Reinforcement Learning for Long-Horizon Unordered Tasks: From Boolean to Coupled Reward Machines

Levina K[1,2], Pappas N[1], Karapantelakis A[2], Feljan AV[2], Seipp J[1]

[1] Linköping University, Sweden & [2] Ericsson Research, Sweden

2026 PRL Workshop

Speaker: Kristina Levina

Q-Learning with Coupled Reward Machines (QCoRM)

A QCoRM agent learns unordered tasks in parallel, respecting the global optimality.

Q-learning

An RL agent learns a policy $\pi : S \to A$ by interacting with an MDP $M = \langle S, s_0, A, p, r, \gamma \rangle$ and receiving rewards $r(s, a, s')$.

Objective: to maximise the expected return $q^{\pi^*}(s, a) = \mathbb{E}_{\pi^*}[\sum_{k=0}^{K-t-1} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$.

Tabular Q-learning estimates $Q(s, a)$ from experiences $\langle s, a, s' \rangle$

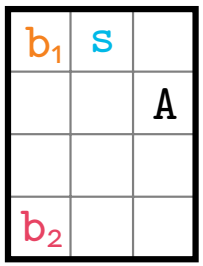
$$Q_{i+1}(s, a) \xleftarrow{\alpha} r(s, a, s') + \gamma \max_{a'} Q_i(s', a'). \tag{1}$$

With enough exploration, $Q(s, a) \to q^{\pi^*}(s, a)$ and $\pi^*(s) = \arg\max_a Q(s, a)$.

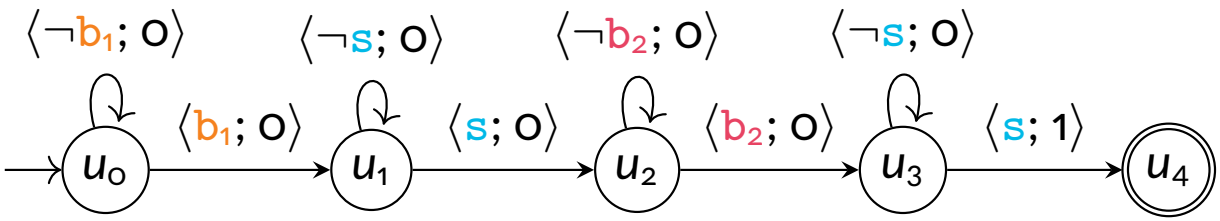
Reward Machines (RMs)

An RM is an automaton over high-level events. Augmenting $\tilde{S} = S \times U$ makes non-Markovian tasks Markovian.

Task 1: Agent A delivers boxes b_1 and b_2 to station s **in consecutive order**.



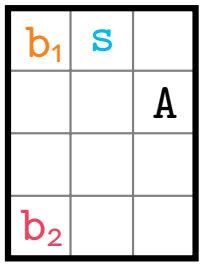
(a) Delivery map.



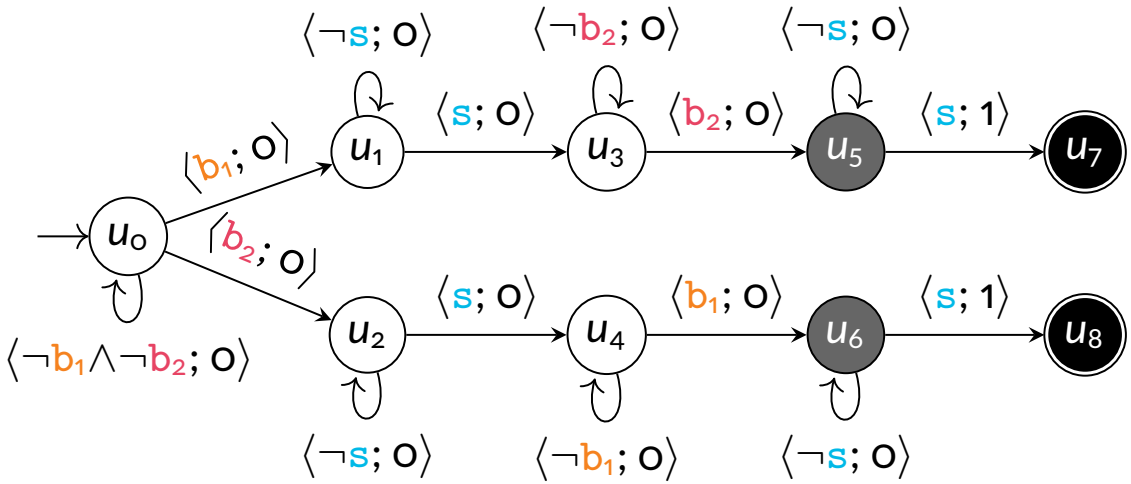
(b) RM.

Challenge: Unordered Subtasks

Task 2: deliver b_1 and b_2 to s **in any order**.



(a) Delivery map.



(b) Boolean RM. Symmetric states in grey and black.

$|U|$ grows exponentially with the number of unordered subtasks, and the agent must learn Q-values over all of $S \times U$.

Solution 1/2: Agenda RMs

Label each RM state with $\langle d, T, \tau \rangle$, where d is an RM depth, T is an agenda (remaining sub-tasks), and τ is a current objective.

States with same labels share one Q-function and can be merged.

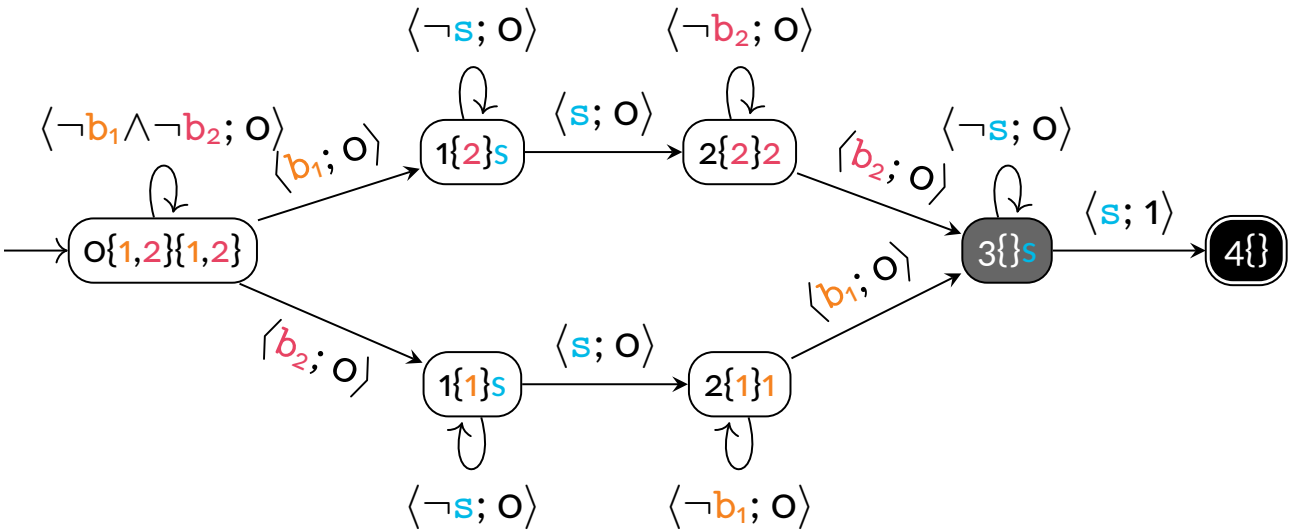


Figure: Agenda RM. Symmetric states merged. Short-hand: $d\{T\}\tau$.

Helpful, but the agent still learns information in the order of $|S \times U|$.

Solution 2/2: Coupled RMs

We further split each state by the subtasks in its agenda T.

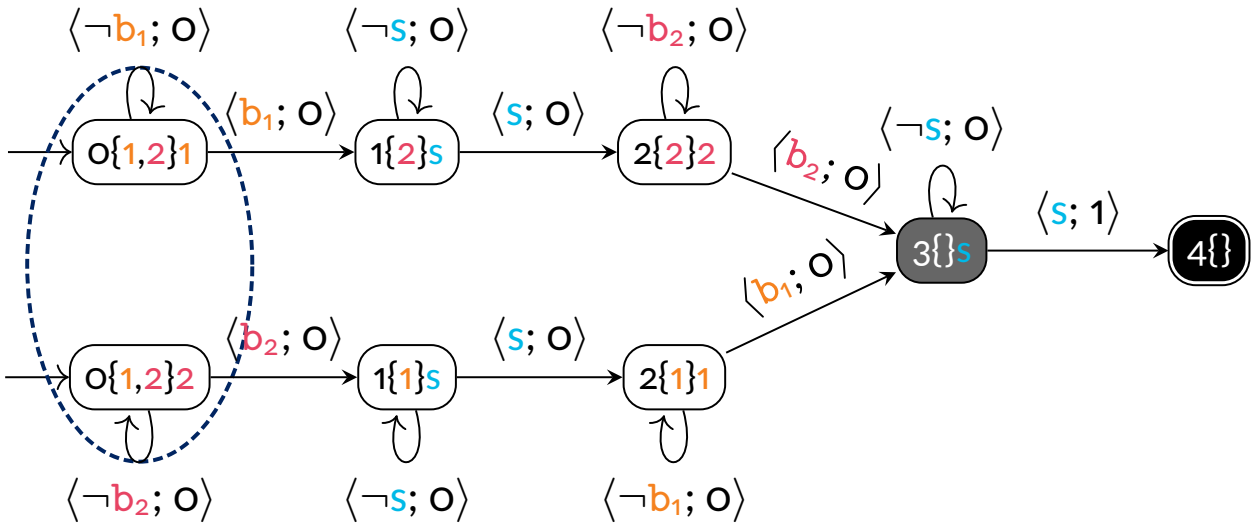


Figure: Coupled RM. The agent occupies the coupled states (dashed) concurrently.

Now, every RM state corresponds to exactly one subtask, and the QCoRM agent only learns to reach b_1, b_2, and s.

QCoRM: Low-Level Policies

The agent learns one low-level policy per subtask τ_k. The MDP state space is now $S \times U'$ with $U' = \mathcal{T} \cup B$ (all subtasks), not the full $S \times U$.

Coupled RM: one policy per subtask $\Rightarrow$ linear in $|U'|$

Boolean RM: one policy per RM state $\Rightarrow$ exponential in $|U'|$

The agent simultaneously updates the Q-values of the coupled states with an agenda T

$$Q_{i+1}(\langle s, \tau_k \rangle, a) \xleftarrow{\alpha} r(\langle s', \tau_k \rangle, a, \langle s, \tau_k \rangle) + \gamma \max_{a'} Q_i(\langle s', \tau_k \rangle, a'), \qquad \forall\, \tau_k \in T. \tag{2}$$

Upon completing τ_ℓ, the update is

$$Q_{i+1}(\langle s, \tau_\ell \rangle, a) \xleftarrow{\alpha} R(K, \tau). \tag{3}$$

QCoRM: Globally Optimal Policies

Problem: low-level policies learned in isolation can be globally sub-optimal.

Solution: on finishing subtask τ, its terminal reward is shaped

$$R(K,\tau) = \begin{cases} \check{R}, & \text{if } \delta K^\tau = 0, \\ \gamma^{\delta K^\tau + 1}\check{R} + \dfrac{\gamma^{1-K^\tau} - \gamma^{\delta K^\tau}}{1-\gamma} r_{\mathrm{mi}}, & \text{otherwise.} \end{cases} \tag{4}$$

where $\delta K^\tau = \max\{\bar{K}^\tau_{\mathrm{op}} - K^\tau,\ K - \bar{K}_{\mathrm{mi}}\}$; $K^\tau/\bar{K}^\tau_{\mathrm{op}}$ is a realised/expected optimal duration of τ; $K/\bar{K}_{\mathrm{mi}}$ is a realised/minimum expected episode length; $\check{R}$ is a positive real-valued final reward; $r_{\mathrm{mi}} \leq 0$ is a minimum immediate reward.

Theorem. QCoRM preserves global optimality of tabular Q-learning under online updates $\bar{K}_{\mathrm{mi}} \xleftarrow{\alpha_K} K$ if $K \leq \bar{K}_{\mathrm{mi}}$ and $\bar{K}^\tau_{\mathrm{op}} \xleftarrow{\alpha_K} K^\tau$.

QCoRM: High-Level Policy Decides the Order

At a coupled state, a high-level policy picks the next subtask by treating the coupled states as a multi-armed bandit. Each arm's η estimates the average steps-to-goal observed after taking that branch.

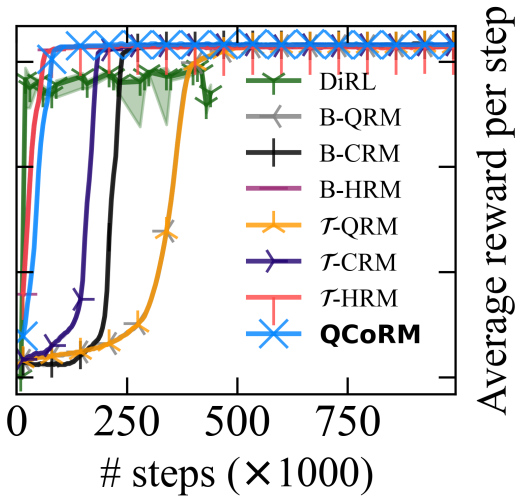
While learning, the agent freely explores the RM; while exploiting, it follows the lowest-η path.

State	o{1,2}1	o{1,2}2	1{1}s	1{2}s	2{1}1	2{2}2	3{}s	4{}
η	12	10	6	9	2	8	1	0

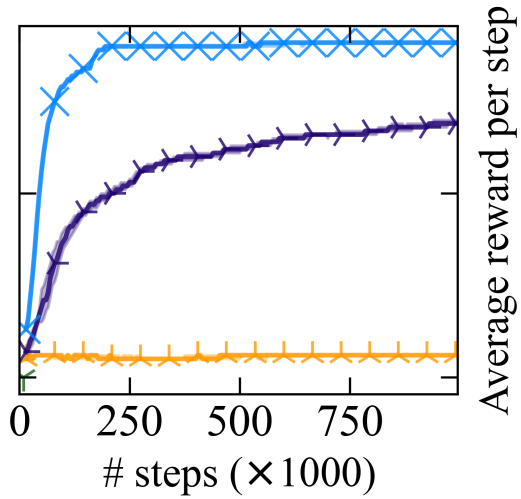
From the two coupled start states, the agent transitions out of o{1,2}2 because η = 10 $<$ 12: collect b_2 first.

QCoRM vs. baselines

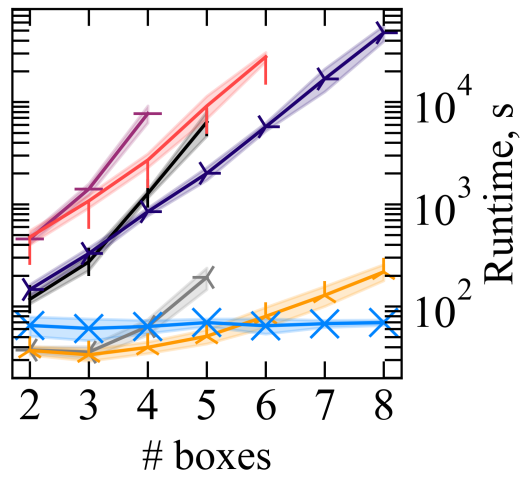
QCoRM converges to an optimal policy, and its runtime scales linearly.



(a) Delivery: 2 boxes



(b) Delivery: 8 boxes



(c) Time to run 10^6 steps

Similar results are obtained in domains with continuous state and/or action spaces.

Thank you — let's discuss!

- Is the global-optimality argument sound?
- Which baselines or domains would convince you?