HTN Method Selection Guided by Multi-Objective Reinforcement Learning and Dynamic Preferences

Anderson G. Alves[1] Lucas N. Alegre[1] Anderson R. Tavares[1]

[1] Universidade Federal do Rio Grande do Sul

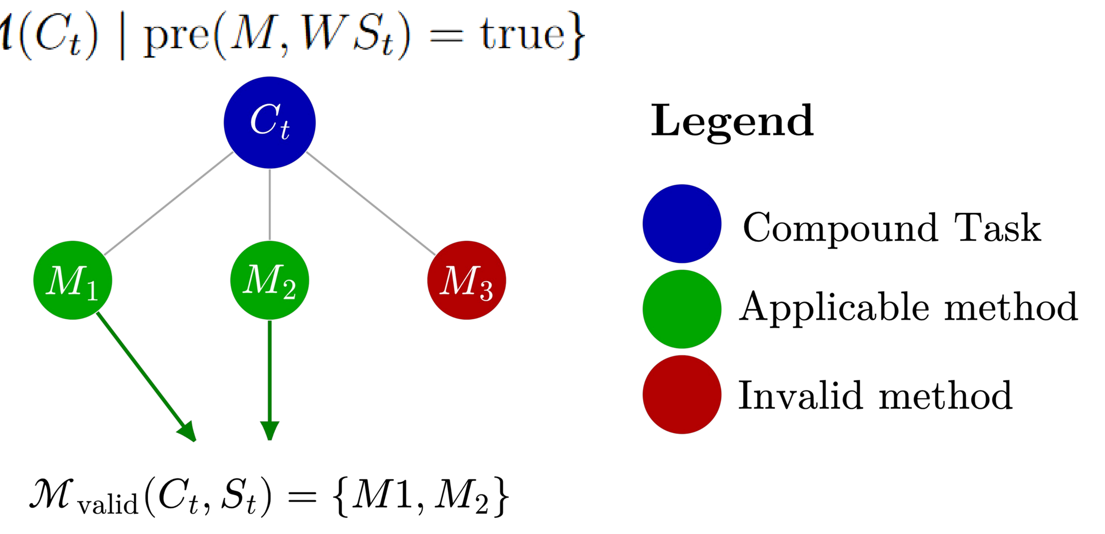
anderson.goncalves@inf.ufrgs linkedin.com/in/agacf https://github.com/AndersonGACFilho/htn-goap-morl-agent

Hierarchical Task Network Method Selection

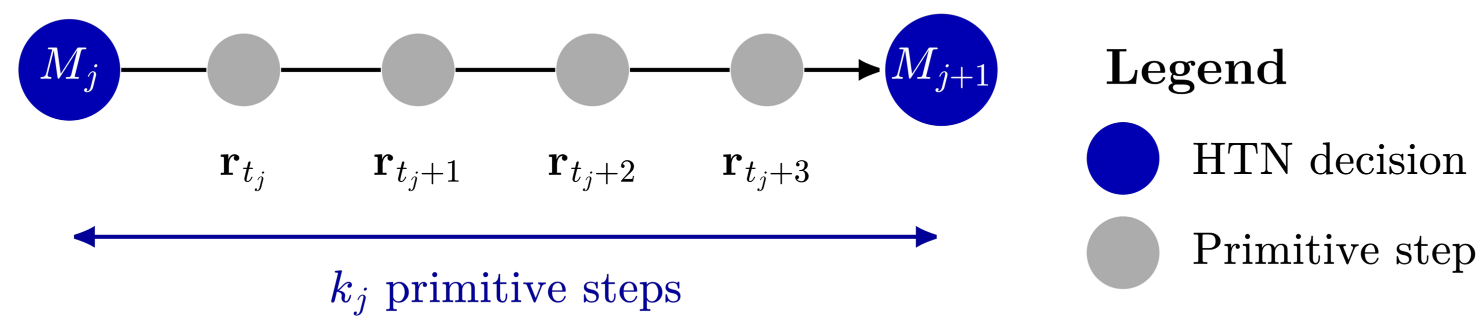
Filter Applicable Methods

$$\mathcal{M}_{\text{valid}}(C_t, WS_t) = \{M \in \mathcal{M}(C_t) \mid \text{pre}(M, WS_t) = \text{true}\}$$

Goal: expose only semantically valid methods under designer-defined constraints.



Hierarchical Credit Assignment



$$\mathbf{R}_j^M = \sum_{i=0}^{k_j-1} \gamma^i \mathbf{r}_{t_j+i}$$

Credit is assigned at the hierarchical decision level, not to primitive actions.

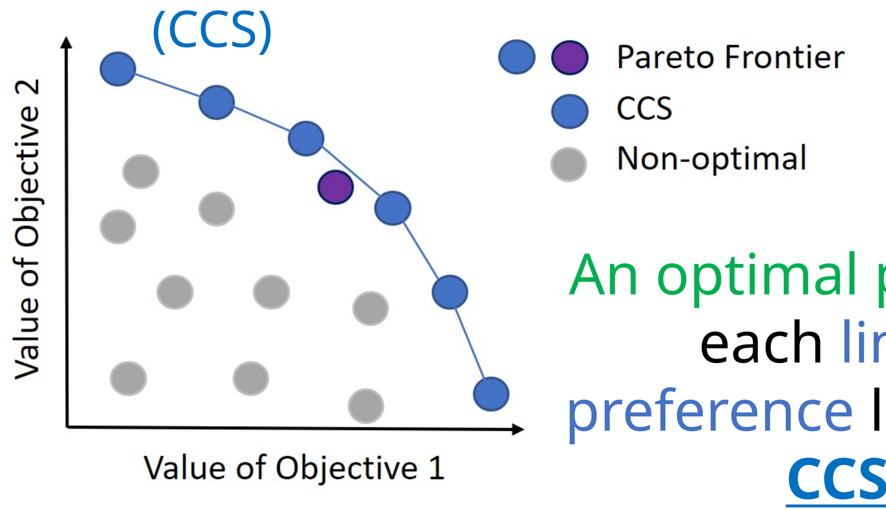
Multi-Objective Reinforcement Learning

Multi-objective reward

$$\mathbf{r} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}^m$$

Goal: optimize behavior under different trade-offs between objectives

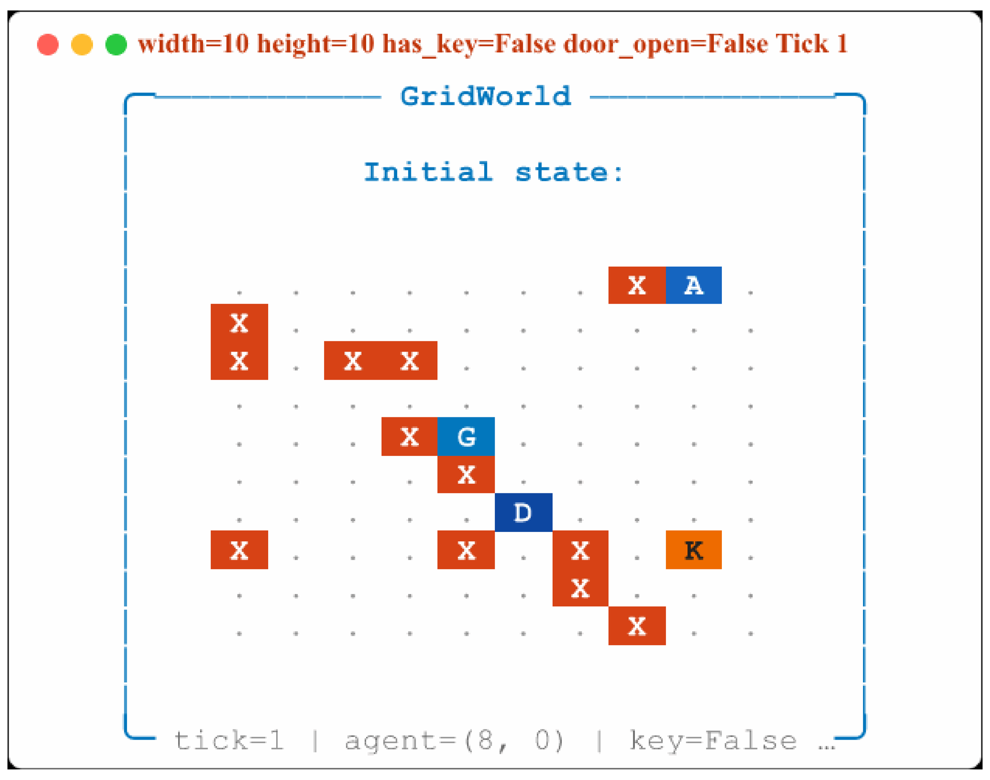
Convex Coverage Set (CCS)



An optimal policy for each linear preference lies in the **CCS**!

Proposed Experiments

Current Prototype:



Current Prototype

- HTN Execution
- Sensing
- Navigation
- Replanning

Next

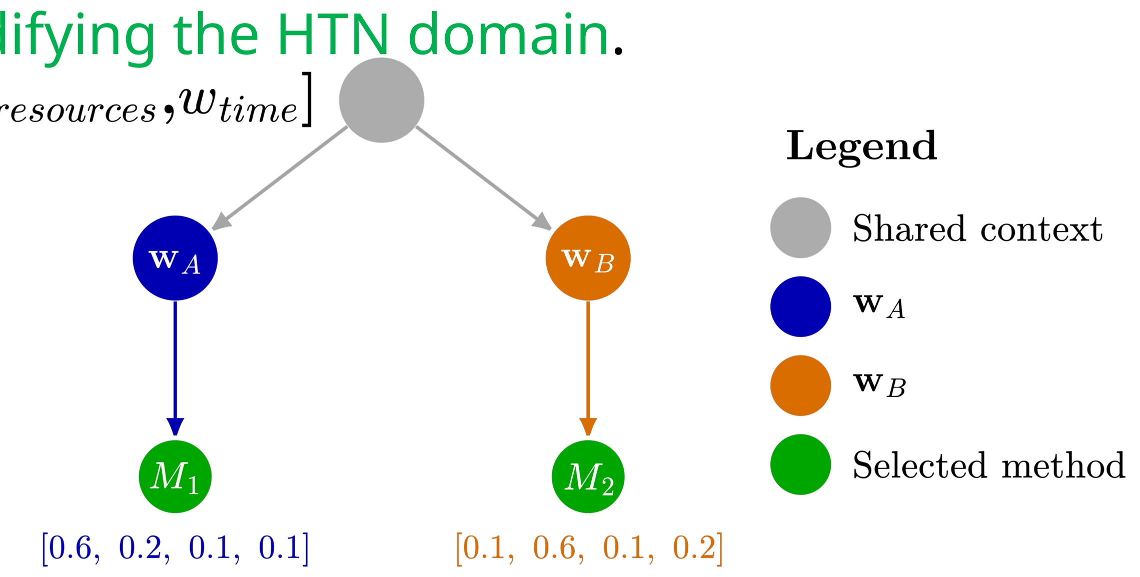
- Vector rewards
- Conflicting Objectives
- Dynamic Preferences

Dynamic Preferences

Preferences encode the current trade-off between competing objectives. Changing **$\mathbf{w}_t$** can change method selection without modifying the HTN domain.

$$W_t = [w_{mission}, w_{safety}, w_{resources}, w_{time}]$$

Goal: adapt method selection to changing objective priorities without modifying the HTN domain.



Compare

- First-applicable HTN
- Heuristic HTN
- Scalar HTN-RL
- Contextual scalar RL+ HTN mask
- **HTN + Preference-Conditioned MORL**

Stress

- Dynamic preferences
- Novel context
- Unseen preferences

Measure

- Performance
- Adaptation
- Generalization
- Semantic Validity
- Latency

Proposed Architecture

HTN filters decomposition methods using symbolic constraints, while preference-conditioned MORL selects the preferred valid method under the current state and preferences.

$$M_t^* = \arg\max_{M \in \mathcal{M}_{\text{valid}}(C_t, S_t)} \mathbf{w}_t^\top \mathbf{Q}_\theta(S_t, M; C_t, \mathbf{w}_t)$$

HTN filters what is valid.

Preferences define what matters.

MORL learns what is preferable.

Only the selected method (M_t^*) is decomposed;

Primitive tasks remain under conventional execution.

Conclusion

- HTN defines what is valid. MORL learns what is preferable.
- Dynamic preferences determine what matters now.
- The proposed architecture moves learning to hierarchical method selection, preserving symbolic constraints and conventional low-level execution
- **Next**: evaluate adaptation, generalization, semantic validity, and runtime decision cost.