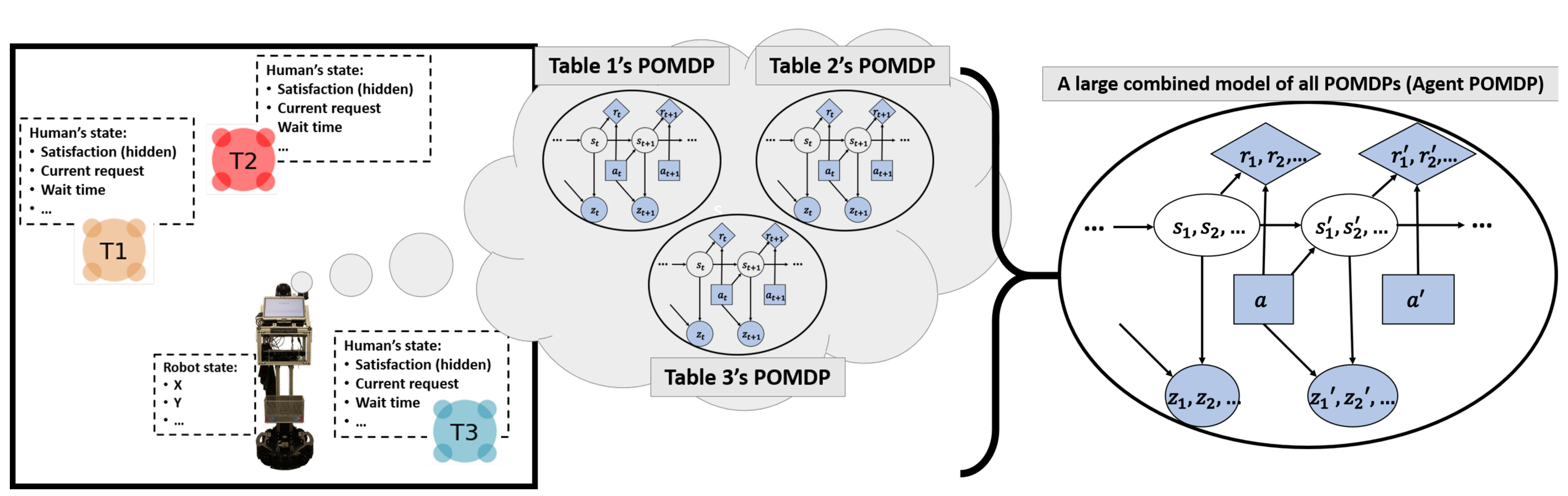
Efficient Robot Planning for Achieving Multiple Independent Partially Observable Tasks that Evolve Over Time

Anahita Mohseni-Kabir, Manuela Veloso, and Maxim Likhachev
{anahitam, mmv, maxim}@cs.cmu.edu
School of Computer Science, Carnegie Mellon University

Introduction

- **Objective:** In a lot of robotics applications, a robot is in charge of multiple tasks. For example,
  - A mobile robot operating in office buildings to deliver messages, deliver objects, give directions to people, etc.
  - A robot deployed in a restaurant.
  - A robot deployed in a factory setting, helping factory workers, *e.g.*, by providing tools.
- **Application:** The robot waiter operating in a restaurant
  - Is presented with an ongoing stream of tasks, *e.g.*, taking food orders and checking on customers.
  - Should attend the customers' needs in a timely and efficient manner to keep everyone satisfied.
  - An action that the robot takes depends on
    - The duration of possible actions.
    - The state of each table.
    - How these tables evolve over time.

- **Restaurant Model**: N models for the N tables [3].
  - **Solution:** the optimal policy must consider all tasks.
  - **Challenge:** for large number of tasks solving the combined model is computationally impractical.



Contributions

1. **Formalize the class of problems with multiple independent tasks that evolve over time**

   Key idea: We call a set of N POMDPs, P, **independent** iff for each two tasks in the set, the following holds:
   - The POMDPs do not share 1) state variables, 2) observation variables, and 3) actions except a no operation action.
   - The transition/observation function for one task does not depend on the states of the other tasks.
2. **Develop optimal, scalable and real-time planning algorithms**

   Key idea: instead of solving the agent POMDP model, **solve series of smaller POMDPs**.
   - In H-step horizon, the robot can only attend to k tasks.
   - The robot can consider all possible subsets of size k. **Dramatically smaller and simpler to solve**
   - We **prune** the set of subsets using the solutions to the individual tasks.

Algorithm

- Consider subsets of the N tasks with size k

$$tpls = \{tpl \in \mathcal{P}(P) : |tpl| = k\}$$

- Prune the subsets
  - Solve each individual POMDP separately
  - Compute a lower-bound LB

    Intuition: the robot can only execute one task in its planning horizon.

$$LB = \max_{p \in P}(V_p^*(b_p) + \sum_{q \in P \setminus \{p\}} V_q^\tau(b_q))$$

  - Compute an upper-bound on the value of the subsets

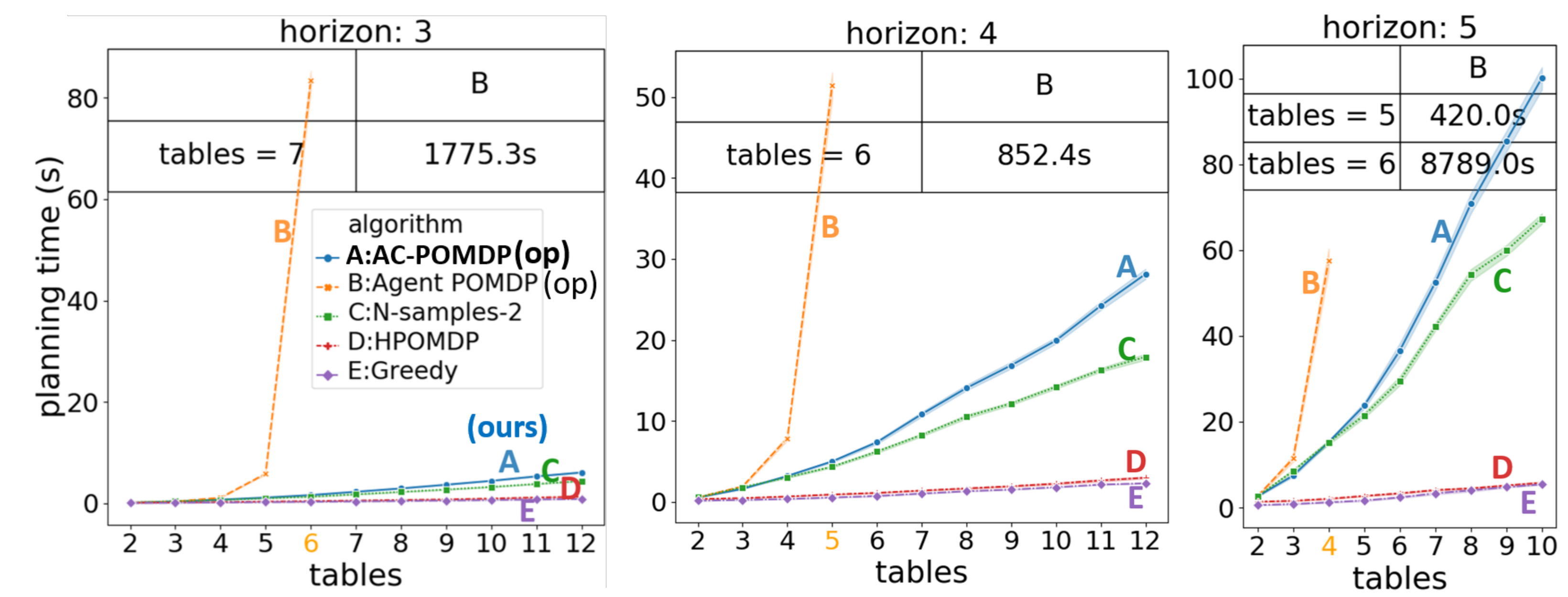
    Intuition: after the first action execution, all tasks can be addressed in parallel.

$$UB_{tpl} = \max_{a \in A_{tpl}}(\sum_{p \in tpl} Q_p^*(b_p, a[p])) + \sum_{q \in P \setminus tpl} V_q^\tau(b_q)$$

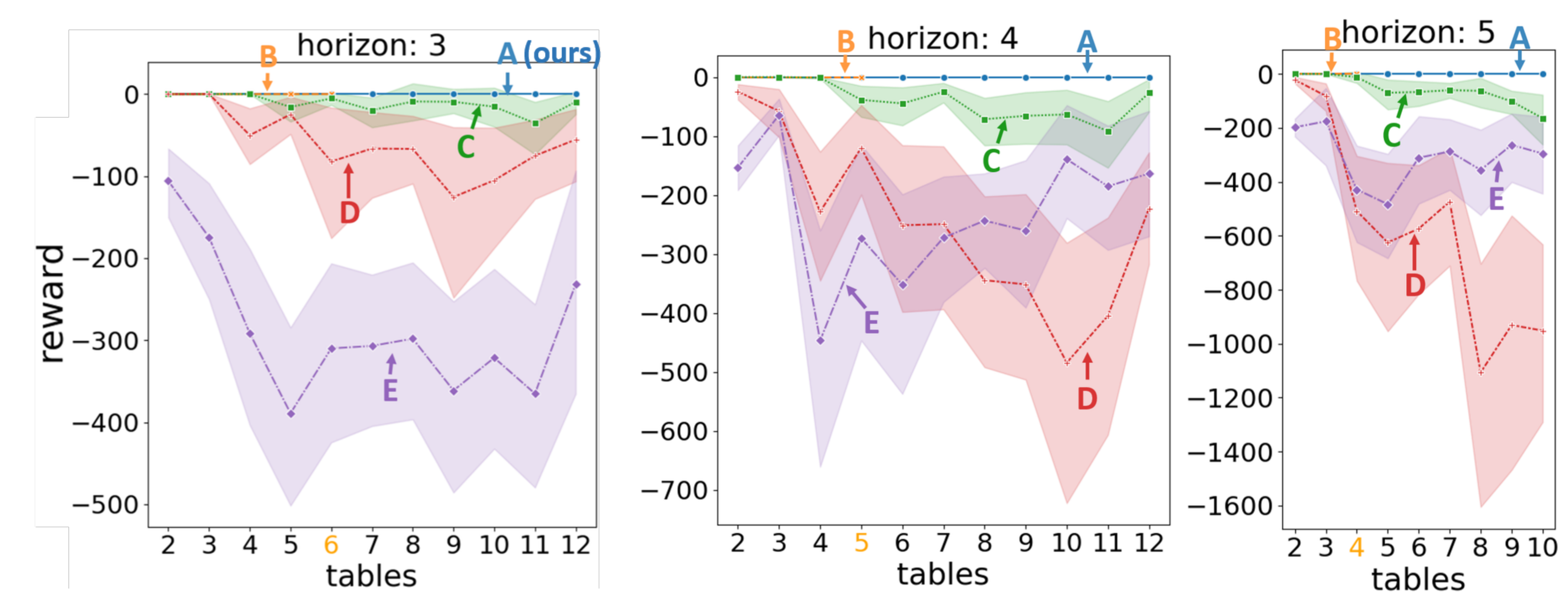
  - Prune all the subsets where $UB_{tpl} < LB$
- Solve the remaining subsets optimally to find the best action

Results

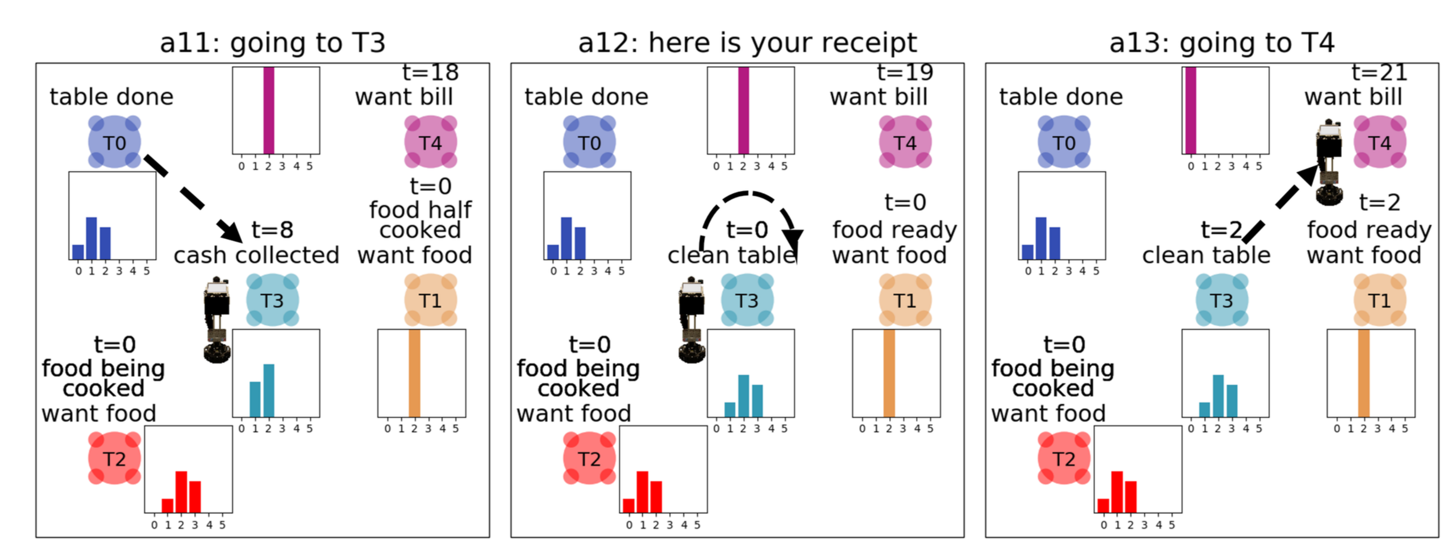
- **Planning time:** $B > A > C > D > E$.



- **Average reward:** we take the difference between the average reward of our approach and the other approaches. The average reward mostly follows $B \approx A > C$.



- **Qualitative results:**



[1] A. Mohseni-Kabir, M. Veloso, and M. Likhachev, Optimal Planning over Long and Infinite Horizons for Achieving Independent Partially-Observable Tasks that Evolve over Time. Under review.
[2] A. Mohseni-Kabir, M. Veloso, and M. Likhachev, Efficient Robot Planning for Achieving Multiple Independent Partially Observable Tasks that Evolve Over Time. ICAPS, 2020.
[3] A. Mohseni-Kabir, M. Likhachev, and M. Veloso. Waiting Tables as a Robot Planning Problem. IJCAI Workshop on AIxFood, 2019.