Multi-Agent Formation Control

For Implementation of Group Strategies

Aubrey Hormel
Mentor: Dr. Jingjin Yu

Problem Formulation

Initial and Goal Positions in Multi-Robot System

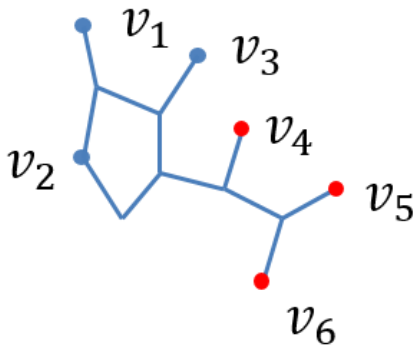
Agents move from initial to target nodes by travelling along edges in a connected graph.

Constraints:

- Optimize total distance travelled by all agents
- Avoid collision

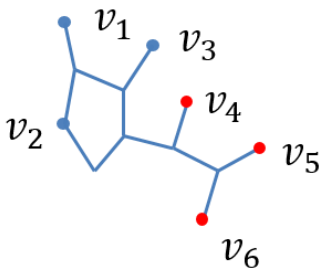
Allowances:

- Agents are indistinguishable - a given agent is allowed to be mapped to any goal node



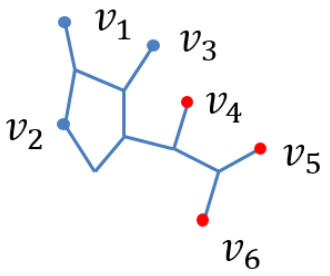
Distance-Optimal Pairing

Perform a Breadth-First Search (BFS) to construct a distance matrix between each initial/goal node pair.



Distance-Optimal Pairing

Perform a Breadth-First Search (BFS) to construct a distance matrix between each initial/goal node pair.

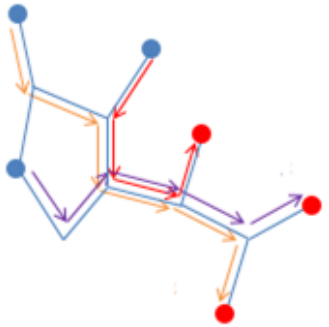


Perform Hungarian Algorithm to find optimal pairing

	v_4	v_5	v_6
v_1	5	6	6
v_2	4	5	5
v_3	4	5	5

Path Planning

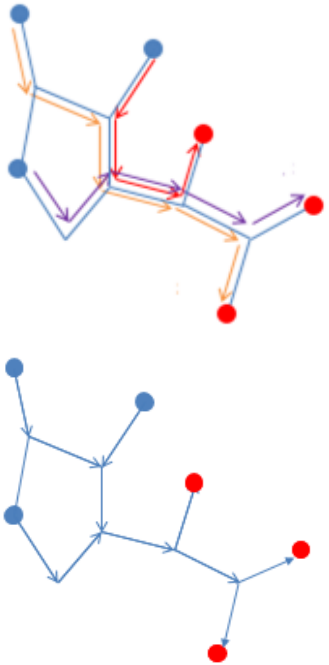
Creates a shortest- yet unscheduled- path set



Path Planning

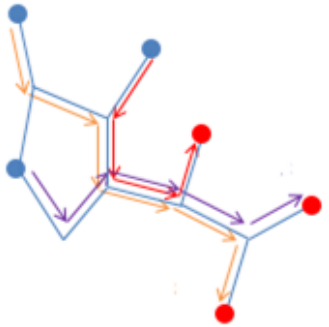
Creates a shortest- yet unscheduled- path set

Path set induces a directed acyclic graph (DAG)

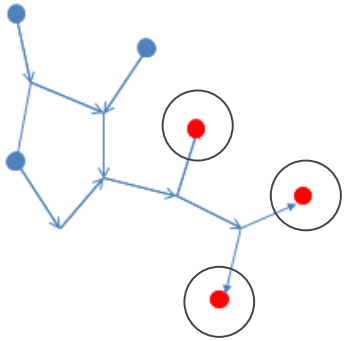


Path Planning

Creates a shortest- yet unscheduled- path set



Path set induces a directed acyclic graph (DAG)

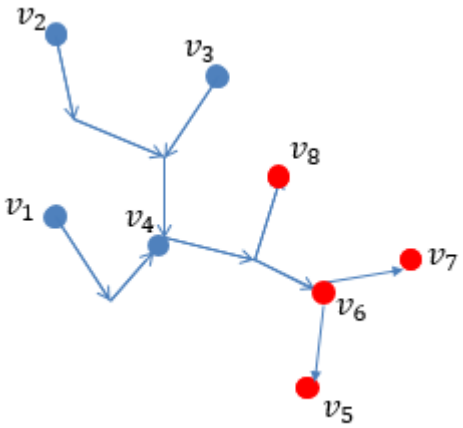


Identify **standalone** goal nodes- those goal nodes which lie in only one path.

Given priority in scheduling algorithm

Path Scheduling Process

Path Scheduling

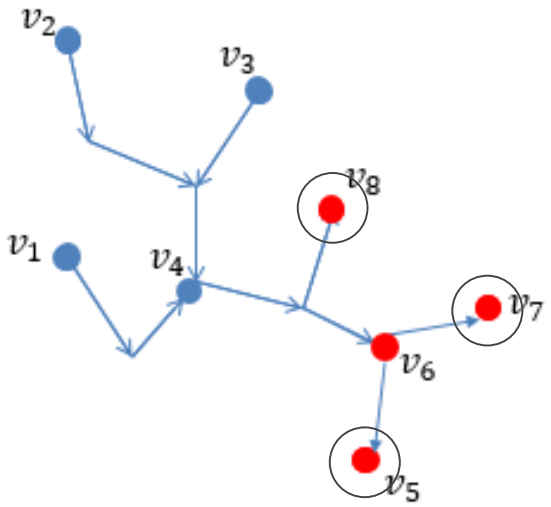


	v_5	v_6	v_7	v_8
v_1	5	4	5	4
v_2	6	5	6	5
v_3	5	4	5	4
v_4	3	2	3	2

Path Scheduling

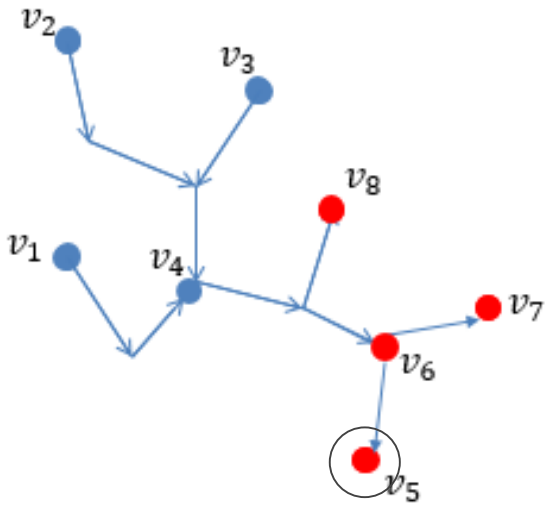
Identify standalone goal nodes

Standalones: {v_5, v_7, v_8}



Path Scheduling

WLOG, choose a standalone goal node to schedule an agent to.

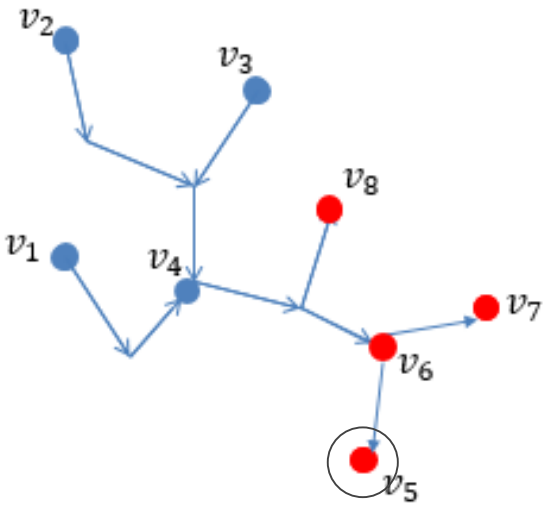


Path Scheduling

Recall:

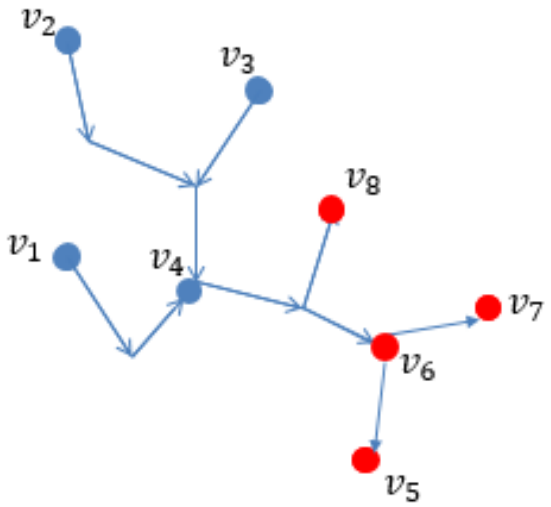
Hungarian Algorithm paired v_1 with v_5

	v_5	v_6	v_7	v_8
v_1	5	4	5	4
v_2	6	5	6	5
v_3	5	4	5	4
v_4	3	2	3	2



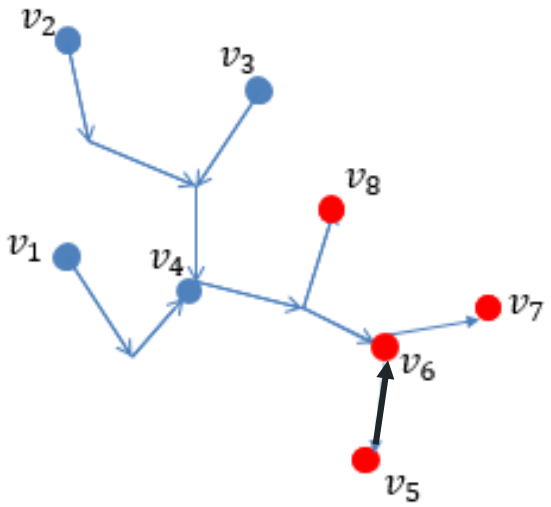
Path Scheduling

Perform “reverse” BFS on directed edges until first initial node is found.



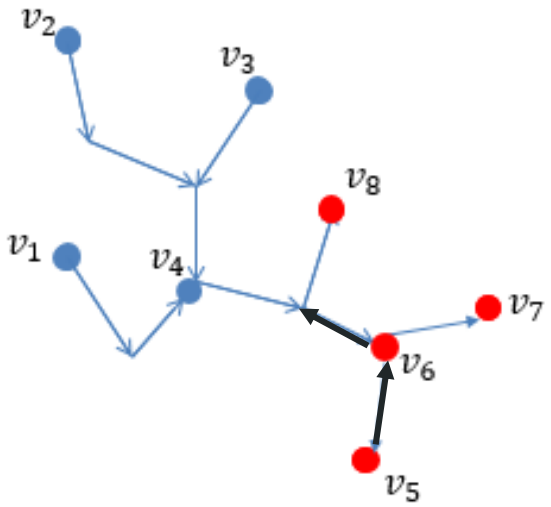
Path Scheduling

Perform “reverse” BFS on directed edges until first initial node is found.



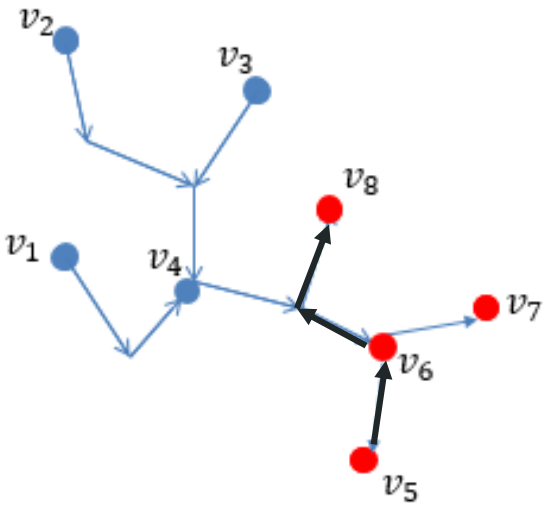
Path Scheduling

Perform “reverse” BFS on directed edges until first initial node is found.



Path Scheduling

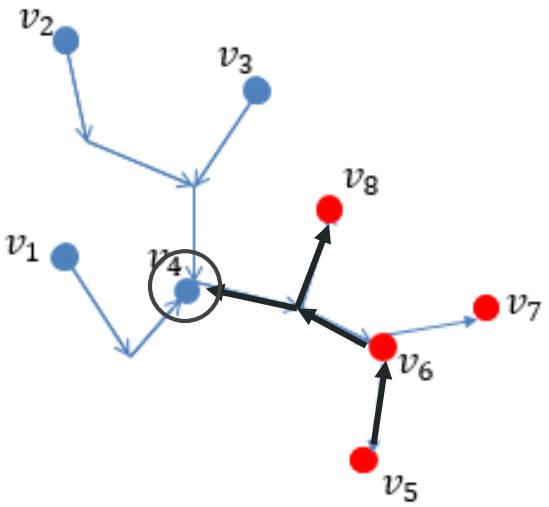
Perform “reverse” BFS on directed edges until first initial node is found.



Path Scheduling

Perform “reverse” BFS on directed edges until first initial node is found.

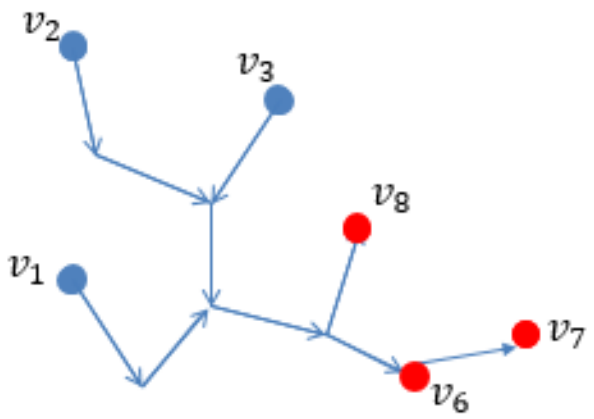
Agent present at v_4 is then assigned to move to v_5 and given start time t = 0



Path Scheduling

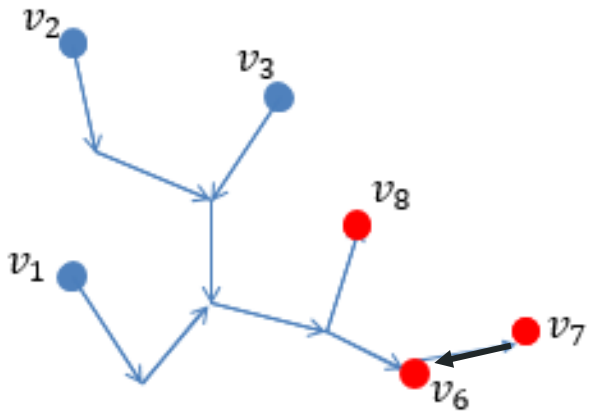
Prune the DAG so that the previously scheduled initial and goal nodes are now ignored.

Updated set of standalone goal nodes: $\{v_7, v_8\}$



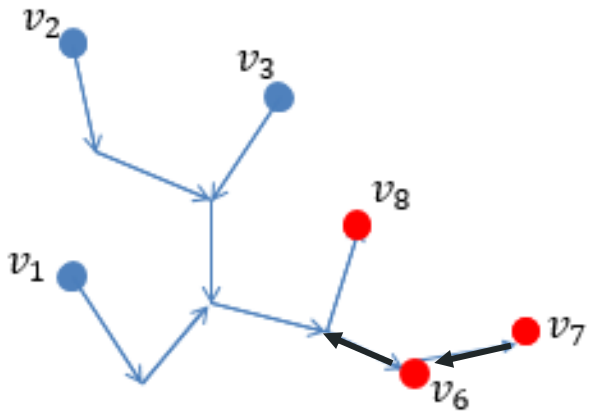
Path Scheduling

Let’s do it again!



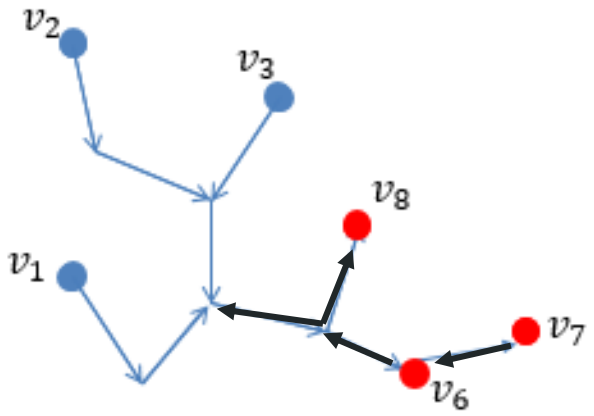
Path Scheduling

Let’s do it again!



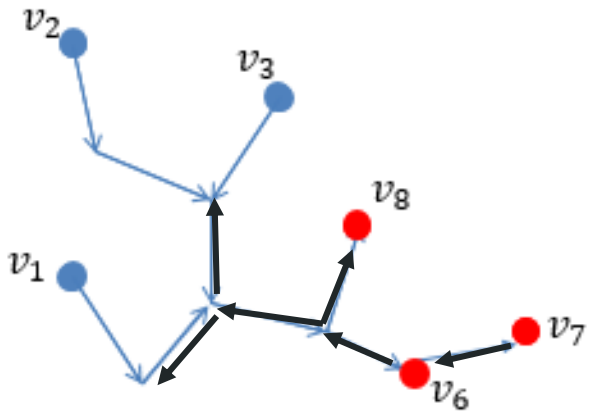
Path Scheduling

Let’s do it again!



Path Scheduling

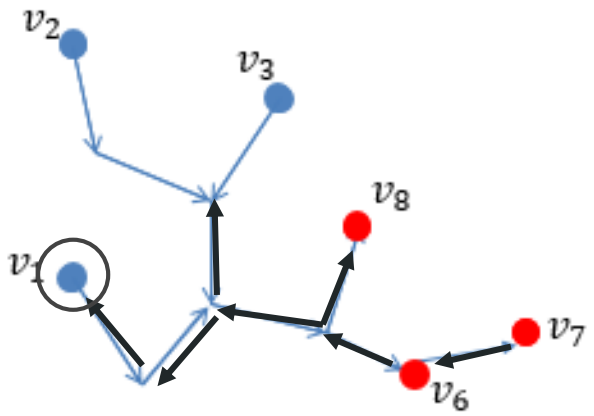
Let’s do it again!



Path Scheduling

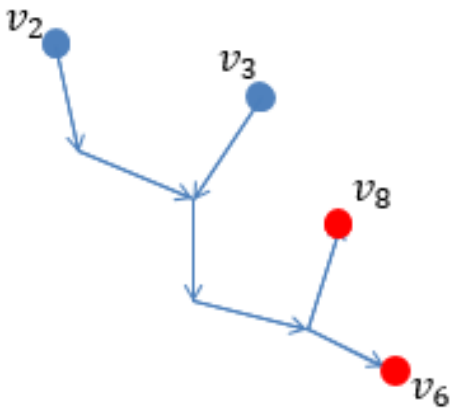
Let’s do it again!

Agent at v_1 moves to v_7 and is assigned start time t = 1

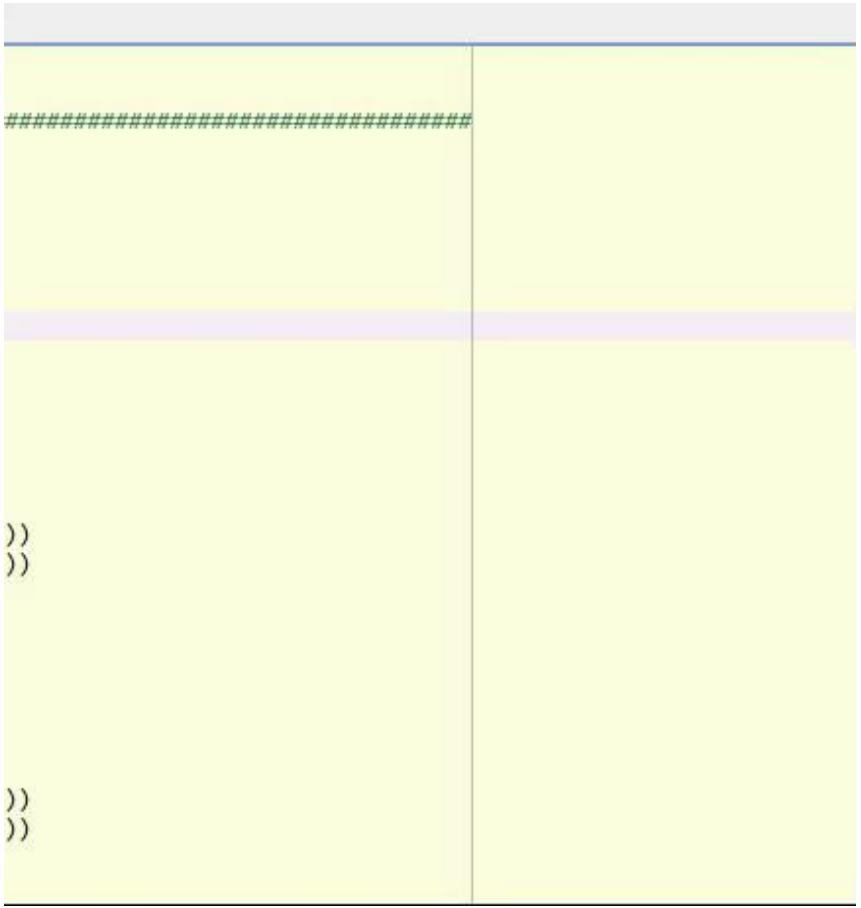


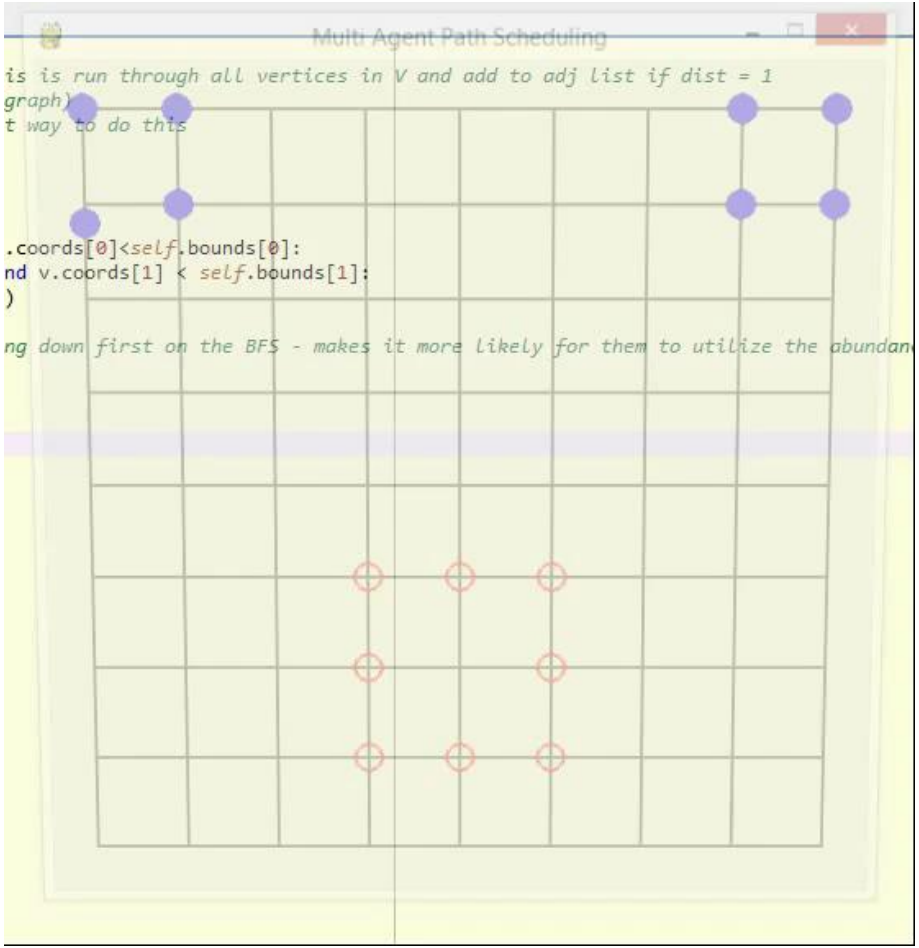
Path Scheduling

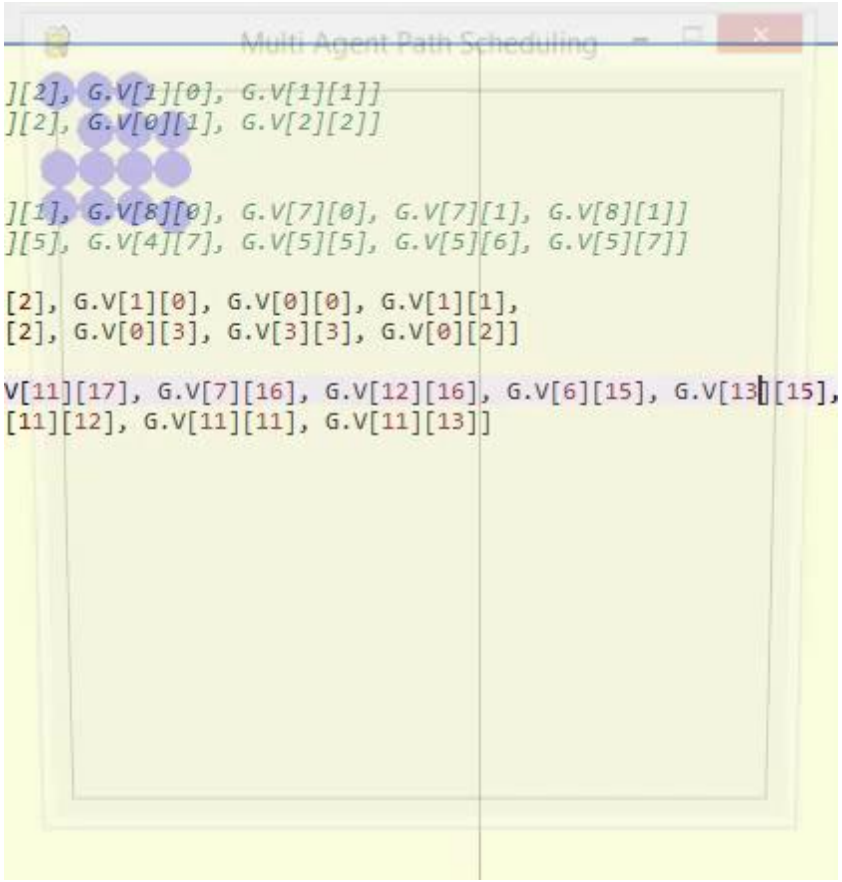
Updated standalone goal nodes: {v_6, v_8}



Simulations



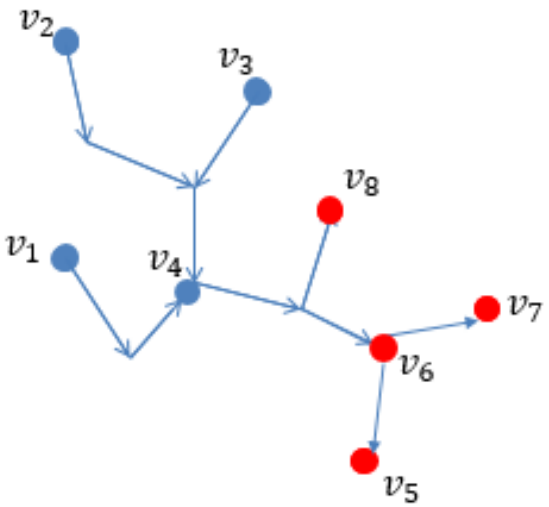

Multi-Agent Path Scheduling
is is run through all vertices in V and add to adj list if dist = 1
graph)
t way to do this
.coords[0]<self.bounds[0]:
nd v.coords[1] < self.bounds[1]:
)
ng down first on the BFS - makes it more likely for them to utilize the abundan


Multi Agent Path Scheduling
][2], G.V[1][0], G.V[1][1]]
][2], G.V[0][1], G.V[2][2]]
][1], G.V[8][0], G.V[7][0], G.V[7][1], G.V[8][1]]
][5], G.V[4][7], G.V[5][5], G.V[5][6], G.V[5][7]]
[2], G.V[1][0], G.V[0][0], G.V[1][1],
[2], G.V[0][3], G.V[3][3], G.V[0][2]]
V[11][17], G.V[7][16], G.V[12][16], G.V[6][15], G.V[13][15],
[11][12], G.V[11][11], G.V[11][13]]

Room for Improvement

Current Issues

- It is not necessary to assign each agent a different start time.



Current Issues

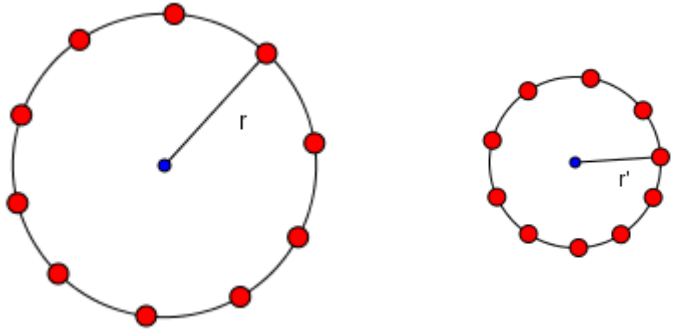
- It is not necessary to assign each agent a different start time.
- Some randomly generated initial and goal formations break my algorithm for pruning the DAG.

Current Issues

- It is not necessary to assign each agent a different start time.
- Some randomly generated initial and goal formations break my algorithm for pruning the DAG.
    - But- I have some ideas :)

Next Step: Group Strategies

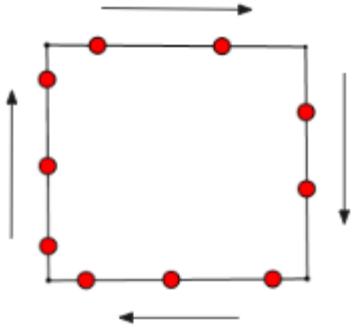
- Surround and close-in on a target



Next Step: Group Strategies

- Surround and close-in on a target

- Patrol a perimeter



Motivation

With enough data generated from implementation of specific tasks, we may train an autonomous model.

References & Acknowledgements

Special thanks to:

Dr. Jingjin Yu, Mentor
National Science Foundation, funding provided through grant ISS-1734419

Reference:

Yu, J., & LaValle, S.M. (2012). Distance Optimal Formation Control on Graphs with a Tight Convergence Time Guarantee, presented at 51st IEEE Conference on Decision and Control, Maui, HA, December 10-13.