Putting RL to Work

B. Ravindran

Reconfigurable and Intelligent Systems Engineering (RISE) Group
Department of Computer Science and Engineering

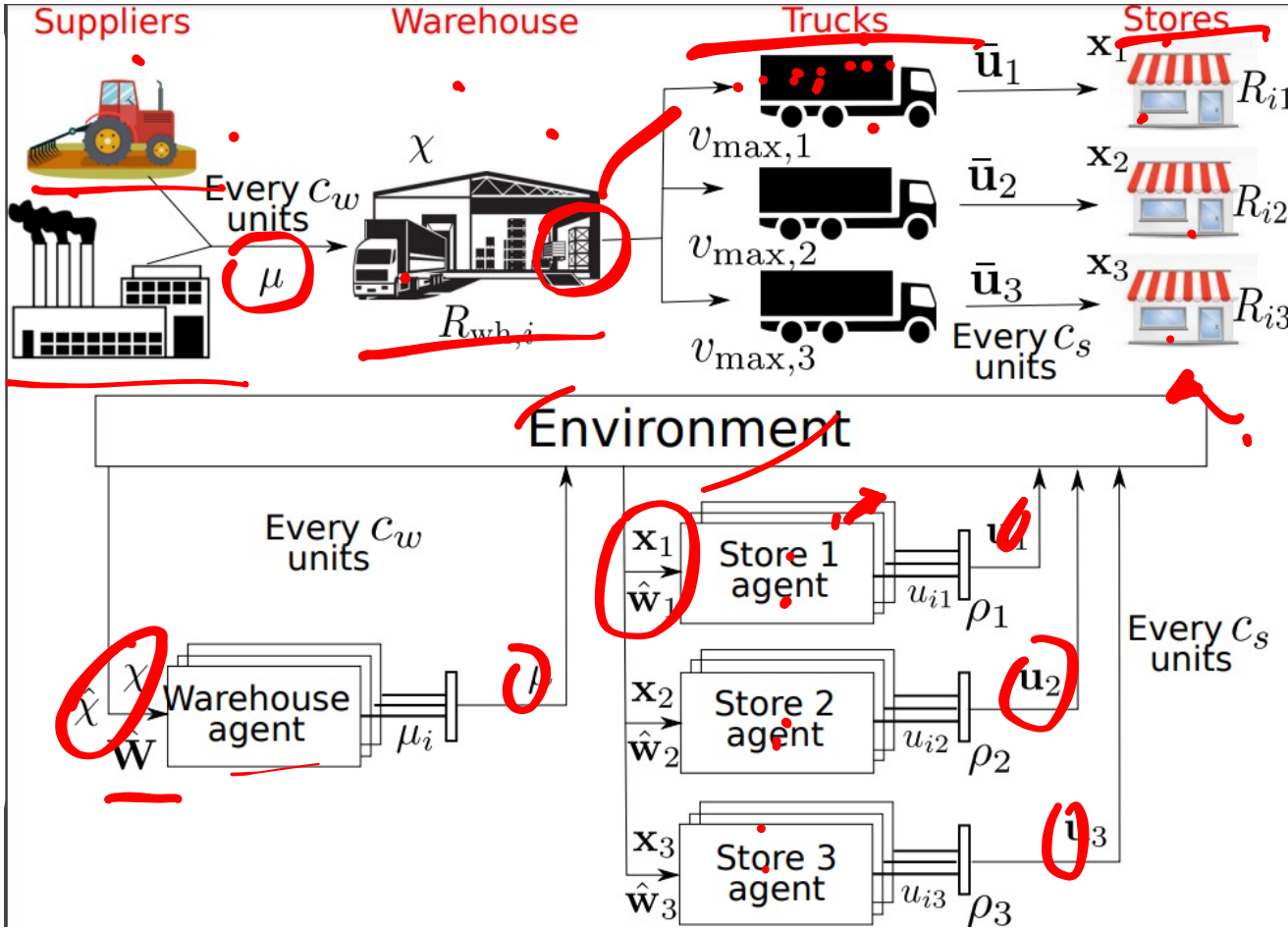
Robert Bosch Centre for Data Science and Artificial Intelligence (RBC-DSAI)
Mindtree Faculty Fellow
TCS Affiliate Faculty
Indian Institute of Technology Madras

Outline

- Supply Chain Management
- Network Structure Discovery
- Green Security Games
- Bidding in Power Markets

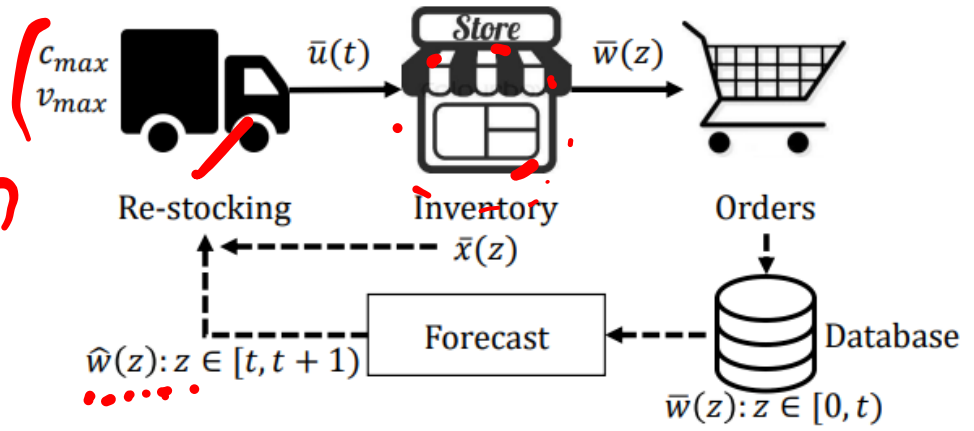
Supply chain: Multi-agent inventory management

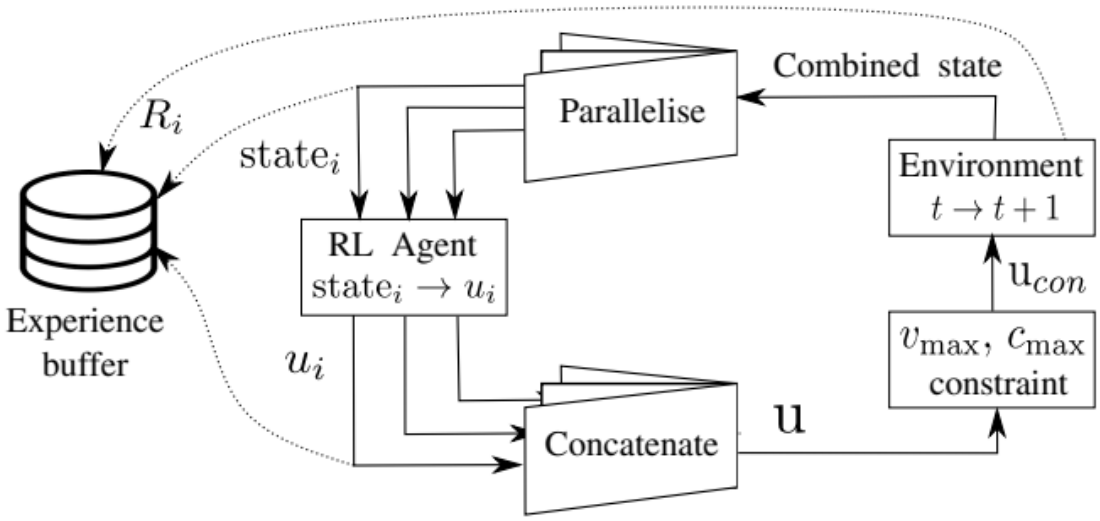
- Hierarchical flow of products
- Challenges:
  - Different time scales at different levels of the tree
  - Capacity constraints
  - Lead times
  - Large number of products
  - Selfish objectives



Step 1: Deconstructing just the single store problem

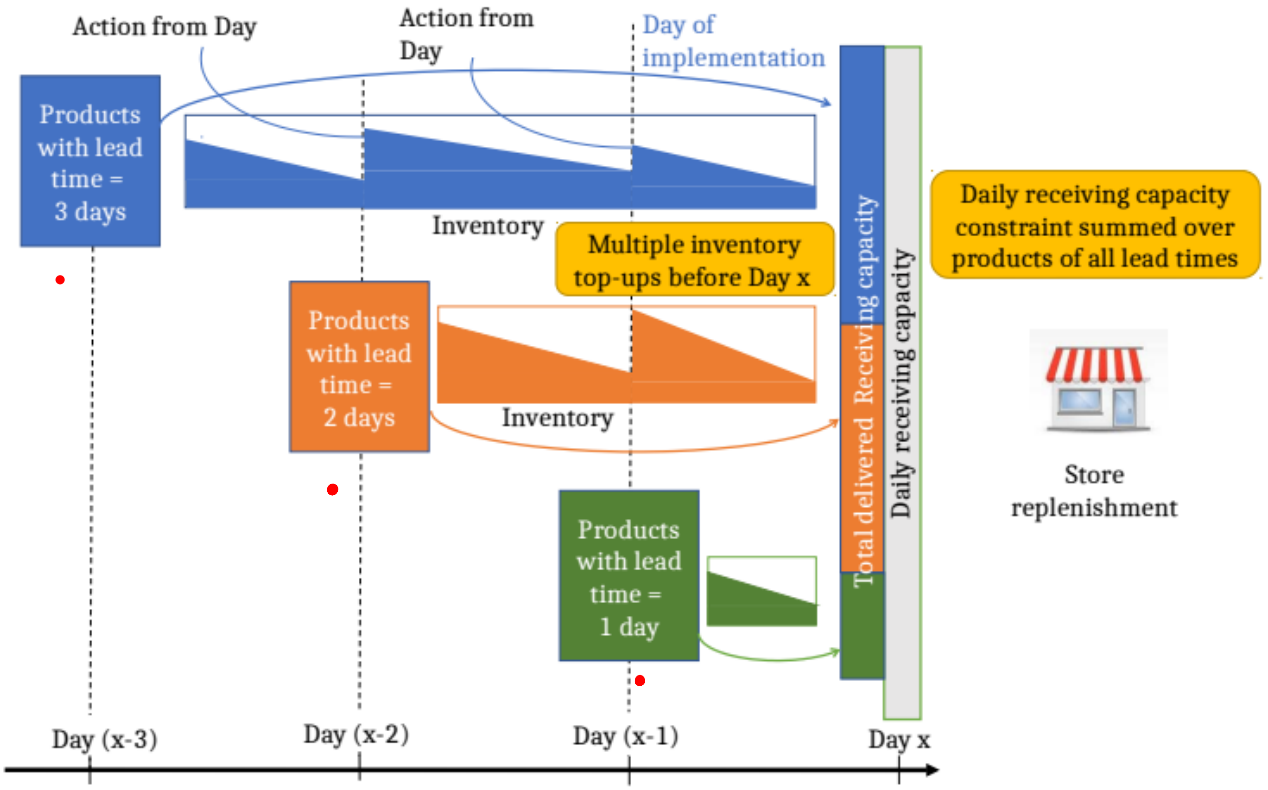
- Single product meta-model
- Input structure
  - Inventory & forecast
  - Product meta-data
  - Strain on system capacity
- Output
  - Replenishment quantity of single product
- Reward
  - Business importance: availability, wastage, etc.
  - Penalty for capacity break
- Parallel computation
  - Fast inference
  - Scalable to arbitrary number of products





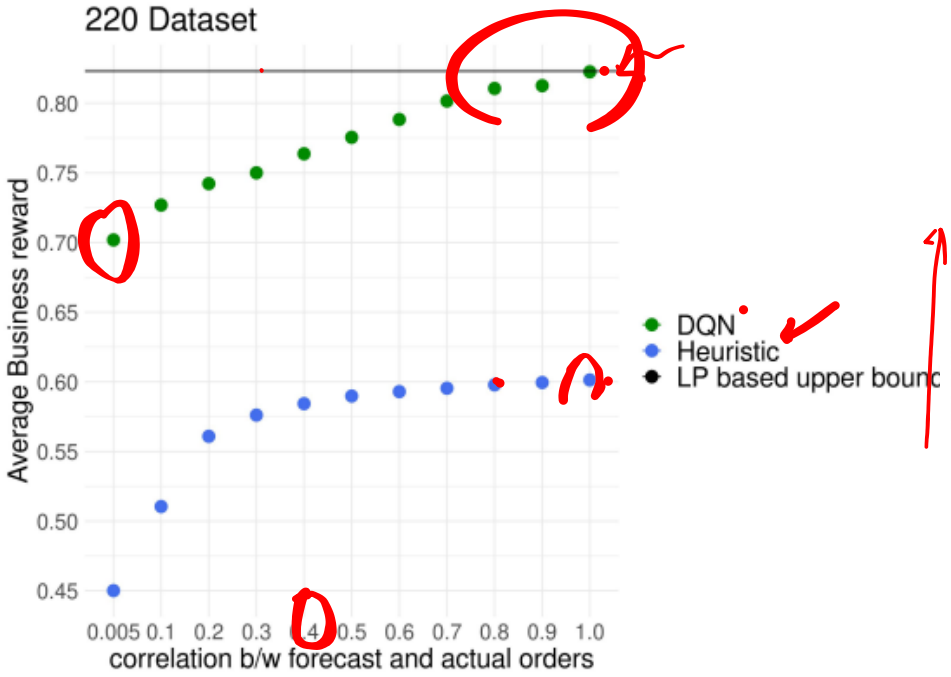
Step 2: Accounting for lead times

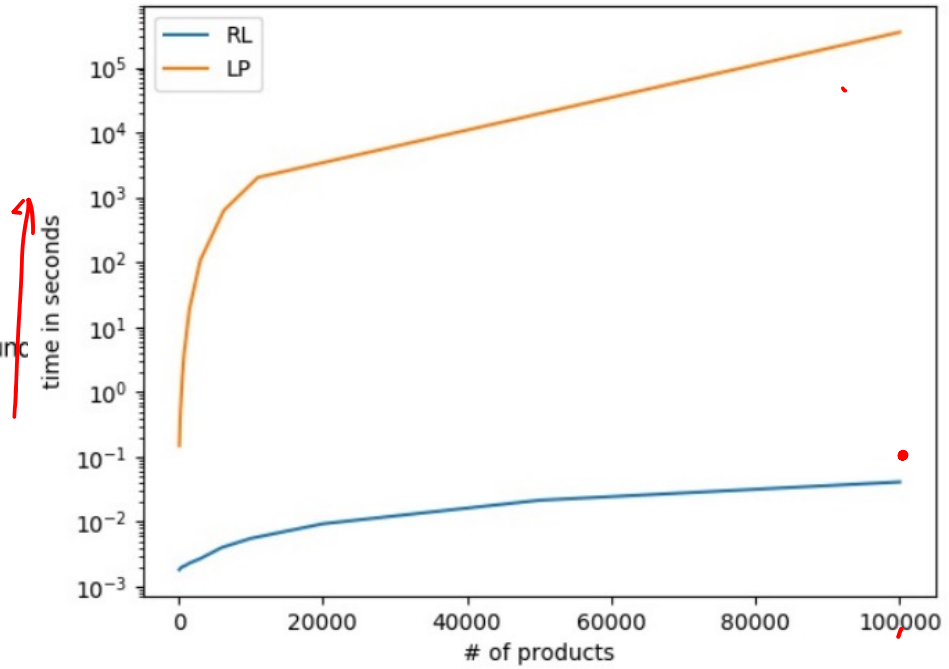
- Need to retain Markov structure despite mismatch
- Separate agents for each lead time value
- Rollouts for managing capacity



Step 1 & 2: Results

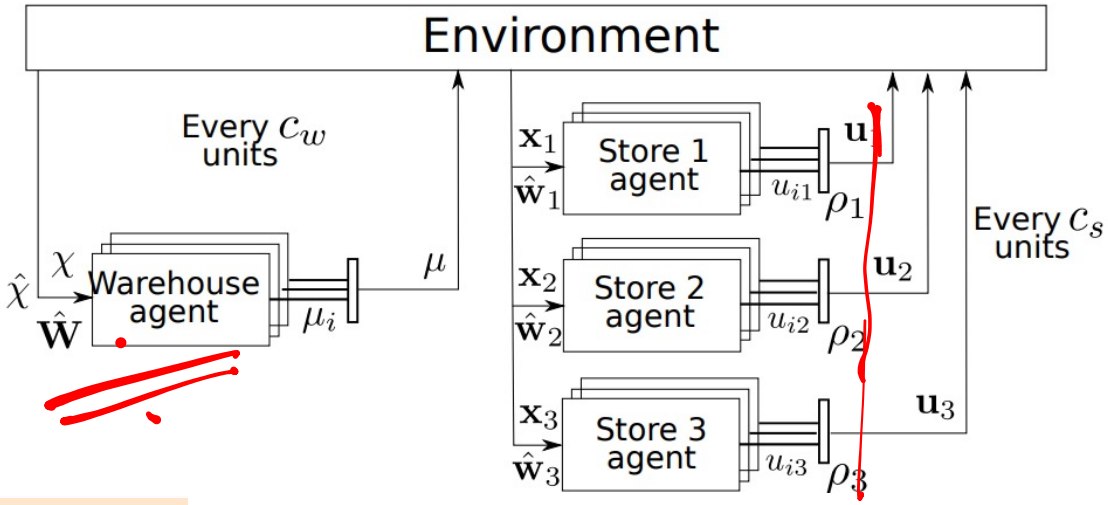
- Close to optimal performance with orders of magnitude lower computational time
- "Scalable Multi-Product Inventory Control with Lead Time Constraints using Reinforcement Learning", Meisheri, H; Sultana, N; Baranwal, M; Baniwal, V; Nath, S; Verma, S; Ravindran, B; Khadilkar, H; Neural Computing and Applications, May 2021

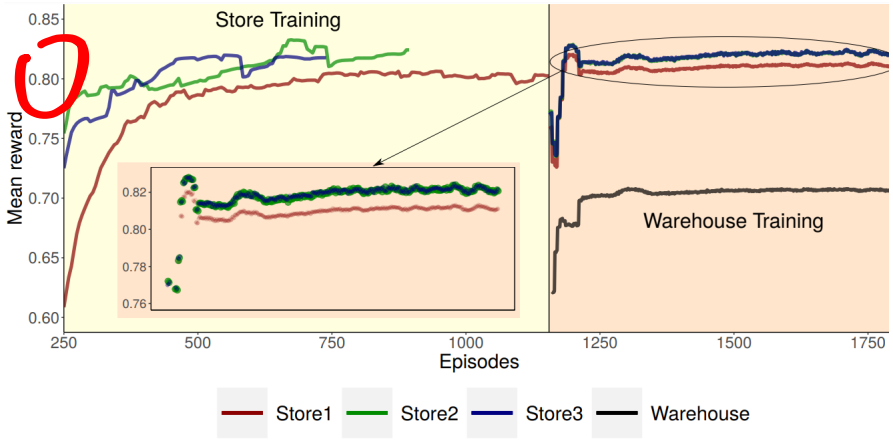


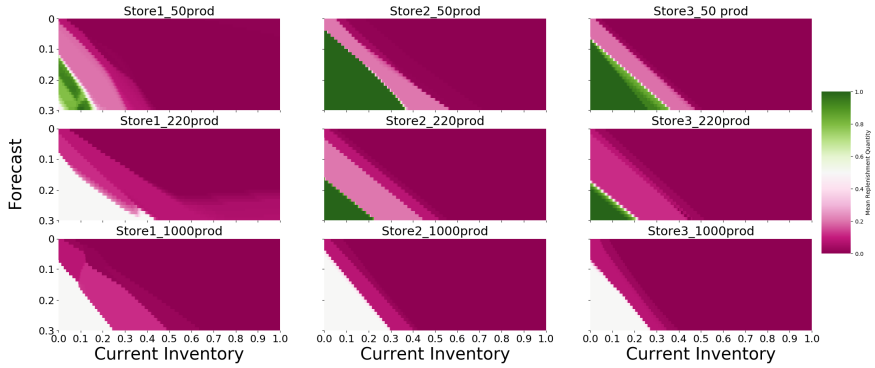


Step 3: Multi-agent problem

- Two-phased training approach for stable training
- Simple, explainable policies
- Still scalable and transferable
- Under review at CIKM







Influence Maximization in Unknown Social Networks: Learning Policies for Effective Graph Sampling

Harshavardhan Kamarthi (IITM->Georgia Tech), Priyesh Vijayan (IITM-> McGill/Mila), Bryan Wilder (Harvard-> CMU), B. Ravindran (IITM), Milind Tambe (Harvard/Google)

Best Paper Runner-up AAMAS 2020.

http://arxiv.org/abs/1907.11625

Influence Maximisation Problem

- Objective: Pick influential nodes from a social network as peer leader to help disseminate information to maximum number of nodes in the network

- More specifically, given a graph $G = (V, E)$, select K nodes to activate such that the information flows across the edges from V results in maximum number of nodes receiving information.

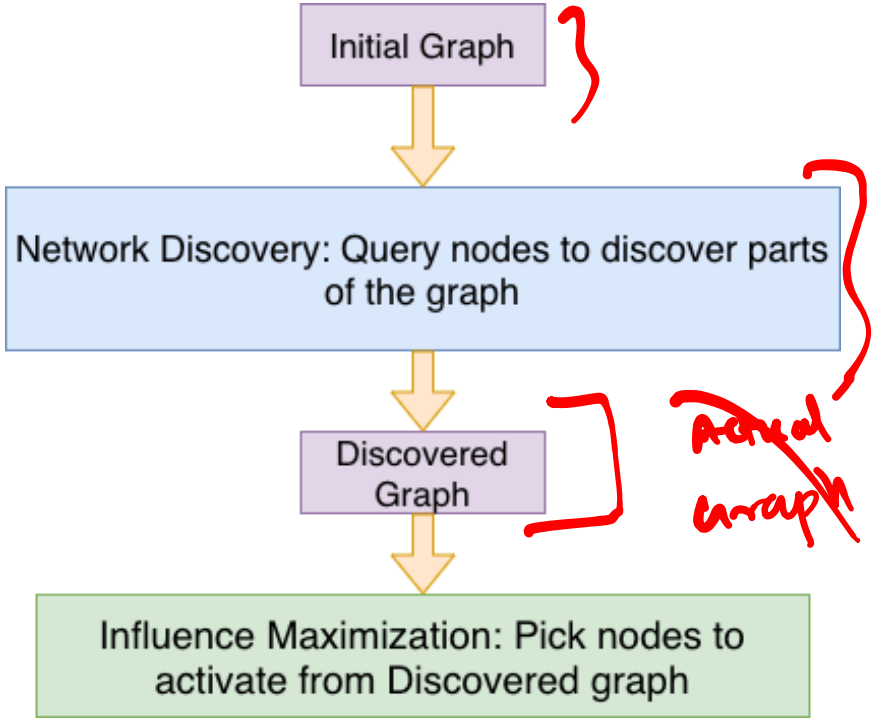
- Have found applications in substance abuse [VP07] interventions, micro-finance adoption [Ban+13], HIV prevention [Yad+18; Wil+18a] , etc.

- Previous works use a greedy algorithm [KKT05] to minimise computational cost of simulating the influence spread over entire networks.

Influencing real world social networks

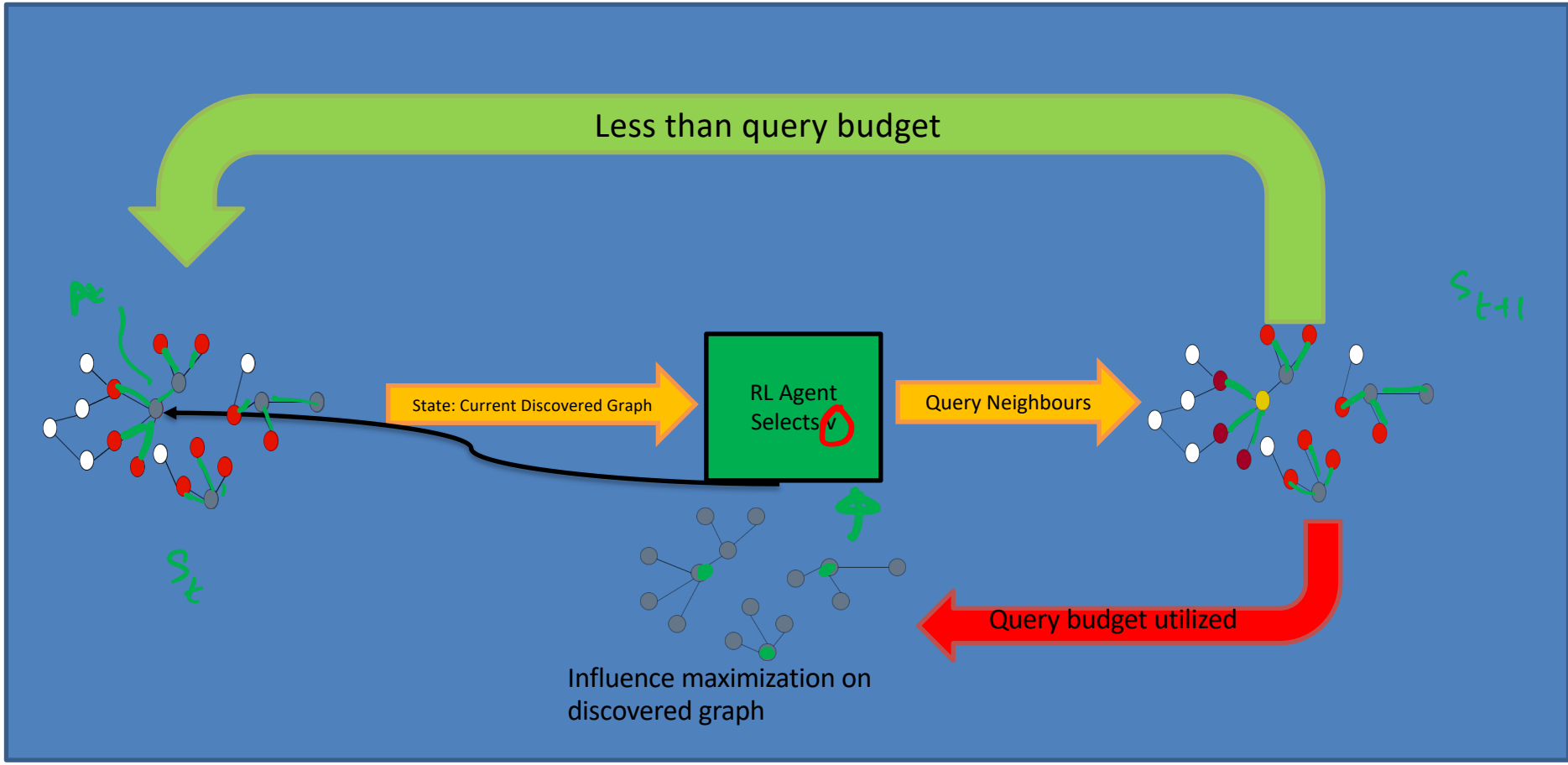
- We have little data on the structure of social network
- Cost of collecting data in terms of time and effort is high
  - Typically operate with a budget of queries

Network Discovery for influence maximisation

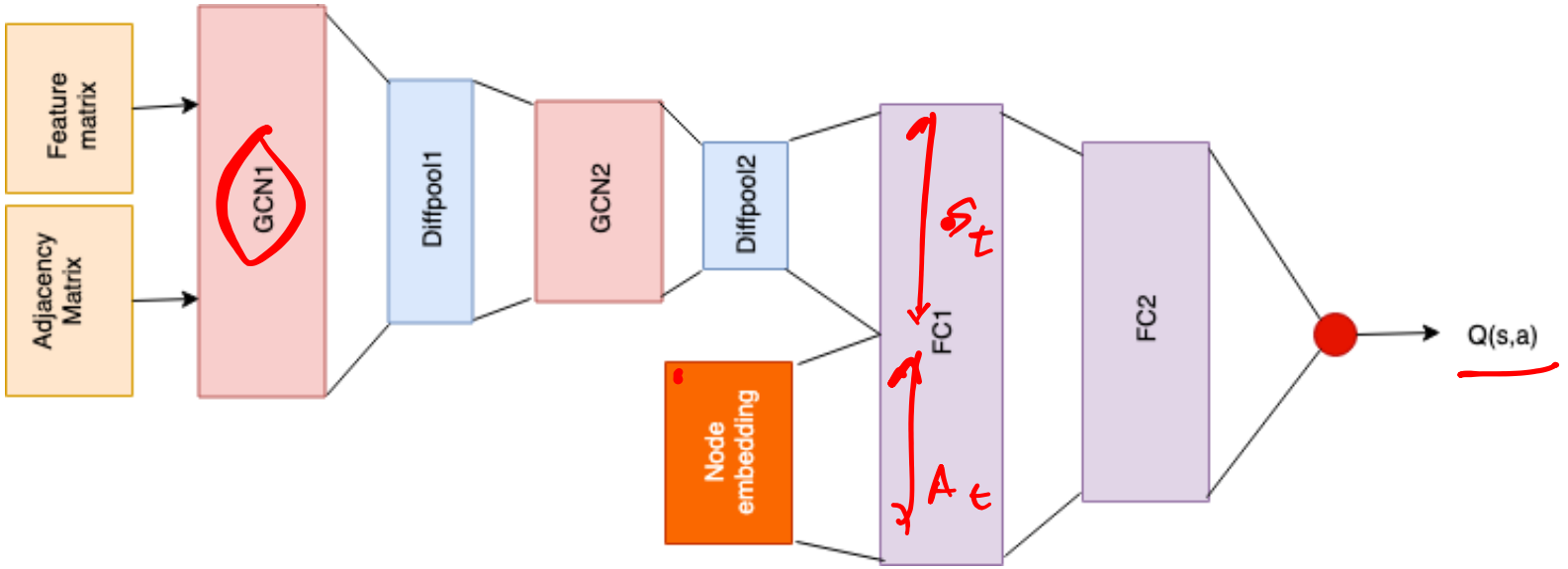


- Sample subset of nodes in the network to query.
- The queried nodes reveal their respective neighbours.
- Then we use any influence maximisation algorithm on discovered graph to pick K nodes to activate.

RL Solution


Less than query budget
S_t
State: Current Discovered Graph
RL Agent
Selects v
Query Neighbours
S_{t+1}
Query budget utilized
Influence maximization on
discovered graph

Geometric DQN



- Differential pooling based Graph convolutional (GCN) architecture [HYL17] to obtain graph representation.
- Deepwalk representation of nodes ϕ [PAS14] for actions as well as node features for (GCN) input.
- We input both state s and action a representation to the DQN and train it to predict the state-action value $Q(s, a)$.

Results: Overview

Network Family	increase %	improve %
Rural	10.54	23.76
Animal	36.03	26.6
Retweet	33.87	19.7
Homeless	21.03	7.91

Scores are averaged over test networks for each class

Conclusions

- Proposed Geometric DQN to leverage structural properties and learn effective policies for the network discovery problem for influence maximization on undiscovered social networks.
- Observed 10-36% improvement over SotA.
- Graph embeddings learned by our models was used to pick nodes with high betweenness centrality with respect to the entire network, which was key to discovering important portions of the social network.

Reinforcement Learning for Unified Allocation and Patrolling in Signaling Games with Uncertainty

Aravind Venugopal, Elizabeth Bondi, Harshavardhan Kamarthi, Keval Dholakia, Balaraman Ravindran, Milind Tambe

Motivation: Green Security Problems

Green Security Games

- Subclass of Stackelberg Security Games used to model strategic interactions between law enforcement agencies (defenders) and their opponents (adversaries).

- Model repeated interactions. [*Fang, Stone, and Tambe 2015; Fang et al. 2016; Xu et al. 2017*].

- Defenders protect a finite set of targets (e.g., wildlife) with limited resources.

Introduction

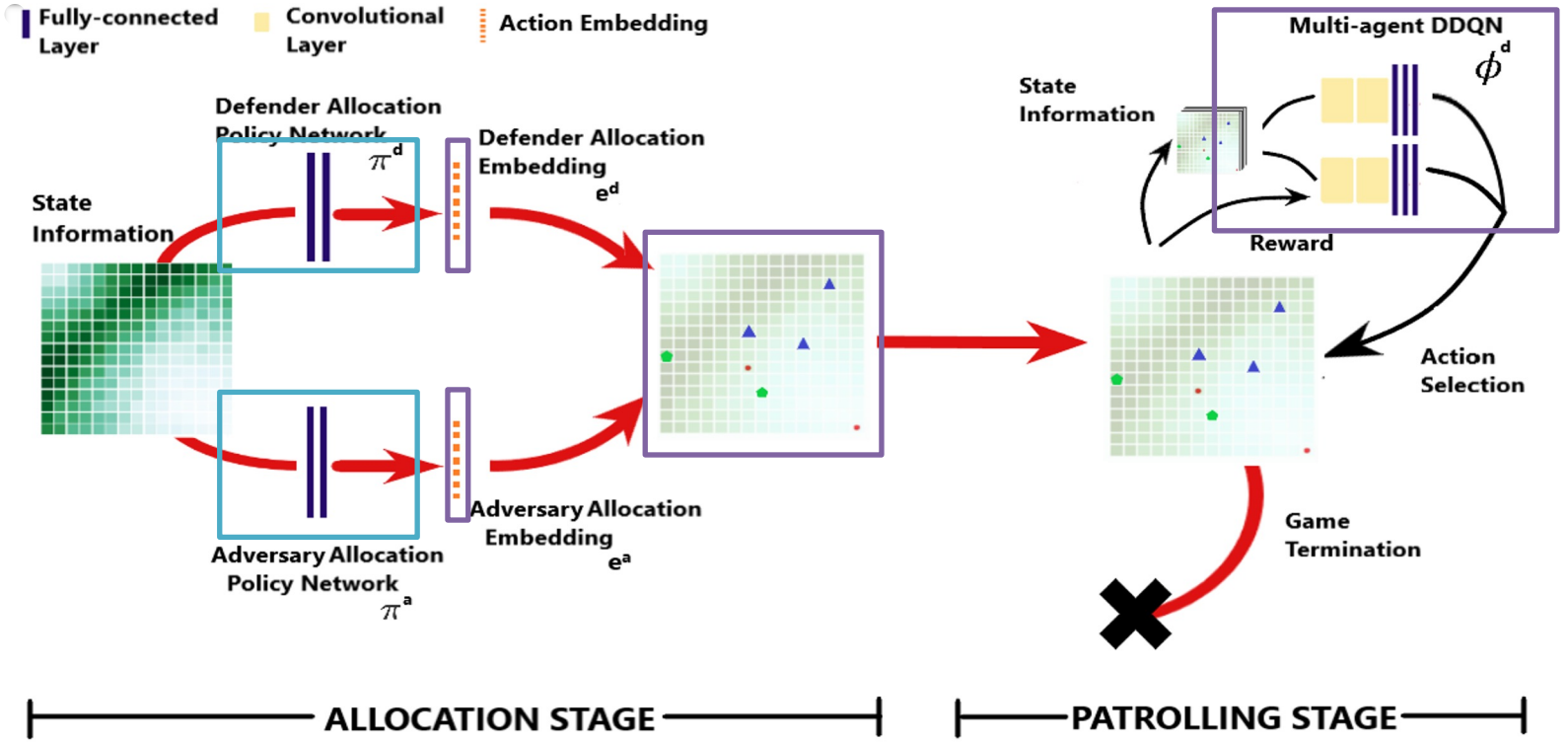
- Focus on real-world scenarios
- Combination of allocation and patrolling
- MILP and LP approaches do not scale well
- Use of Reinforcement Learning

Contributions

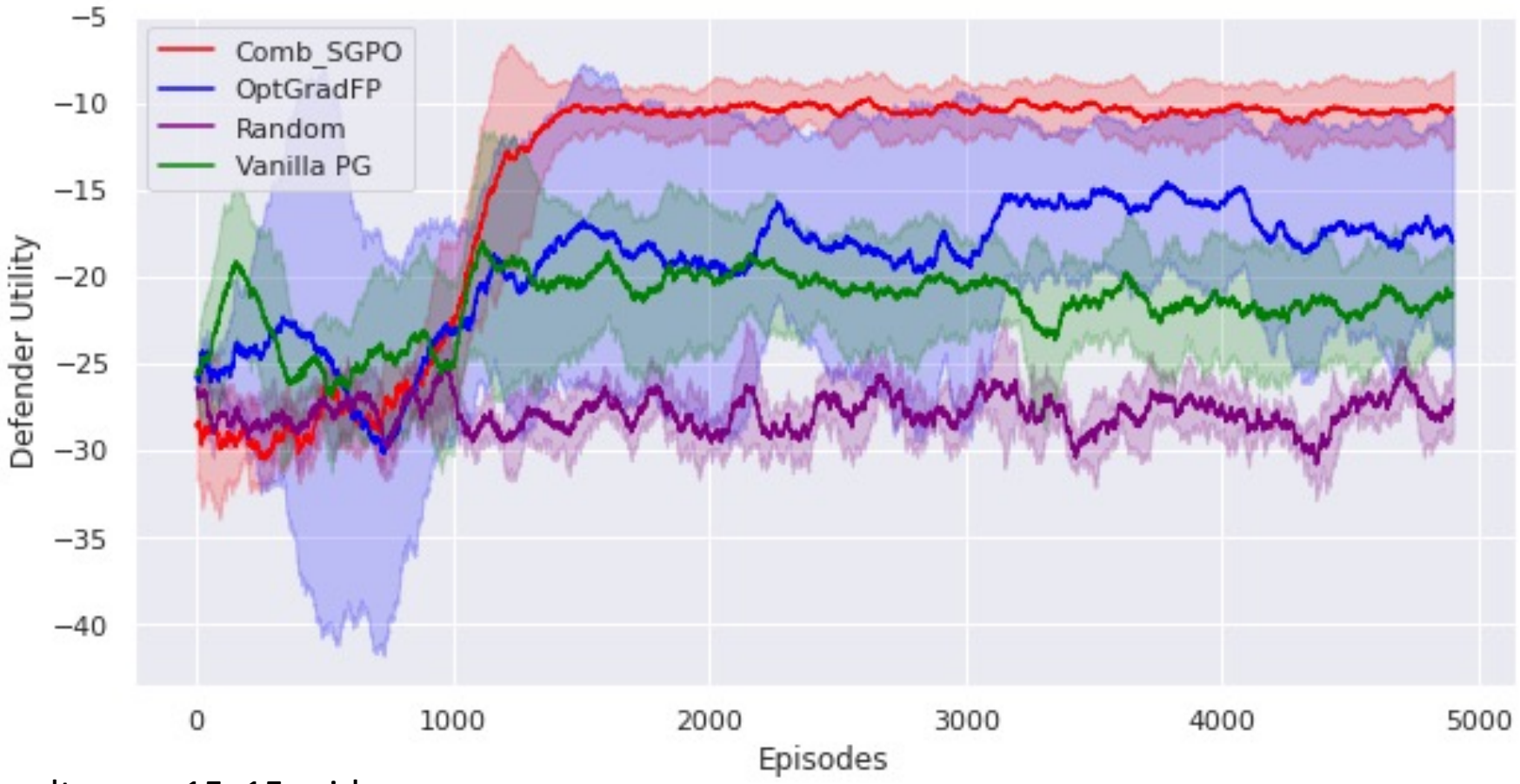
- Game model incorporating:
  - Allocation
  - Patrolling
  - Communication
  - Real-time Information
  - Uncertainty

- Novel solution strategy: **CombSGPO** (**Comb**ined **S**ecurity **G**ame **P**olicy **O**ptimization)
  - Multi Agent Reinforcement Learning
  - Action Representation Learning
  - Competitive Optimization

CombSGPO


Fully-connected Layer
Convolutional Layer
Action Embedding
Defender Allocation Policy Network
π^d
State Information
Defender Allocation Embedding
e^d
Adversary Allocation Embedding
e^a
Adversary Allocation Policy Network
π^a
Multi-agent DDQN
ϕ^d
State Information
Reward
Action Selection
Game Termination
ALLOCATION STAGE
PATROLLING STAGE

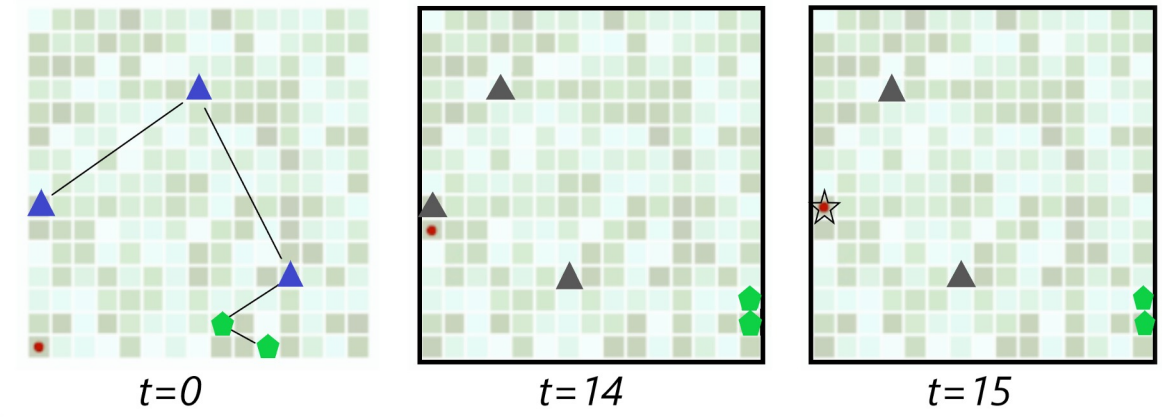
Experimental Results



Results on a 15x15 grid:
CombSGPO improves greatly over vanilla PG

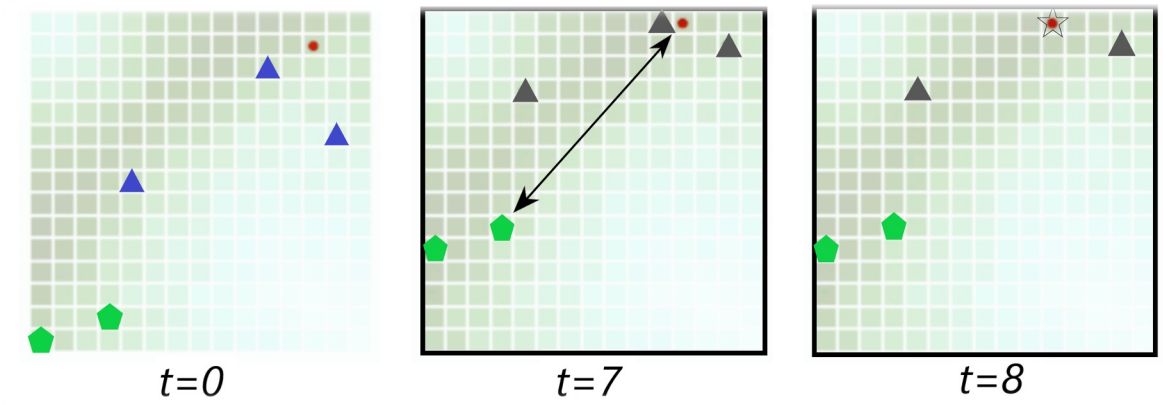
Emergent Strategic Behavior

- Qualitative Analysis of learnt policies showed:
  - CombSGPO learns to deploy agents in groups.
  - Strategic allocation based on animal density.



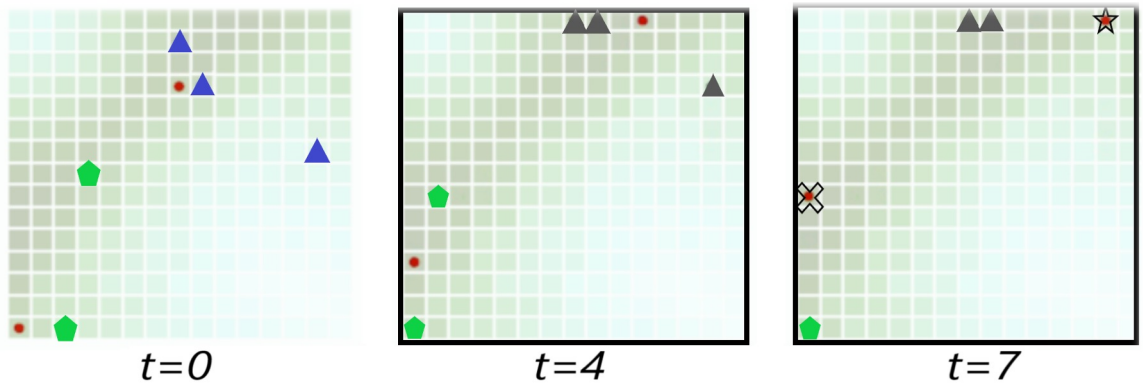
Emergent Strategic Behavior

- Ranger too far away from drones to apprehend an adversary.
- Drones learn to signal adversary and make him flee.



Emergent Strategic Behavior

- With 2 adversaries, rangers and drones split into separate groups.
- Drone group forces adversary 1 to flee. Rangers catch adversary 2.



Learning optimal bidding strategy in reactive power markets

Jahnvi Patel, Devika Jay,
B. Ravindran, K. Shanti Swarup

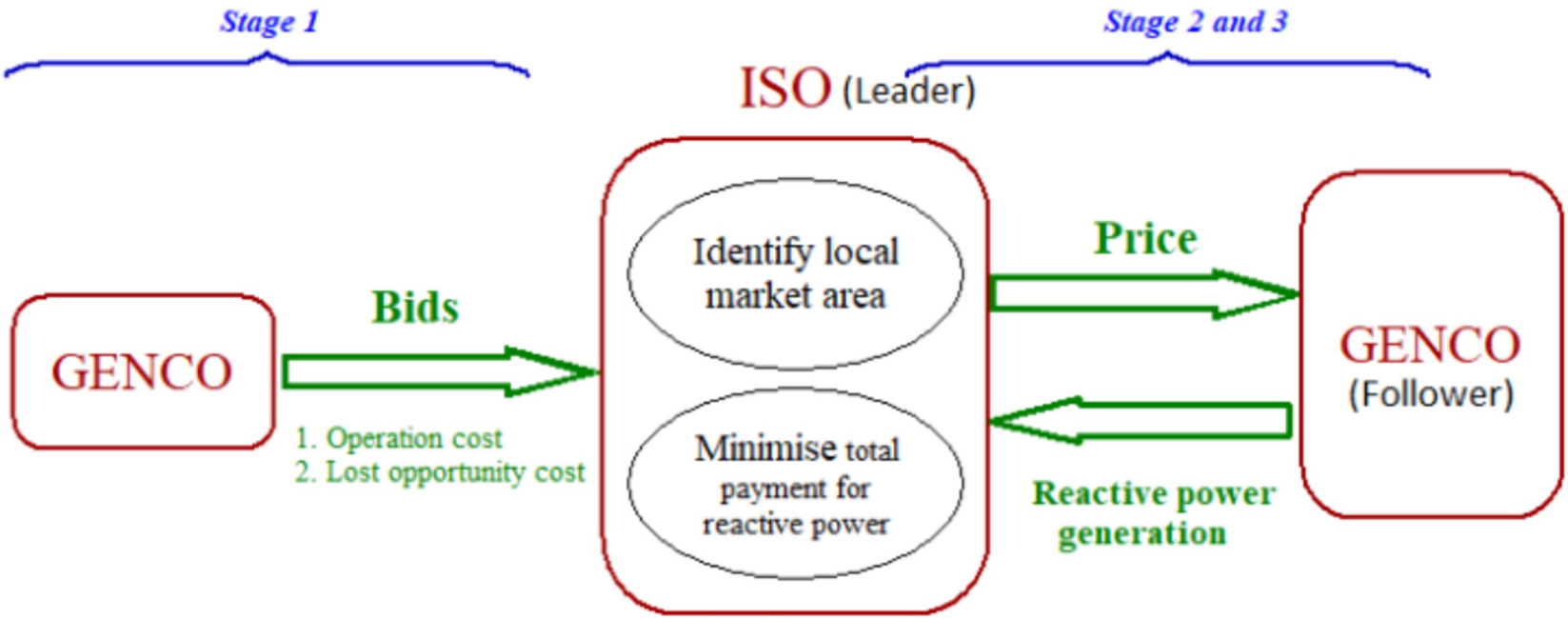
Motivation

- Reactive power markets consist of several generating companies (GENCOs) interacting with the Independent System Operator (ISO) and trying to maximize their profits subject to market constraints.
- Traditional market models based on uniform price assignment according to highest bid works only under the assumption of perfect competition and market participants bidding their actual marginal costs.
- To prevent generators from monopolizing the market and reducing price volatility, a novel three stage market mechanism is used.

Three-stage Reactive Power Market Model

- At the start of every hour, market participants submit their price bids to the ISO for the next hour in the form of operation cost and lost opportunity cost.
- ISO responds with individual price signals to each of the GENCOs such that total payment is minimized while meeting the demand and other criterion.
- GENCOs respond by submitting the quantities they would generate at the next time step.

The last two steps are repeated till all the market conditions have been satisfied.


Stage 1
Stage 2 and 3
ISO (Leader)
GENCO
Bids
1. Operation cost
2. Lost opportunity cost
Identify local market area
Minimise total payment for reactive power
Price
Reactive power generation
GENCO
(Follower)

Formulating the RL problem and approach

State Representation	Actions	Rewards
Previous bids and reward signals for {t − 48, t − 24, t − 2, t − 1} and demand are concatenated as: <Previous Bids | Feedback | Quantity Estimate>	Bids sent by the GENCO relative to the actual costs. Bounded by the box (1,1) to (5,5).	Profit generated over baseline. Baseline is defined as profit of GENCO when it bids actual costs.

Neural Fitted Q Iteration is used which is a batch mode, data-efficient Q-learning algorithm that can be extended to continuous state spaces. NFQ uses a neural network for regression on Q values. Actions are discretized using step size of 0.5. Soft updates, prioritized experience replay and target network is used to improve the performance of original implementation. Two different architectures are compared:

- NFQ-1: Input to the network is a pair and output is the Q-value for that action.
- NFQ-2: Input to the network is the state representation and output is the Q-value for all actions.

Results and Conclusion

Consider two rival bidding strategies:

B-1 - All the agents except learner send bids as a multiplier of demand fluctuations

B-2 - All the agents except learner bid their true cost

The table shows the optimal bidding strategy was able to beat the rival strategies in most scenarios.

GENCO	NFQ-1		NFQ-2	
#	B-1	B-2	B-1	B-2
1	0.0	1.26	0.0	1.94
2	9.08	26.73	14.05	45.81
3	4.70	5.49	7.26	5.49
4	12.26	6.50	16.29	9.67
5	0.0	0.0	0.0	0.0
6	0.0	0.10	0.0	3.30

https://arxiv.org/pdf/2101.02456.pdf

Final Thoughts

- Finally we have building blocks to use RL for real problems
- We need a concerted effort to widen the scope of RL
- Evolve a set of best practices for building data science applications, enterprise applications built on RL
- Put RL to work! Go beyond fun and games! :-)

RBC DSAI
ROBERT BOSCH CENTRE FOR DATA SCIENCE AND ARTIFICIAL INTELLIGENCE IIT MADRAS

5th Floor, Block II,
Bhupat and Jyoti Mehta School of Biosciences,
Indian Institute of Technology Madras,
Chennai, India

https://rbcdsai.iitm.ac.in/

@rbcdsai

@rbc_dsai_iitm