Hierarchical Task Allocation and Planning for Multi-Robots under Hierarchical Temporal Logic Specifications

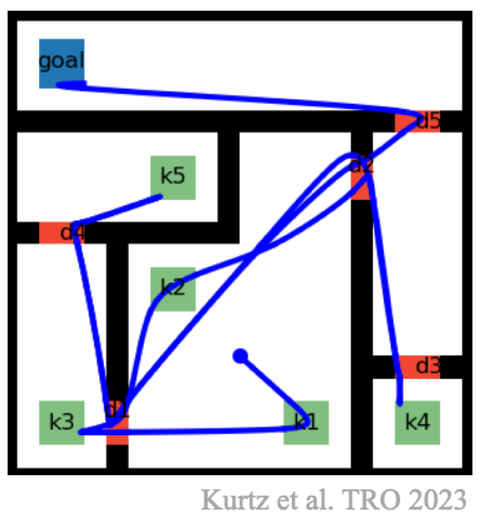
Xusheng Luo[1], Shaojun Xu[2], Ruixuan Liu[1] and Changliu Liu[1]

[1]Carnegie Mellon University, [2]Zhejiang University

Introduction

Motivation:

- Linear temporal logic (LTL) is expressive for long-term tasks but struggles with computational scalability.



Task: Pick up five keys in order to pass through corresponding doors before eventually reaching a goal.

⚠ Conversion from LTL to automaton required 32 minutes (Intel i7 CPU and 32-GB RAM)

Hierarchical sc-LTL
Luo et al. Arxiv 2024

- Hierarchical syntactically co-safe LTL (sc-LTL) improves computational efficiency by structuring the specifications in a hierarchical manner.
- There are currently no planning algorithms for hierarchical sc-LTL that explicitly account for inter-robot interactions.

Contribution: To the best of knowledge, our work is **the first one** that can tackle multi-robot collaboration under hierarchical LTL specifications.

Problem Formulation

Hierarchical sc-LTL for multi-robot pickup and delivery:

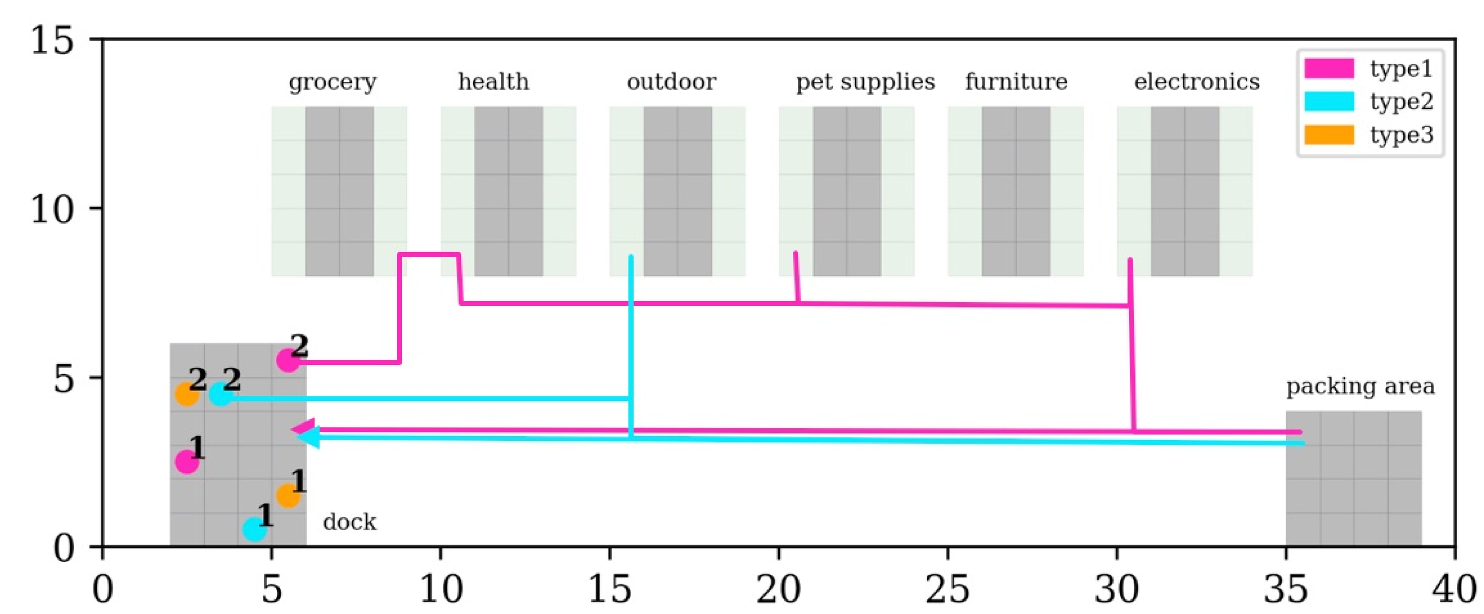


Figure 1: Topological map of the supermarket.

- A type 1 robot gathers items from the health, grocery, electronics and pet sections in any order;
- Either a type 2 robot or a type 3 robot gathers items from the outdoor section;
- After the items are delivered to the packing area, all robots eventually return to the dock.

Hierarchical sc-LTL:

high level

$$\phi(1,1) = \Diamond\, (\phi(2,1) \wedge \Diamond\, \phi(2,2)) \wedge \Diamond\, (\phi(2,3) \vee \phi(2,4))$$

low level

$$\phi(2,1) = \Diamond\, \pi^1_{pet} \wedge \Diamond\, \pi^1_{electronics} \wedge \Diamond\, \pi^1_{health} \wedge \Diamond\, \pi^1_{furniture}$$

$$\phi(2,2) = \Diamond\, (\pi^1_{pack} \wedge \Diamond\, \pi^1_{dock})$$

$$\phi(2,3) = \Diamond\, (\pi^2_{outdoor} \wedge \Diamond(\pi^2_{pack} \wedge \Diamond \pi^2_{dock}))$$

$$\phi(2,4) = \Diamond\, (\pi^3_{outdoor} \wedge \Diamond(\pi^3_{pack} \wedge \Diamond \pi^3_{dock}))$$

Problem formulation: Given domain of multi-robot navigation or manipulation and a task represented by hierarchical sc-LTL specifications, the goal is to generate a satisfying plan.

Decomposition-based Planning

Overview:

Task allocation — What robot to do which subtask at what time step (e.g., robot 1 of type 1 reaches pet section at time 10)

Mixed Integer Linear Programming (MILP)

⬇

Motion planning — How to carry out the subtask while abiding by the temporal relations.

Task decomposition:

- Extract the temporal relations between sub-tasks for each specification;
- Deduce the temporal relations between sub-tasks across different specifications to construct the task network;

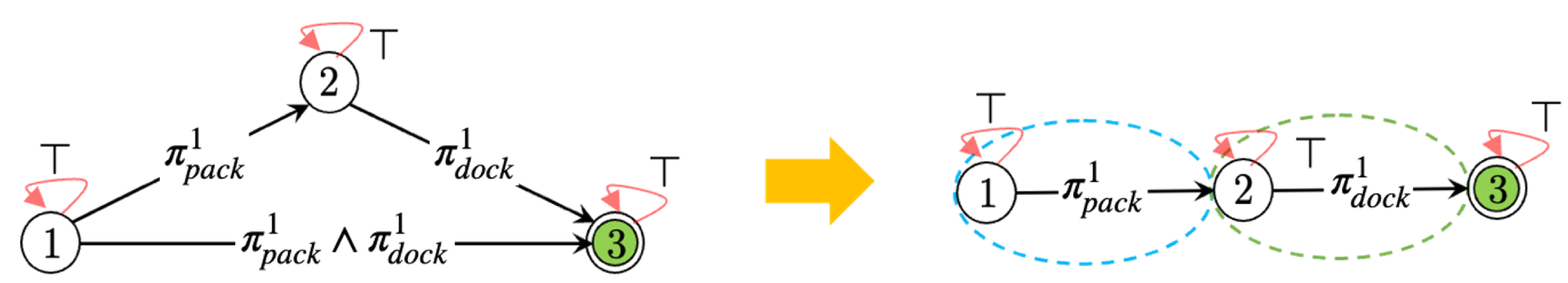


Figure 2: Sub-tasks along with temporal relations in specification $\phi(2,2)$.

Task network:

- Sub-tasks inherit the temporal relations of their respective composite propositions at the upper level.

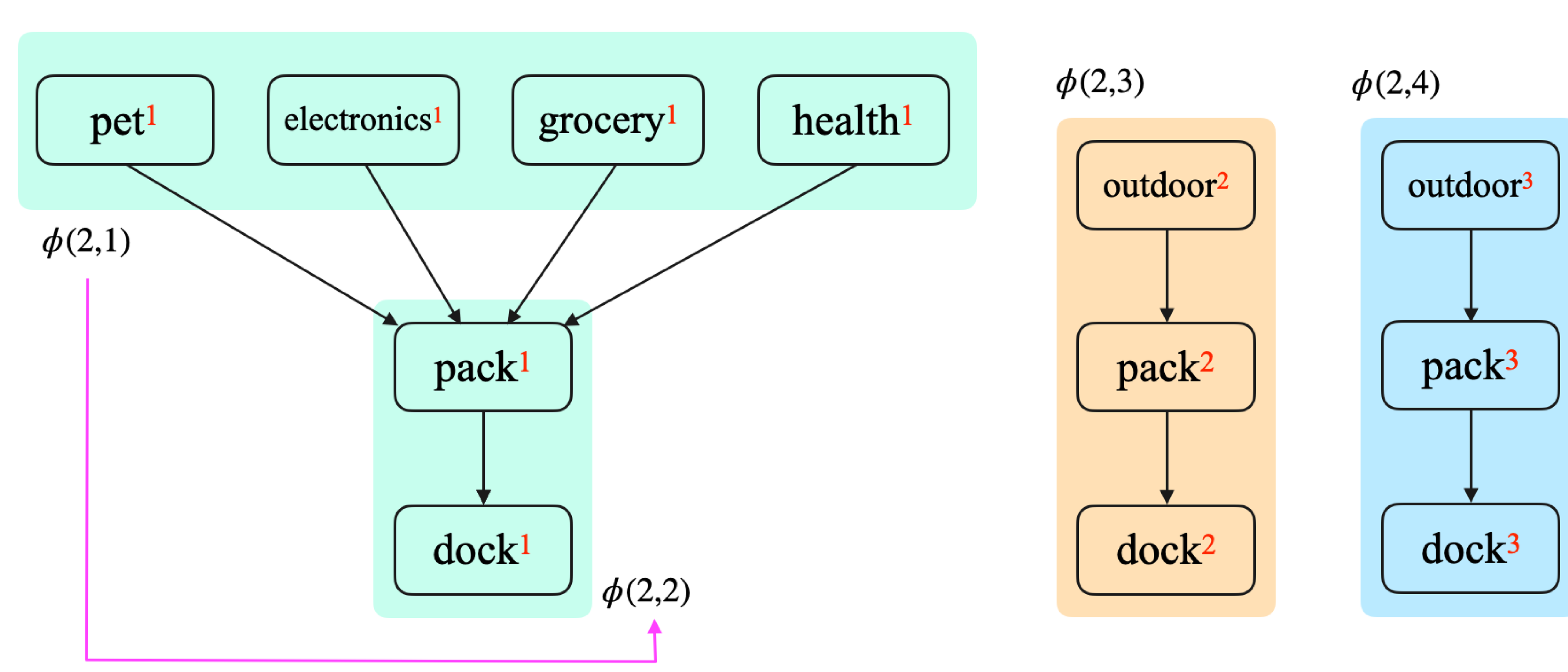


Figure 3: Task network consisting of all sub-tasks at the lowest level.

Simulation Results

Navigation:

task	l_{std}	l_{hier}	$\mathcal{A}_{\text{std}}$	$\mathcal{A}_{\text{hier}}$	t_{std}	t_{hier}	c_{std}	c_{hier}
1	51	**35**	(387, 13862)	**(33, 45)**	126.1±2.7	**18.4±0.8**	289.3±3.0	**237.5±3.3**
2	45	**19**	(12, 63)	**(20, 28)**	30.8±0.5	**24.8±0.4**	115.6±1.9	**114.8±1.6**
3	35	**27**	(113, 1891)	**(35, 101)**	25.8±1.8	**21.4±0.5**	148.1±2.1	**147.7±2.1**

Table 1: Comparative results. l_{std}, l_{hier}: length of standard and hierarchical sc-LTL formulas, $\mathcal{A}$: size of automata (nodes, edges), t: runtimes (seconds), c: plan horizons.

Manipulation:

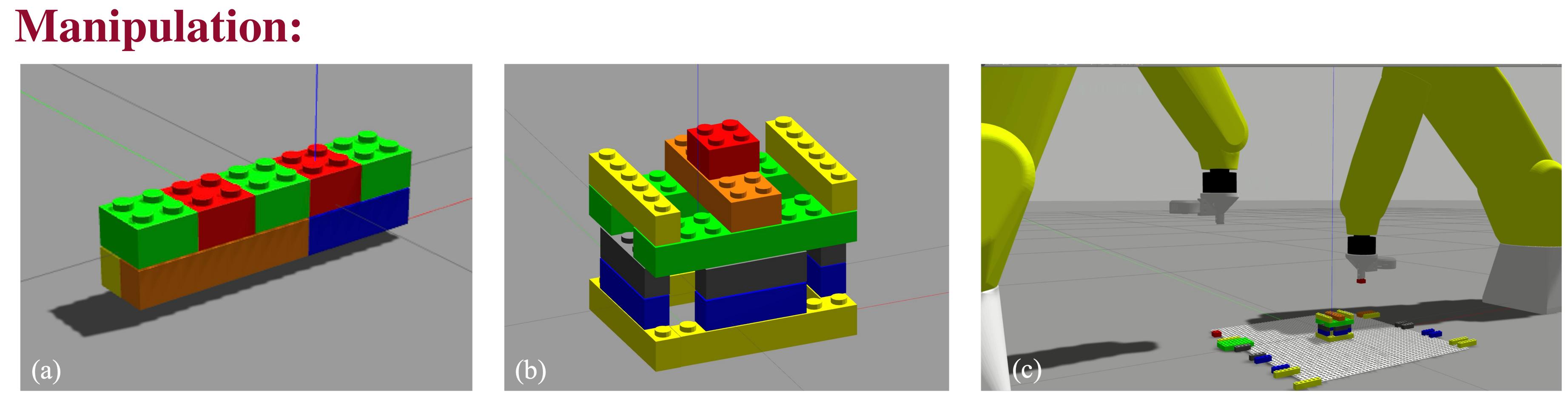


Figure 4: Collaborative construction of LEGO models. Time: (a) 4.9s (b) 4.8s