Knowledge Compilation for Constrained Combinatorial Action Spaces in Reinforcement Learning

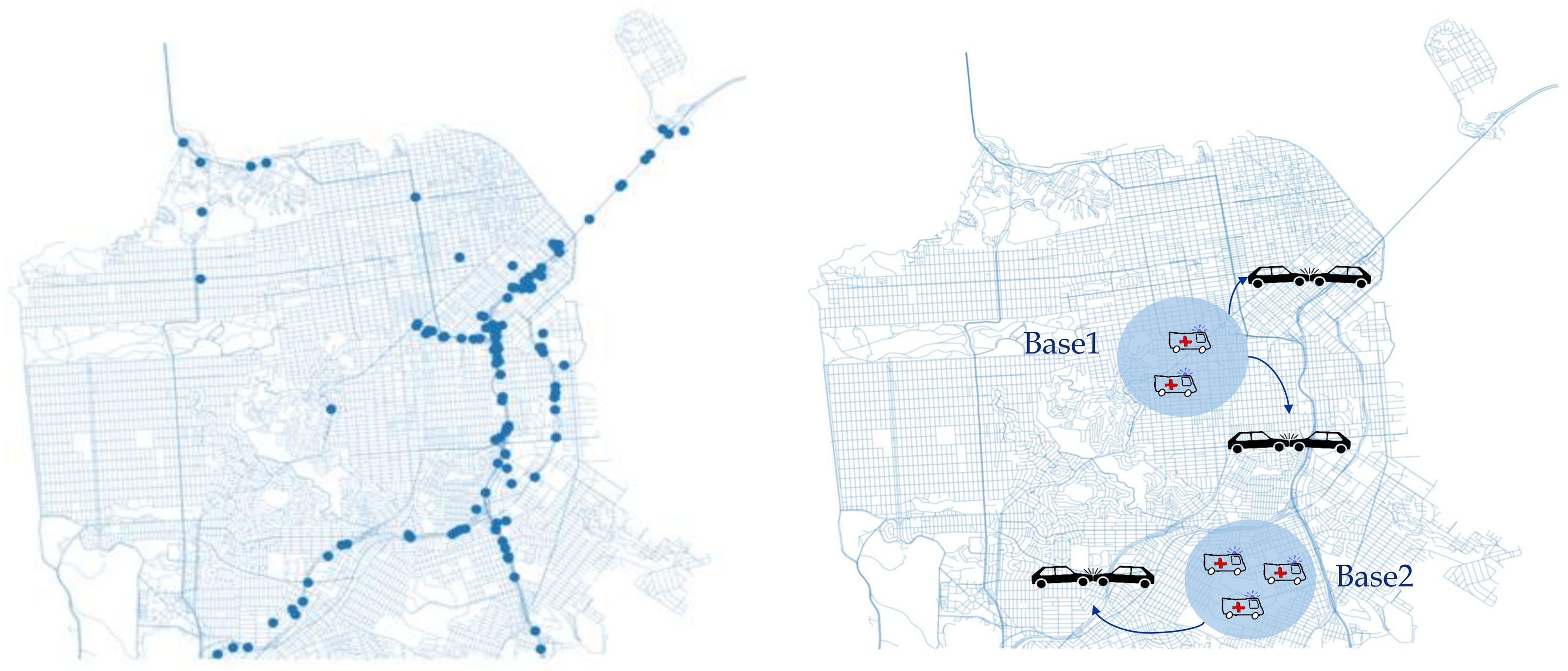
SMU Singapore Management University — School of Computing and Information Systems

Jiajing Ling*, Moritz Lukas Schuler*, Akshat Kumar, Pradeep Varakantham

{jjling.2018, akshatkumar, pradeepv}@smu.edu.sg, m-schuler@mailbox.org

Motivating Example

Emergency Response System:
Serve ambulance request due to emergency incidents using the limited resources



Monthly traffic incidents occurring in San Francisco in 2021.
Source: https://smoosavi.org/datasets/us_accidents

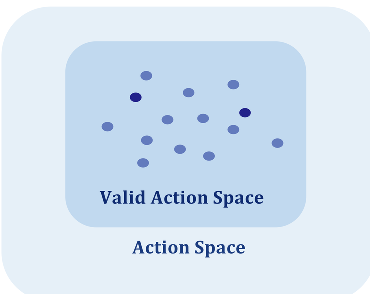
Problem Formulation

- **Action-constrained Markov Decision Process (MDP)**
  - An MDP $(S, A, T, r, \gamma, b_0)$ + Explicit constraints on actions
    - Action space is discrete and combinatorial
    - For each state s , there is a valid action set $\mathcal{C}(s) \subseteq A$ determined by constraints.
  - No cost function $c(s,a)$ defined as in standard constrained MDP
- **Optimization problem**

$$\max_{\theta} \quad J(\pi_\theta) = \mathbb{E}_{s\sim b_0}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s; \pi_\theta]$$
$$s.t. \quad a_t \in \mathcal{C}(s_t) \quad \forall t$$

Key Challenges

- **Action space is combinatorial**
  - Example: allocate 32 ambulances into 25 base stations
  - $|A| = \binom{32+25-1}{25-1} = 4.355031703$ E+15
- **Hard constraints**
- **Limitations of previous methods**
  - Constraint violations
  - Computationally expensive



Contributions

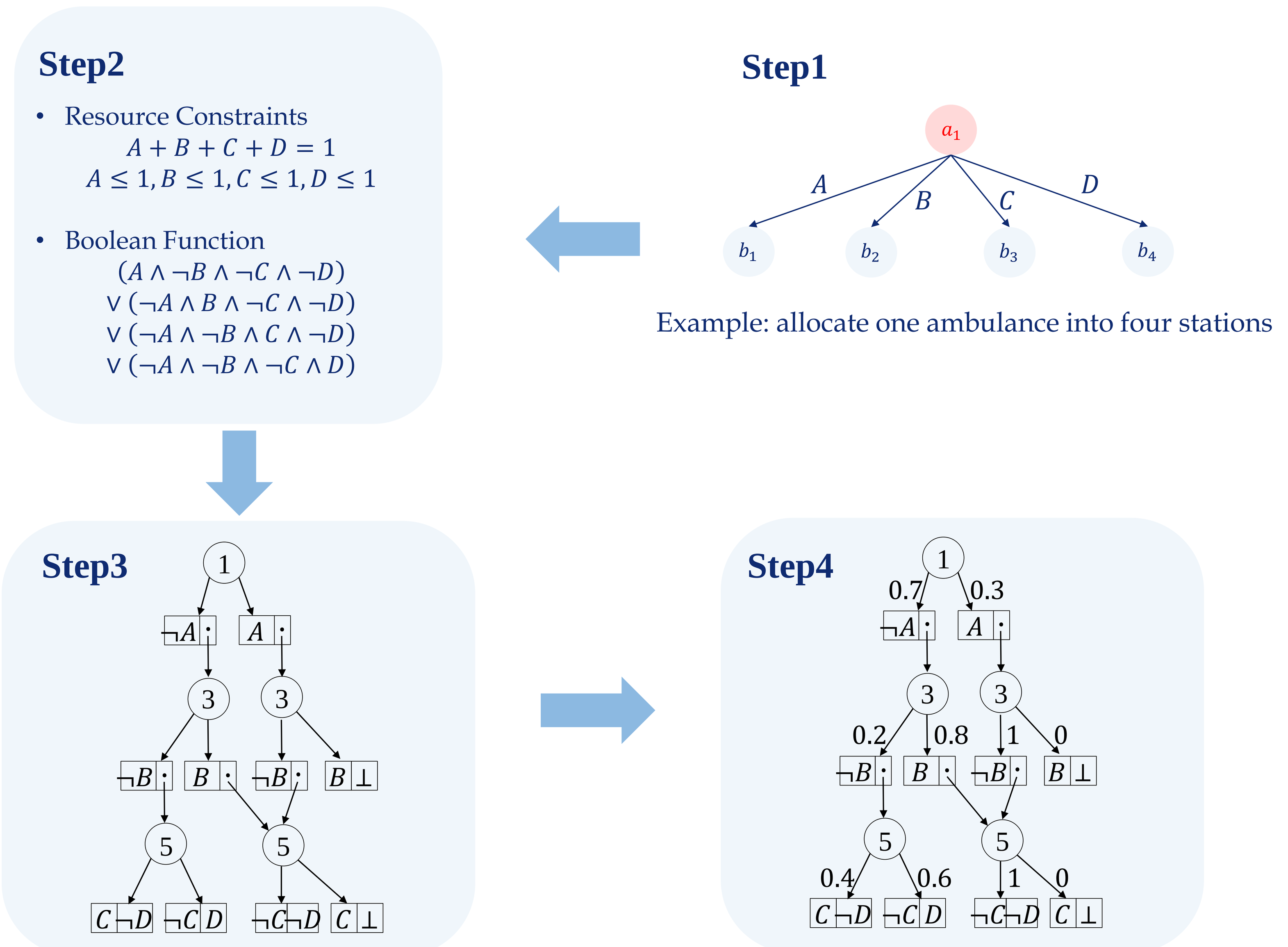
We propose a neuro-symbolic method to address action-constrained RL

Symbolic Representation
- Compile action constraints using (P)SDDs
- Able to compactly represent a distribution over all valid actions

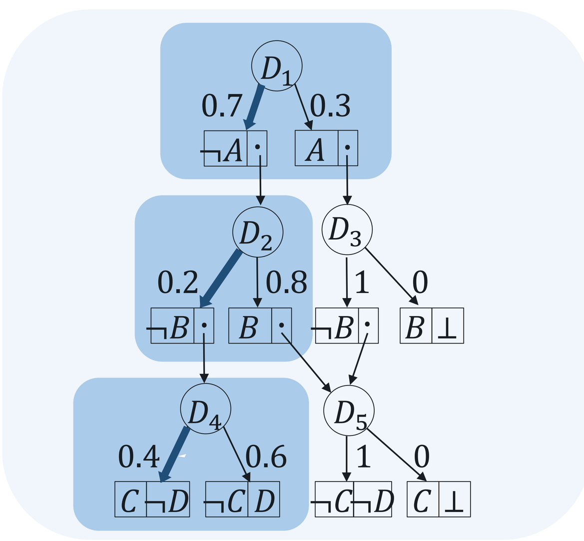
Neural Optimization
- Integrate PSDD with deep NN
- Optimize PSDD parameters using deep RL

Symbolic Representation

- **Steps**
  1. Define state and action using propositional variables $\boldsymbol{X_S}, \boldsymbol{X_A}$
  2. Define constraint using Boolean functions $C_k(\boldsymbol{X_S}, \boldsymbol{X_A})$
  3. Compile Boolean functions into a Sentential Decision Diagram (SDD)[1]
  4. Parameterize the SDD to obtain a Probabilistic SDD(PSDD)[2]



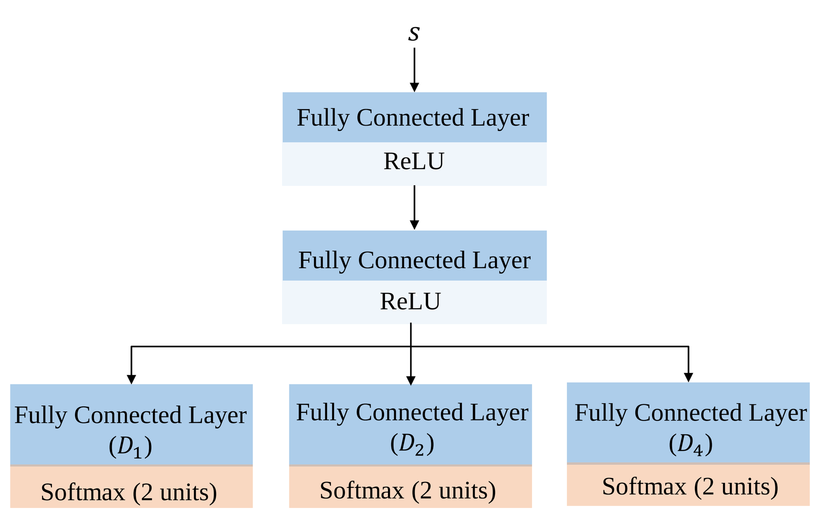
- **Fast sampling actions from the PSDD**
  - Select a branch for decision nodes
  - $P(\neg A, \neg B, C, \neg D) = 0.7 \times 0.2 \times 0.4 = 0.056$
  - Linear complexity in depth

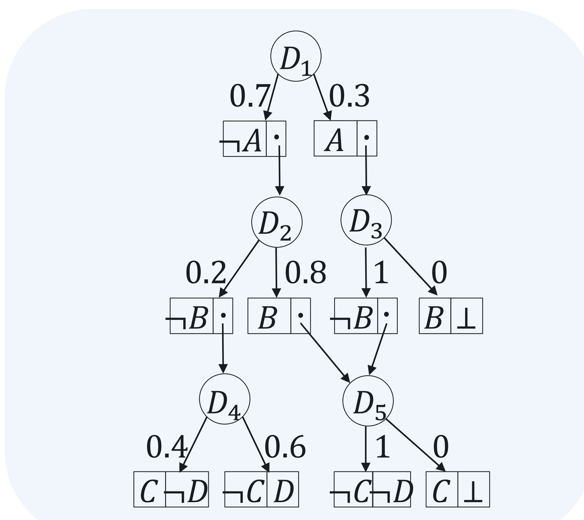


- **Benefits of using PSDDs**
  - Decomposition of complex constraints
  - Fast sampling of an action
  - Actions guaranteed to be valid
  - Tractable number of parameters

Neural Optimization

- **Steps**
  1. Integrate PSDD with deep NN
     - Each head outputs a probability distribution for a decision node

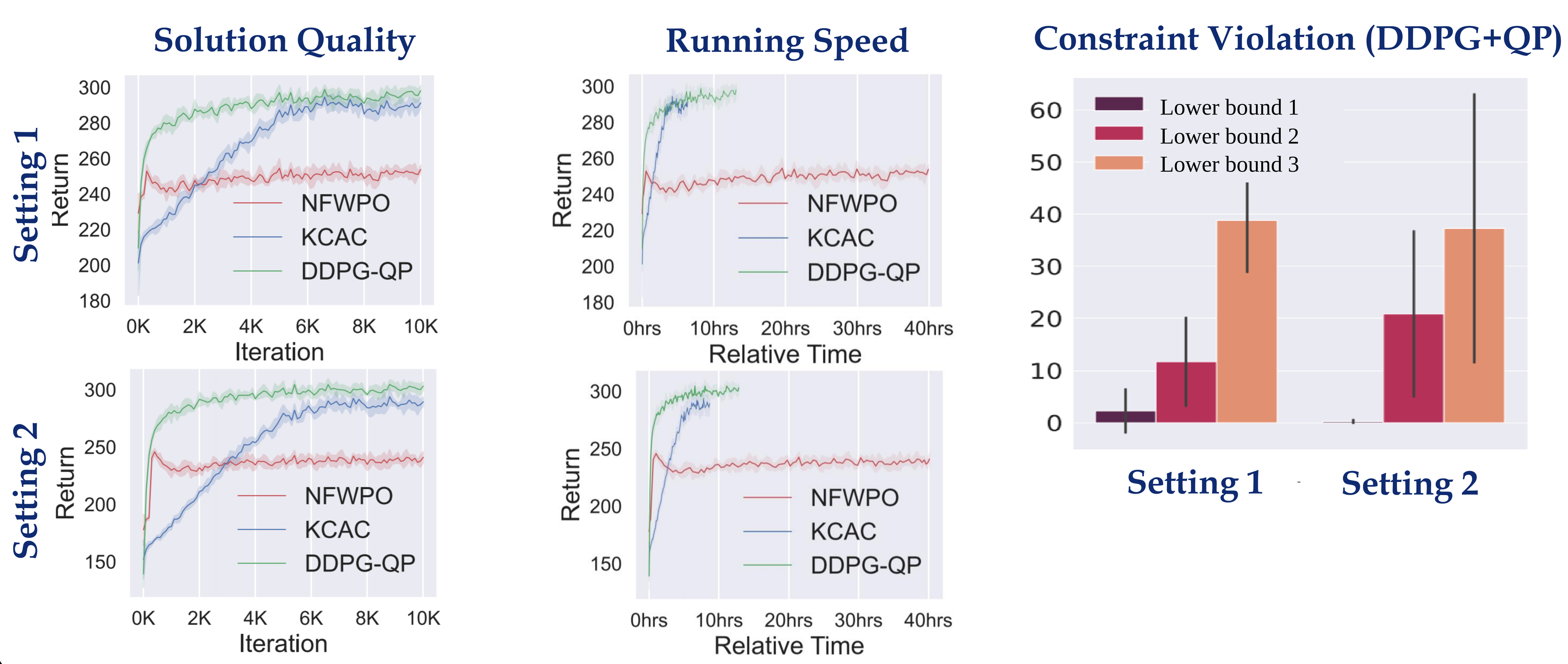




  2. Optimize PSDD parameters using amortized Q-learning[3]
     - a_{PSDD}: sampled branches of all decision nodes
     - Maximize the probability of getting a PSDD action with the highest Q-value
     - Critic network is learned using a_{env} mapsped from a_{PSDD}

Experiments

- **Emergency Response System**
  - Allocate 32 ambulances into 25 base stations
  - Constraints:
    - Global sum, local min and max, group min and max
- **Baselines**
  - NFWPO
    - The state-of-the-art approach for ACRL with continuous action space
    - Involve solving an Integer Quadratic Program (IQP)
  - DDPG+QP
- **Results**



References

[1] Adnan Darwiche. 2011. SDD: A new canonical representation of propositional knowledge bases. In IJCAI
[2] Doga Kisa et. al. 2014. Probabilistic sentential decision diagrams. In PKRR.
[3] TomVan de Wiele et. al. 2020. Q-Learning in enormous action spaces via amortized approximate maximization.