Motivation
How to navigate autonomous vehicles in a partially observable environment with uncertainty?

Our Contributions
- We formulated a zone based path finding problem (ZBPF)
- Under uncertainty
- Partial Observability
- We presented a novel formulation of policy optimization
- Based on difference-of-convex functions (DC) programming
- We developed a simulator for ZBPF using Unity3D game engine

Problem Formulation
- A graph $G = (V, E)$
- Each zone has a capacity
- A set of agents with sources and destinations
- Crossing two zones requires minimum and maximum time

Objective: Minimize travel time and Congestion

Our Simulator
- Simulator: Unity 3D game engine
- The spheres are the zones, and cubes are the agents.
- Highlighted zone: there is congestion
- Highlighted agent: reached destination

Agent’s Decision Model
- State
  - State is a tuple of three components: $(current\ zone, next\ zone, remaining\ time)$
- Hybrid Action (discrete and continuous)
  - Discrete action: which zone to go
  - Continuous action: with what speed
- An example
  - An agent is newly arrived in zone 1 at time $t$.
  - We have $s_t = (x_t, \phi, \psi)$, $a_t = x_0, 0.5$.
  - The realized time (say, 2) is sampled from a binomial distribution.
  - We have $t_{t+1} = (x_2, x_3, 1), a_{t+1} = move$.

Transition Function (exponential family)
- $p(s'|s, a, s') = f(s', a', s') \exp\{\phi(s, s') - \mathcal{A}(v')\}$
- Uncertainty movement can be modeled
- Applicable in several applications
- Partial Observation
  - Agents can only observe local neighboring zones
- Reward Function
  - A positive reward for reaching the goal
  - A negative reward for time step or congestion

Policy Optimization in DC Form
- Original Objective Function
  $$J(\pi, \mu) = \sum p(c)G(c)$$
- Why DC programming?
  - The objective is non-linear and non-convex. Direct optimization is difficult.
  - Nonlinear solvers cannot scale to large number of agents.

DC Programming
- $\min\{u(x) - v(x) : x \in \Omega\}$
- $u(x)$ and $v(x)$ are convex functions
- Concave-Convex Procedure (CCP) can solve it iteratively.
- $x_{k+1} = \arg\min\{u(x) - x^T Fr(x) : x \in \Omega\}$

Objective Function in CCP
- $G = \{ x(\mu, x) \sum_{(v', y')} x(\mu, x') \{ \sum_{(y', z')} x(\mu, y') \} - \sum_{v', y'} x(\mu, x') \}$

Experimental Results
- 2D Open Grids
  - Settings
    - 4x4 grid, 2 agents to 10x10 grid, 10 agents
    - Starting and goal locations were the top and bottom rows.
    - The capacity of each zone was sampled uniformly from a range e.g., $[1, 4]$
    - $t_{\min} = 1, t_{\max} = 5$ (Binomial distribution as travel time dist.)
  - Comparison against
    - DCRL (our approach); VPG (vanilla Policy Gradient);
    - SP (each agent follows shortest path)
    - HA (multiagent Q-learning based for hybrid action space)
  - Results:
    - Our approach DCRL provides much better SOC quality than can minimize congestion
    - VPG suffers due to lack of effective credit assignment
    - HA isn’t able to handle a large number of agents

3D Maps ("SmallCube" and "TwoFloor")
- Settings
  - 10 and 20 agents for SmallCube and TwoFloor
  - Capacity of was uniformly sampled from $[1, 3]$ and $[1, 4]$.

Acknowledgements
This research is supported by the Ministry of Education, Singapore under its Academic Research Fund Tier 2 (MOE2018-T2-1-179).

Reinforcement Learning for Zone Based Multiagent Pathfinding
Under Uncertainty
Jiajing Ling, Tarun Gupta, Akshat Kumar
(jjling2018, akshatkumar)@smu.edu.sg, tarun.gupta@cs.ox.ac.uk