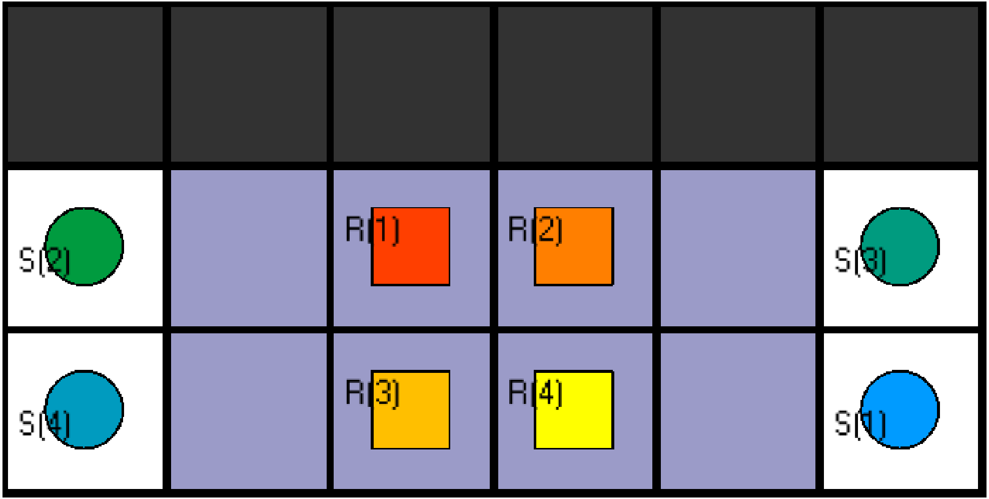
How to solve conflicts?

Experimental Results

Conclusion

Multi Agent Pathfinding in general

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 Asprilo

Asprilo is a framework in ASP. It can represent a multitude of problems.

We use the ‘M’ - Domain of Asprilo.

MAPF in particular

Given

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

Find path from start to goal for each robot.

Paths in Asprilo

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

Find path from start to goal for each robot.

Constraints:

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

Table of Contents

General Problem Setting

How to calculate solution?

How to solve conflicts?

Experimental Results

Conclusion

How do we find a path?

We use k - Individual agent merger

How do we find a path?

First generate individual plan per robot

Then “merge” these plans.

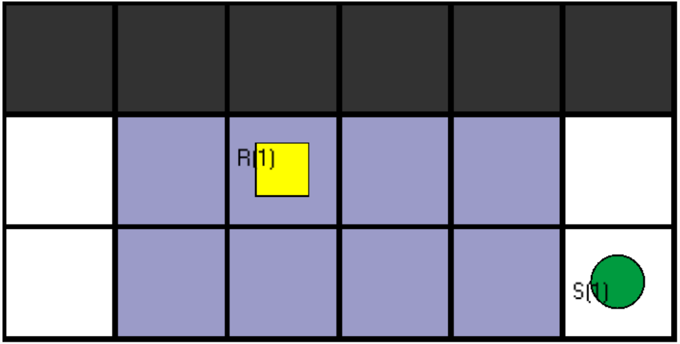
How do we find a path?

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.

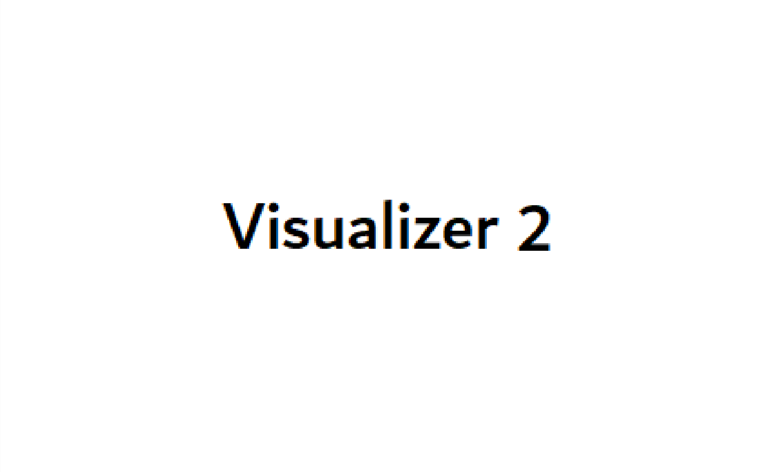
How do we find a path?

First generate individual plan per robot

Then “merge” these plans.

How do we find a path? - Merging

Is ‘merging’ these plans difficult?



Visualizer 2

How do we find a path? - Merging

Is ‘merging’ these plans difficult?

YES

Conflicts may arise.

Table of Contents

General Problem Setting

How to calculate solution?

How to solve conflicts?

Experimental Results

Conclusion

MAPF Mergers

We need to modify the initial plans.

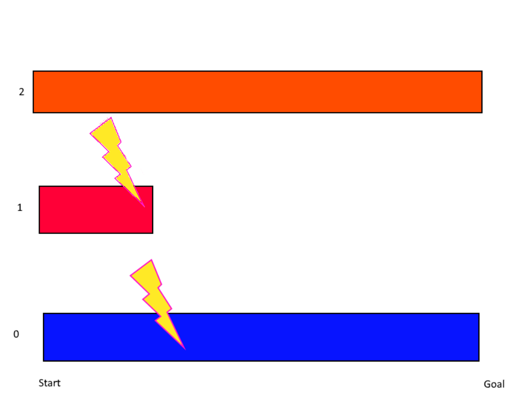
We created various plan mergers.

Clean Plans

Here you can see the original plans

Iterative Conflict Resolution

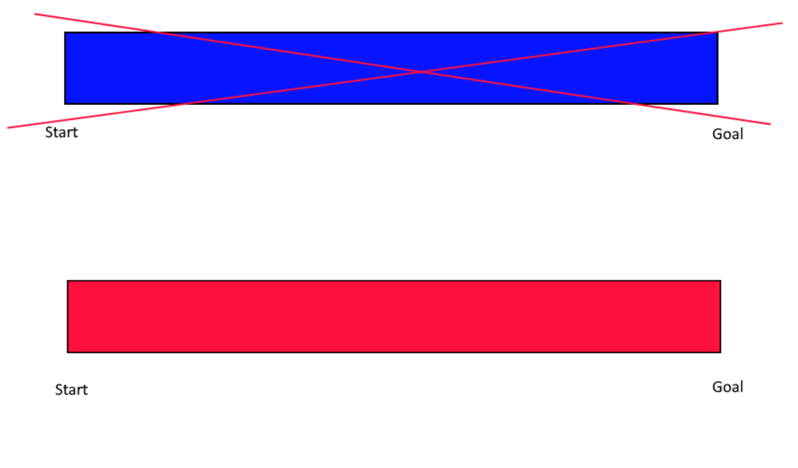
Here you can see the ‘Iterative Conflict Resolution’ merger
It tries to solve conflicts in multiple layers by using predefined movement.



Random Moves

Here you can see the ‘Random Moves’ merger

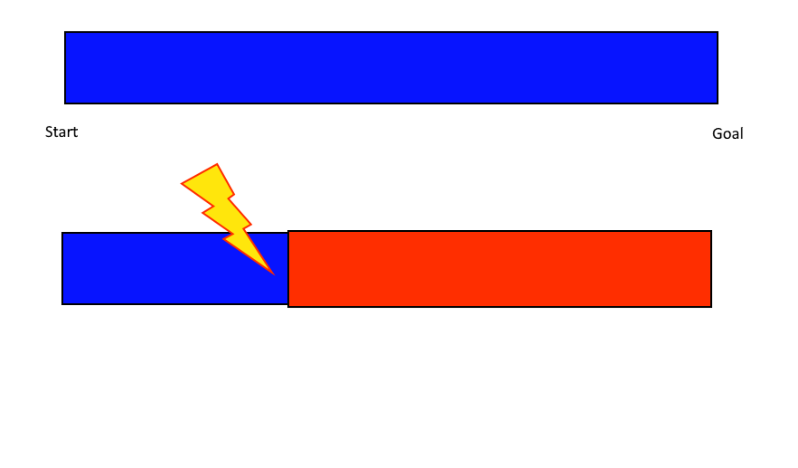
It throws away every original plan.



Specific conflict

Here you can see the ‘Specific Conflict’ merger

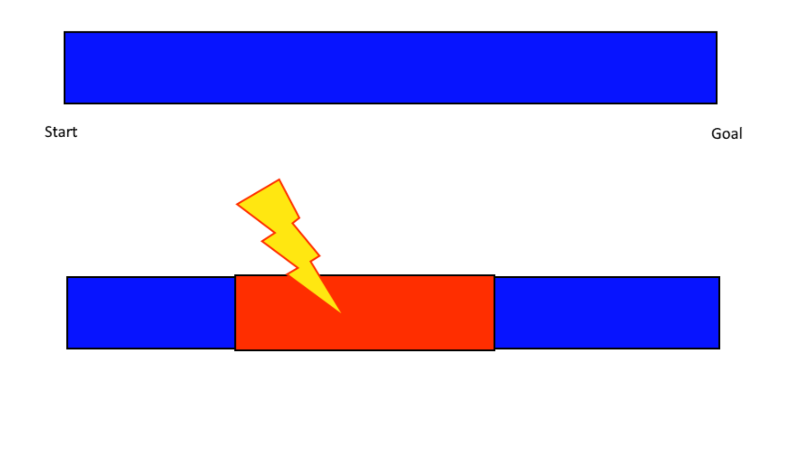
It keeps the original plans until the first conflict



Change Time

Here you can see the ‘Change Time’ merger

It cuts out the original plans around a conflict.



Dynamic Time

It cuts the original plans before a conflict.

However it keeps the moves after a conflict. It may delay these moves.

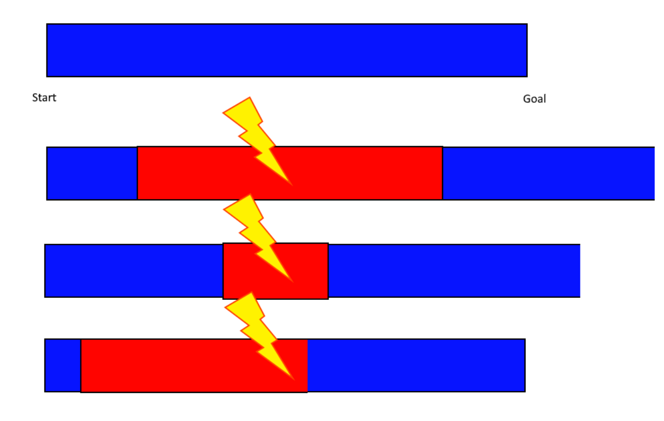


Table of Contents

General Problem Setting

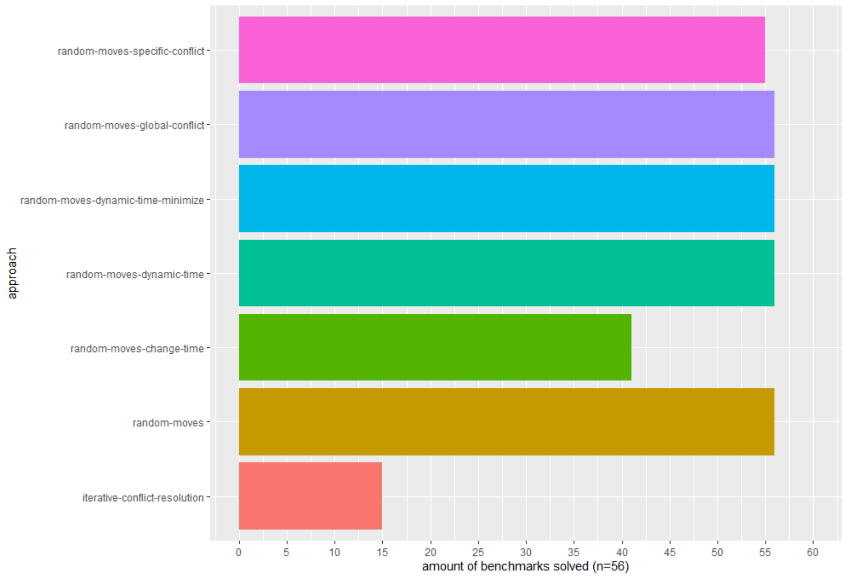
How to calculate solution?

How to solve conflicts?

Experimental Results

Conclusion

Amount of Solvable Benchmarks



Amount of Wins per Approach grouped by the number of robots

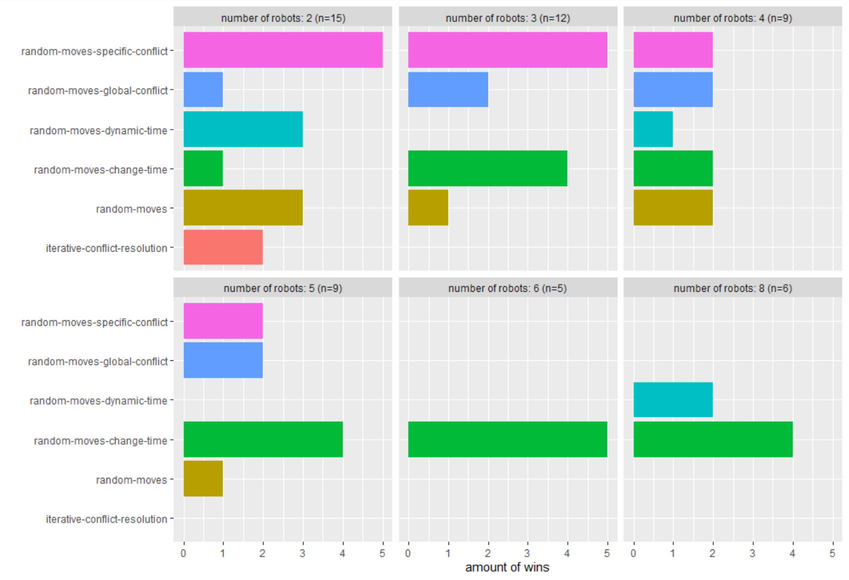


Table of Contents

General Problem Setting

How to calculate solution?

How to solve conflicts?

Experimental Results

Conclusion

Conclusion

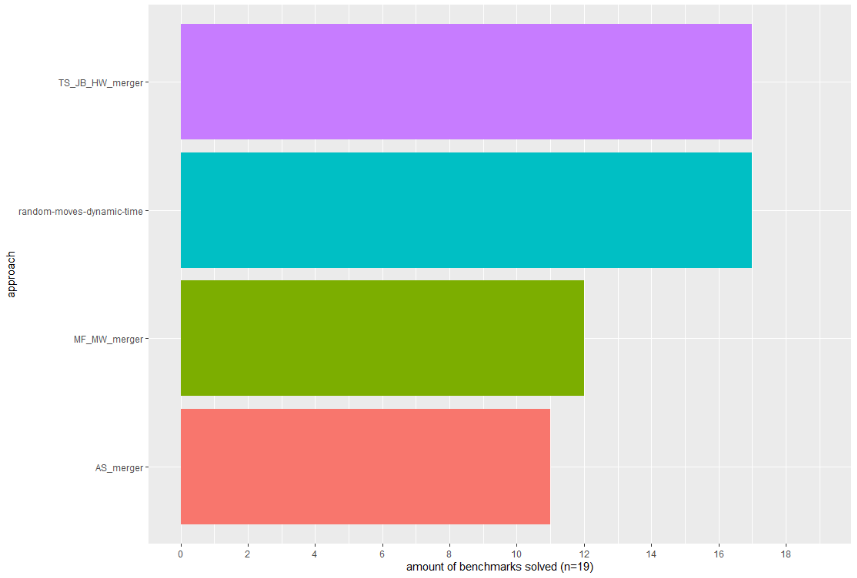
There is no perfect approach.

It is a trade off between performance and solving ‘power’.

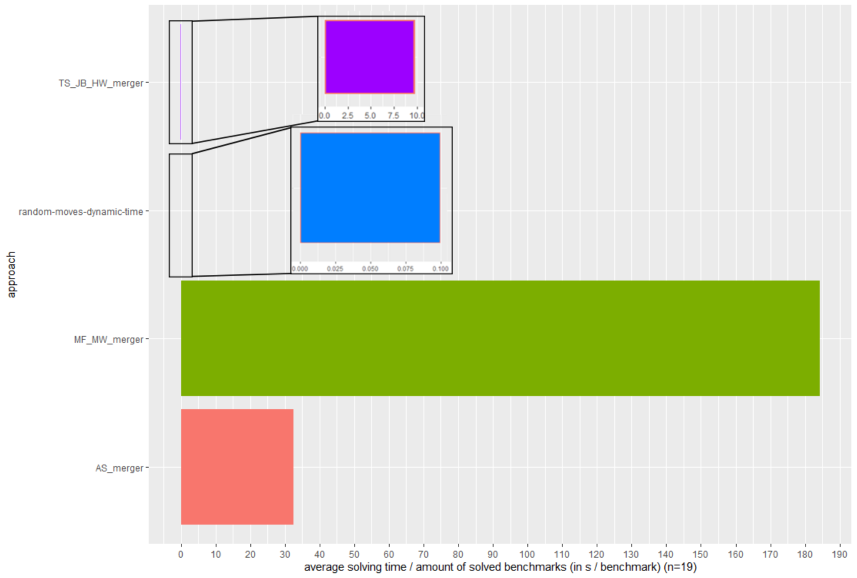
Thank you for your attention

Any Questions?

Appendix: Comparison to other Groups



Appendix: Comparison to other Groups


TS_JB_HW_merger
random-moves-dynamic-time
MF_MW_merger
AS_merger
approach
average solving time / amount of solved benchmarks (in s / benchmark) (n=19)