Multi-Agent Path Finding with Reinforcement Learning

James Ellis

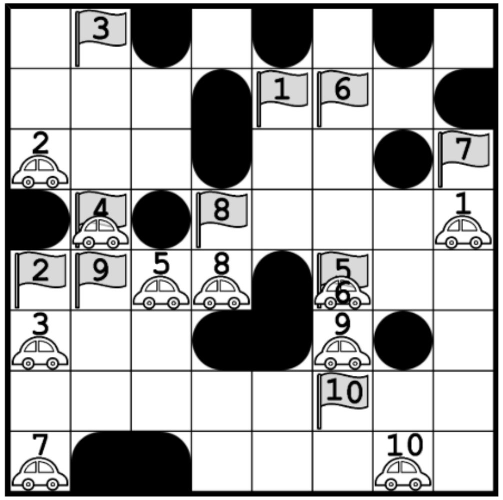
Supervisor: Prof. HA Engelbrecht

13 November 2020

Multi-Agent Path Finding

- Applications in robot navigation, traffic control and gaming.
- Multiple agents navigate to their goal locations.
  - Avoid collisions.
  - Minimise the sum of all agent path lenghts:

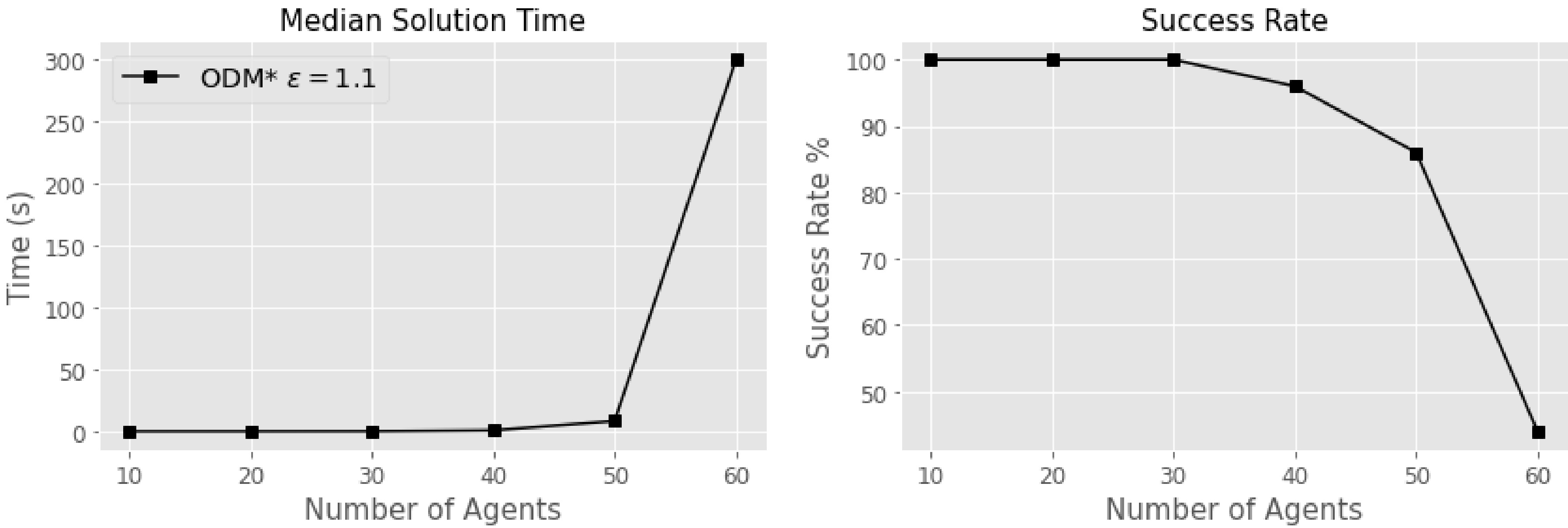
Example [1]:



[1]: T. Standley, “Finding optimal solutions to cooperative pathfinding problems,” Proceedings of the National Conference on Artificial Intelligence, vol. 1, pp. 173–178, 2010.

Multi-Agent Path Finding

- Reasons for using reinforcement learing:
  - 1. Centralised MAPF planners scale poorly to large environments with many agents.



Multi-Agent Path Finding

- Reasons for using reinforcement learing:

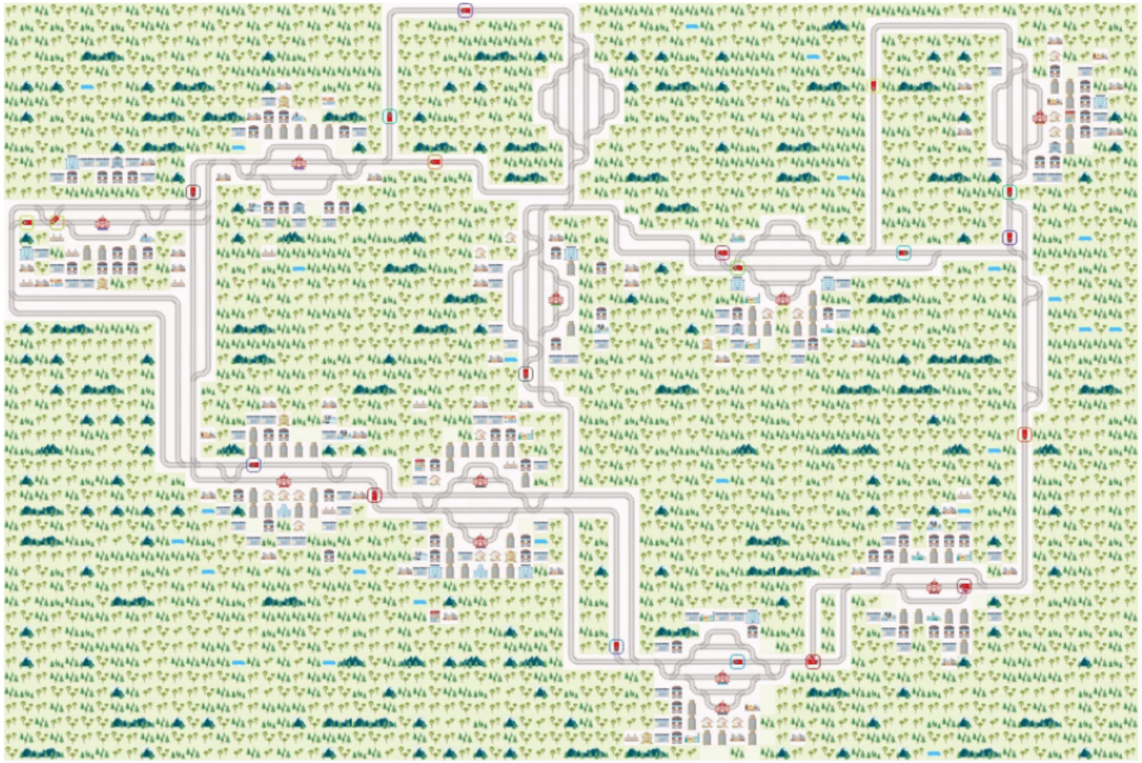
  2. Real time execution.
     - Centralised planners not suitable for scenarios which require **re-planning**.

  3. Reinforcement learning does not require a model of the environment.

Multi-Agent Path Finding

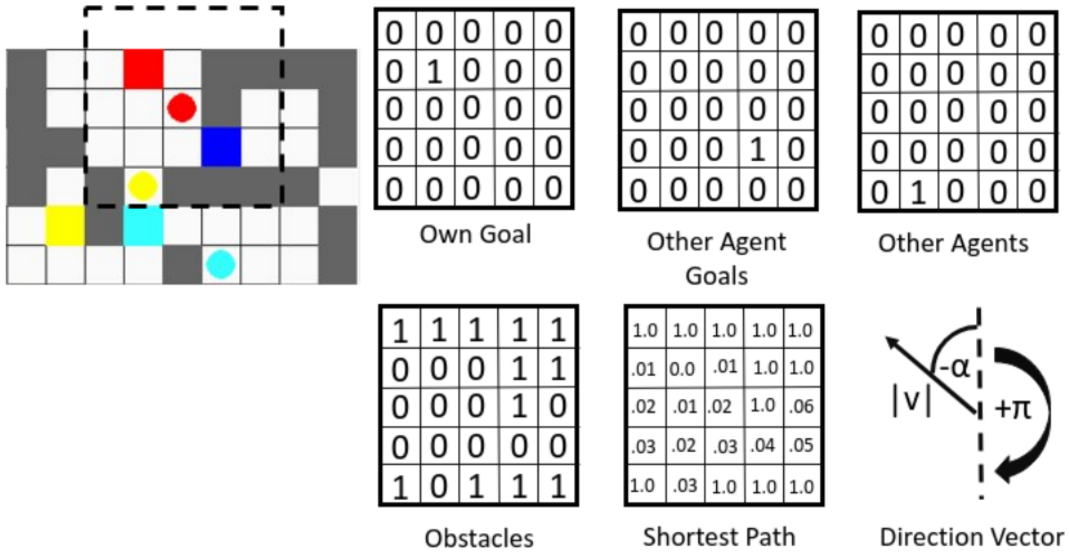
- Example:

Flatland Challenge
Multi Agent Reinforcement Learning on Trains

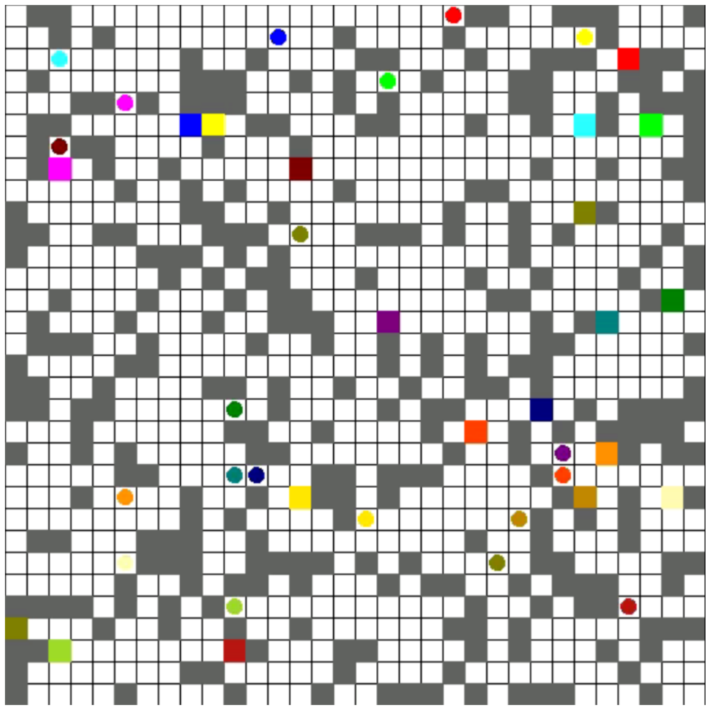


From: https://www.aicrowd.com/challenges/flatland-challenge

Environmnet



Observation Space



32x32 Gridworld

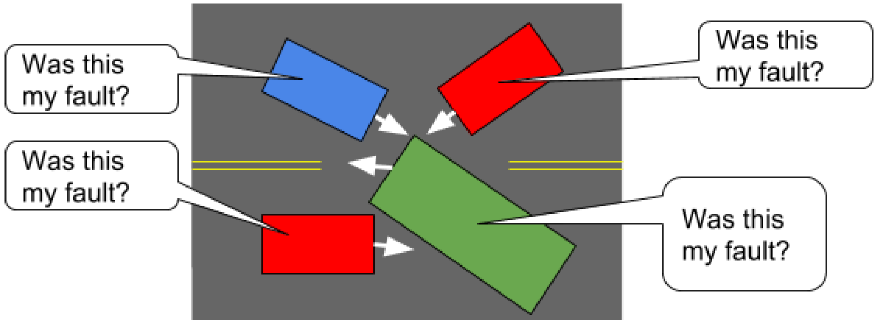
Multi-Agent Reinforcement Learning

- Single-Agent RL
  - Only one learner
- Multi-Agent RL (MARL)
  - Many learners
  - Interacting agents (environment dynamics depends on all agent actions).
  - Agent Autonomy

Multi-Agent Reinforcement Learning

MARL Challenges

- Scalability: Exponential increase in state-action space with increasing number of agents.
- Non-Stationarity: Best action depends on other agent actions. All agents are learning and changing their policies.
- Credit assignment problem

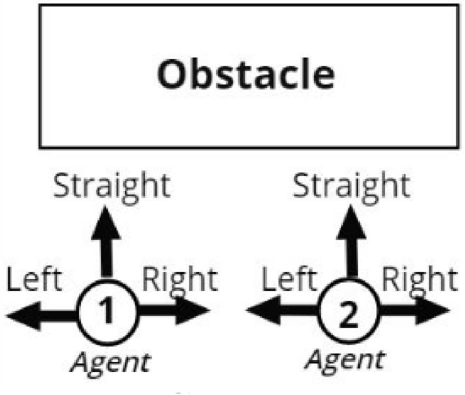


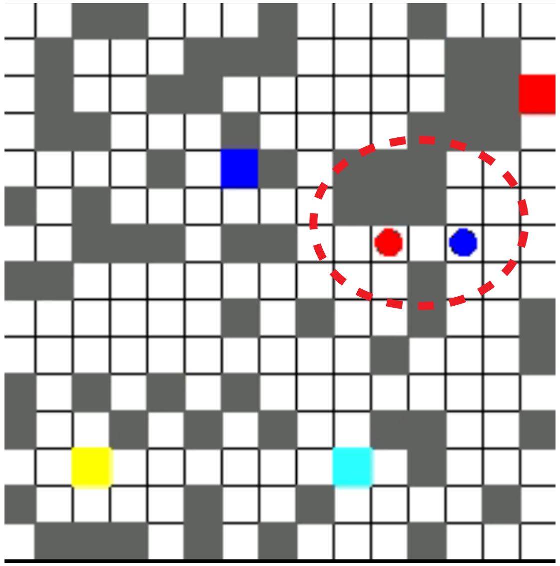
From: https://bair.berkeley.edu/blog/2018/12/12/rllib/

Multi-Agent Reinforcement Learning

MARL Challenges (...continued)

- Coordination problem





Approach 1

- Train RL agents using a purely reinforcement learning approach.
  - No handcrafted heuristics or supervision.

Algorithms selected:

- Independent learners
  - Proximal Policy Optimisation **(PPO)**
- Centralised learning with decentralised execution
  - Actor-Attention-Critic for Multi-Agent Reinforcement Learning **(MAAC)**
- Differentiable communication
  - Individualised Controlled Continuous Communication Model **(IC3Net)**

Results

Comparisons on fully observable 7x7 gridworlds.

- **IC3Net:** Did not learn to communicate in the MAPF environment.
- **MAAC:** Surprisingly, MAAC did not perform well on global rewards:
  - Agents obtain a shared global reward when **all** agents reach their goals:

Algorithm	Episode Length	Agent Collisions	Obstacle Collisions	Per Agent Success Rate	Task Success Rate
PPO	**18.58±2.2**	0.55±0.5	1.14±0.6	**0.78±0.1**	**0.48±0.1**
MAAC	26.0±0.0	**0.08±0.4**	**0.57±0.9**	0.01±0.0	0.0±0.0

- **PPO:** Using curriculum learning policies can be trained to have performance comparable to MAAC.
- RL in partially observable environments struggle to scale to larger environment sizes

Approach 2

- In [2], agents are trained using both RL and imitation learning. They achieve very good results by using several heuristics during training, as well as imitation learning.
- Approach used in [2]:
  - For each episode, 50% change of training with either RL or behaviour cloning (immitation learning).
  - A blocking penalty is introduced to discourage agents from blocking one another.
  - Invalid action are removed during training.
  - Environment sizes and obstacle densities are sampled so that agents are trained on difficult environments more often.

[2]: G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset, “PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning,” [Online]. Available: http://arxiv.org/abs/1809.03531

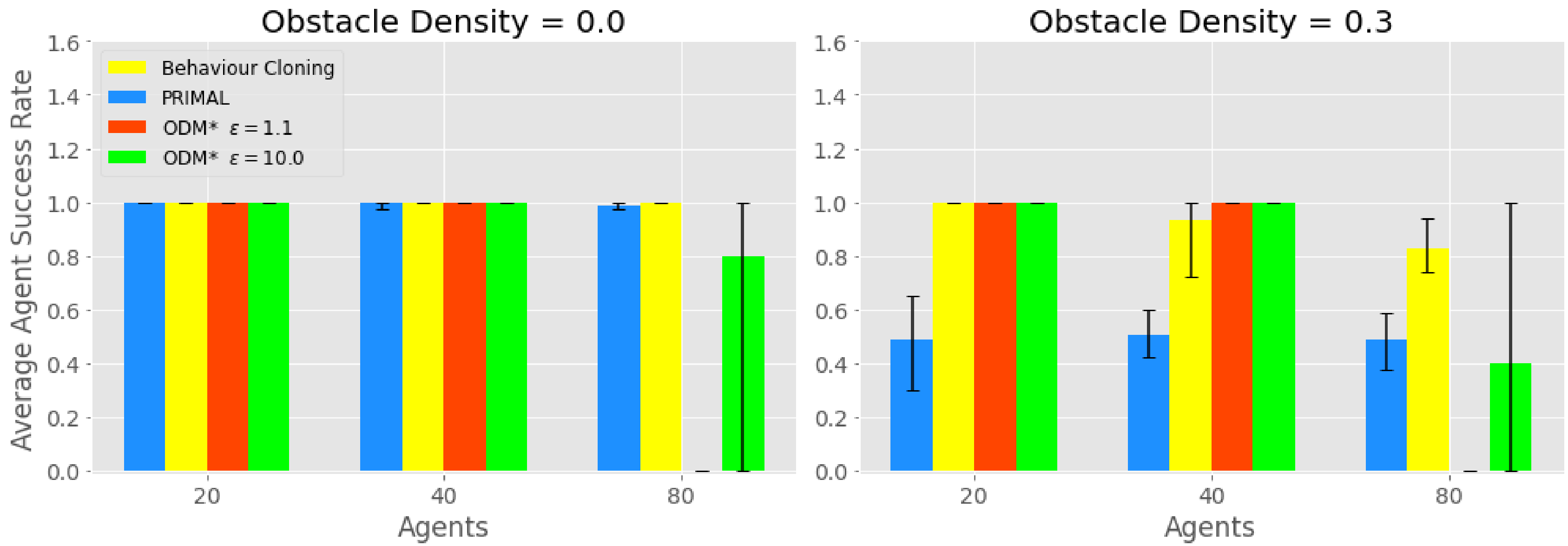
Approach 2

- Would like to investigate the effect of these heuristics on performance.
  - Ablation study on PRIMAL.
  - How does behaviour cloning perform on its own?
  - Compare with baseline ODM*.

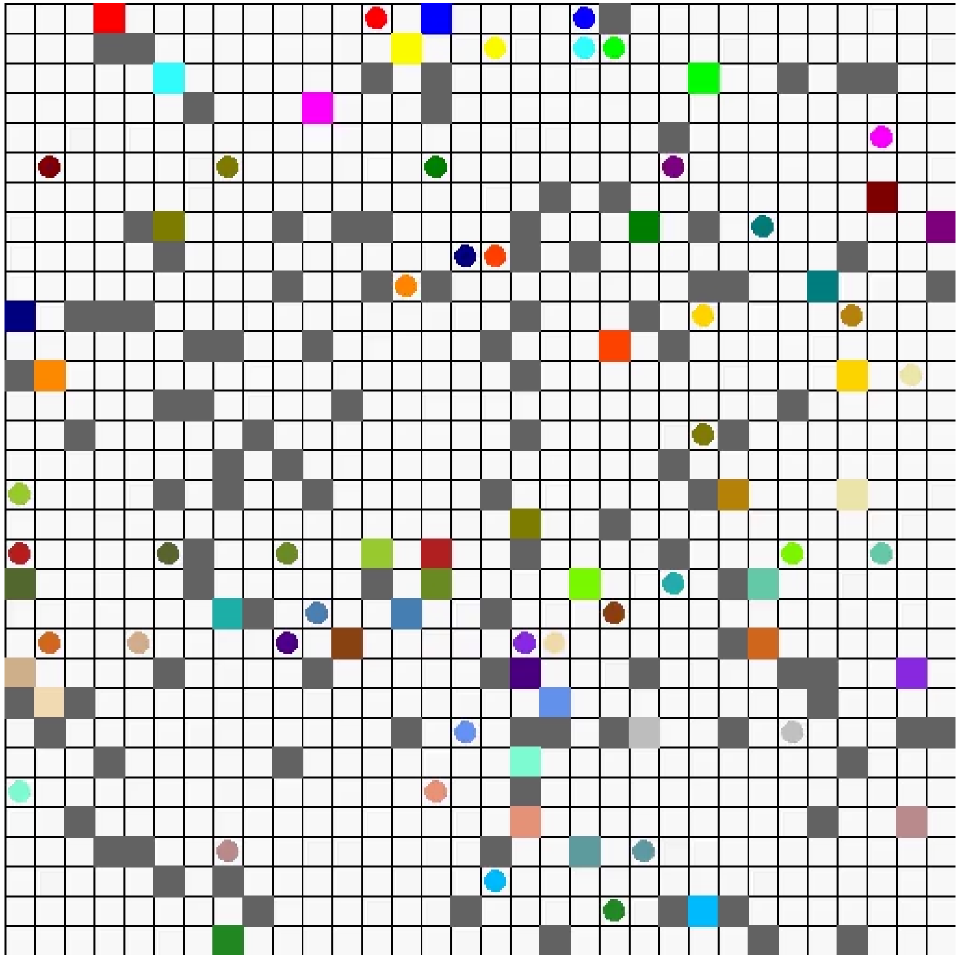
Comparison with ODM*

- ODM* limited to 5 minutes execution time.
- Our implementation of PRIMAL did not include blocking penalties.

32x32 Environment Size



Approach 2



Conclusion

- In larger environmnets with many agents:
  - Deep learning / RL approaches outperform ODM*.
- In smaller environments:
  - ODM* outperforms deep learning / RL approaches.
- Imitation learning becomes necessary when scaling to larger environmnet sizes.
- Using a MARL approach (MAAC) has no benefit over a single agent approach (PPO) for this environment.
- The MAPF environment is not suitable for learning communication with RL.

Thank You • Dankie • Enkosi