Lucia Cipolina-Kun
Ignacio Carlucho
Kalesha Bullard

University of BRISTOL
HERIOT WATT UNIVERSITY
Google DeepMind

Adaptive Dynamic Coalition Structure Generation

Motivation

- **How to form coalitions of agents in a changing environment?**
- Examples: ridesharing, wireless communications networks, ad-hoc teamwork, etc.
- Users/ agents are added or removed from the environment dynamically.

How do we form coalitions in a changing environment?

- Brute force requires 2^{N_t} computations every time the environment changes.
- Multi-agent approaches introduce non-stationarity of the environment

Our Approach

- Method based on Deep RL for dynamic coalition games. Solved it as a single-agent problem. Each agent is trained using a deal-or-no-deal game
- Training with contextual data (such as distance matrix) achieves generalization.
- The **state** space is the 2^N possible coalitions at any time. **Action** selection policy is to be able to identify the right coalition faster than exponential. The **Reward** is linked to the value of the selected coalition.
- **Objective** is to max Social Optimum. Reward encourages selection of good coalitions.

The Deal-or-No-Deal Game

A Method to Obtain **Socially Optimal** Coalition Structures in a **Dynamic Environment**

	agent state		state proposal		agent state		state proposal		agent state
state	{A}	⇨	{A,B,C}	accepted	{A,B,C}	⇨	{A,B}	rejected	{A,B,C}
reward	V(A) – V(0)				V(ABC) – V(A)				- V(A,B) + V(ABC)

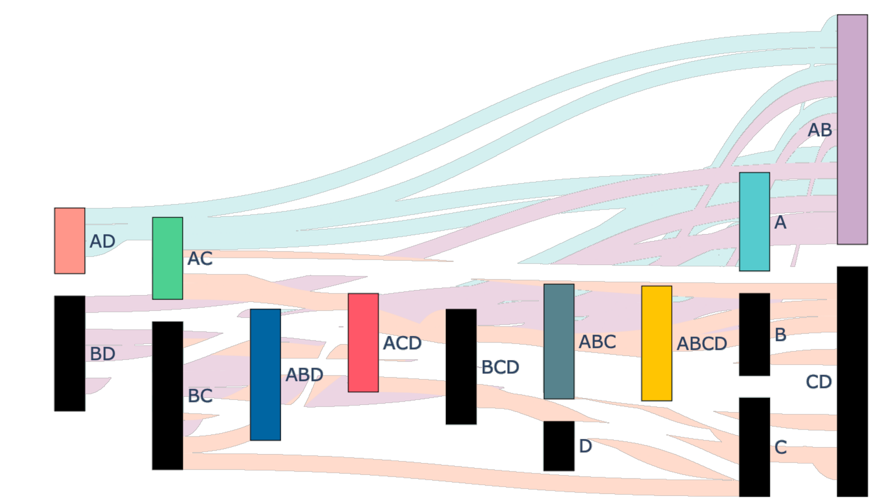
	agent state		state proposal		agent state		state proposal		agent state
state	{B}	⇨	{A,B,C}	accepted	{A,B,C}	⇨	{A,B}	rejected	{A,B,C}
reward	V(A) – V(0)				V(ABC) – V(B)				- V(A,B) + V(ABC)

	agent state		state proposal		agent state		state proposal		agent state
state	{C}	⇨	{A,B,C}	accepted	{A,B,C}	⇨	{A,C}	rejected	{A,B,C}
reward	V(A) – V(0)				V(ABC) – V(C)				- V(A,B) + V(ABC)

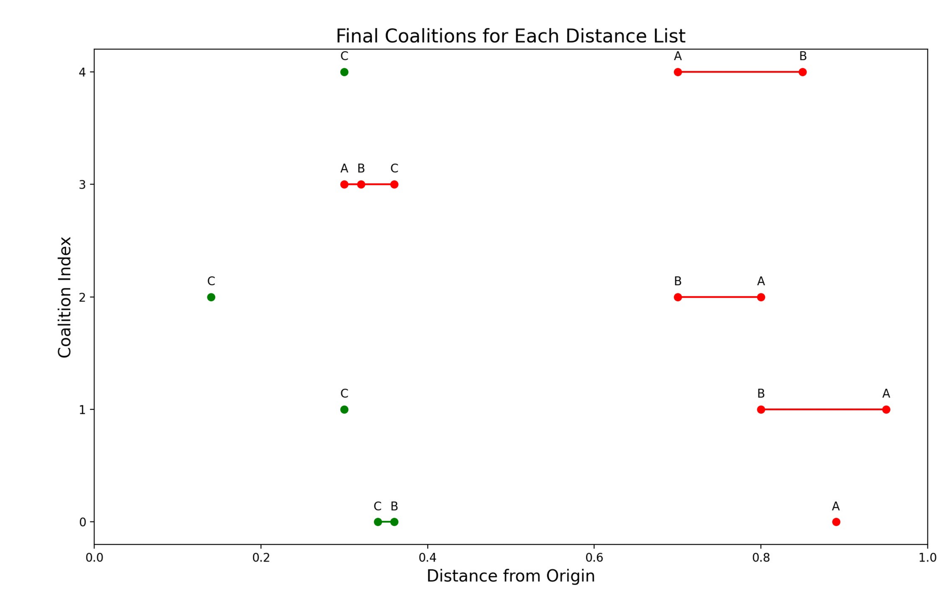
Time

Results

Coalition dynamics



Generalization – Ridesharing game



$$accuracy = \frac{\#\ correct\ actions}{\#\ coalitions\ proposed}$$

Accuracy of each agent's policy across tasks (1 seed). The dashed line represents the accuracy of a random policy