Decision-Theoretic Planning with Communication in Open Multiagent Systems (Supplementary material)

Anirudh Kakarlapudi[1]   **Gayathri Anil**[1]   **Adam Eck**[2]   **Prashant Doshi**[1]   **Leen-Kiat Soh**[3]

[1]Computer Science Department, University of Georgia, Athens, Georgia, USA
[2]Computer Science Department, Oberlin College, Oberlin, Ohio, USA
[3]Computer Science and Engineering Department, University of Nebraska, Lincoln, Nebraska, USA

A   MONTE CARLO TREE SEARCH

Here, we provide the pseudocode for single agent POMCP Silver and Veness [2010] in Alg 2, as well as an example AND-OR tree in Fig. 5 referenced in Section 4.1.

Algorithm 2 POMCP [Silver and Veness, 2010]

1: **procedure** CREATEPLAN(b)
2:    **for** $traj \in 1, 2, \ldots, \tau$ **do**
3:       $s \leftarrow$ SampleParticle (b)
4:       UpdateTree$(s, 0, \varepsilon)$
5:    **return** $\underset{a \in A}{\operatorname{argmax}} Q(\varepsilon, a)$
6: **procedure** UPDATETREE(s, t, h)
7:    **if** $t \geq H$ **then**
8:       **return** 0
9:    **if** h is a leaf **then**
10:      Expand(h)
11:      **return** Rollout(s, t)
12:   $a^* \leftarrow$ ChooseAction(h)
13:   $s', r, o \leftarrow$ SimulateComm(s, a)
14:   $R \leftarrow r + \gamma *$ UpdateTree$(s', t + 1, hao)$
15:   StoreResults(h, s, a, R)
16:   **return** R
17: **procedure** EXPAND(h)
18:   $B(h) \leftarrow \emptyset, \quad n(h) \leftarrow 0$
19:   $n(ha) \leftarrow 0, \; Q(h, a) \leftarrow 0 \; \forall a \in A$
20: **procedure** CHOOSEACTION(h)
21:   **return** $\underset{a \in A}{\operatorname{argmax}} Q(h, a) + \sqrt{\frac{\log n(h)}{n(ha)}}$
22: **procedure** STORERESULTS(h, s, a, R)
23:   $B(h) \leftarrow B(h) \cup \{s\}$
24:   $n(h) \leftarrow n(h) + 1, , \quad n(ha) \leftarrow n(ha) + 1$
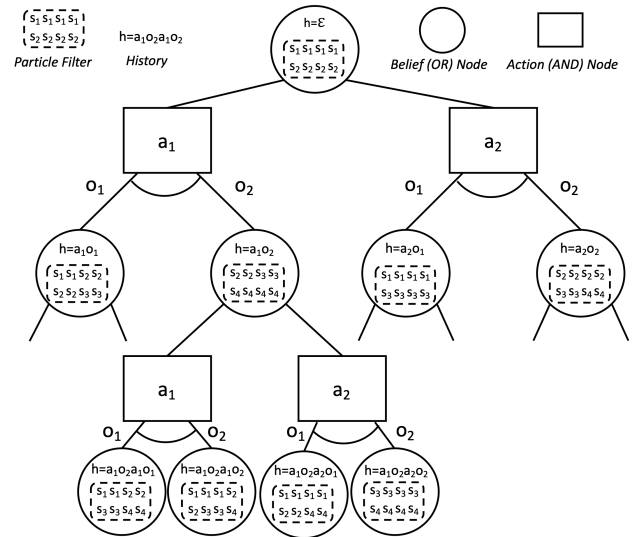25: $Q(h, a) \leftarrow Q(h, a) + \frac{R - Q(h, a)}{n(ha)}$

Figure 5: The first two levels of an example AND-OR tree created by MCTS for a POMDP with 2 actions, 2 observations, and 4 states

B   AGENT ACTIONS PERFORMED IN EXPERIMENTS

In Figs. 6-14, we document the actions chosen by each agent in all three setups for both the I-POMCP-PF$_O$ and CI-POMCP-PF$_O$ algorithms (using communication costs of 0 and 1 to show both ends of the spectrum). Note: for all setups, action 0 = the left most fire, action 1 = the middle fire, action 2 = the right fire, and action 3 = NOOP.

C   MESSAGES SENT IN EXPERIMENTS

In Figs. 15-22, we document the messages sent by each agent in all three setups for CI-POMCP-PF$_O$ algorithms (using communication costs of 0 and 1 to show both ends of the spectrum).
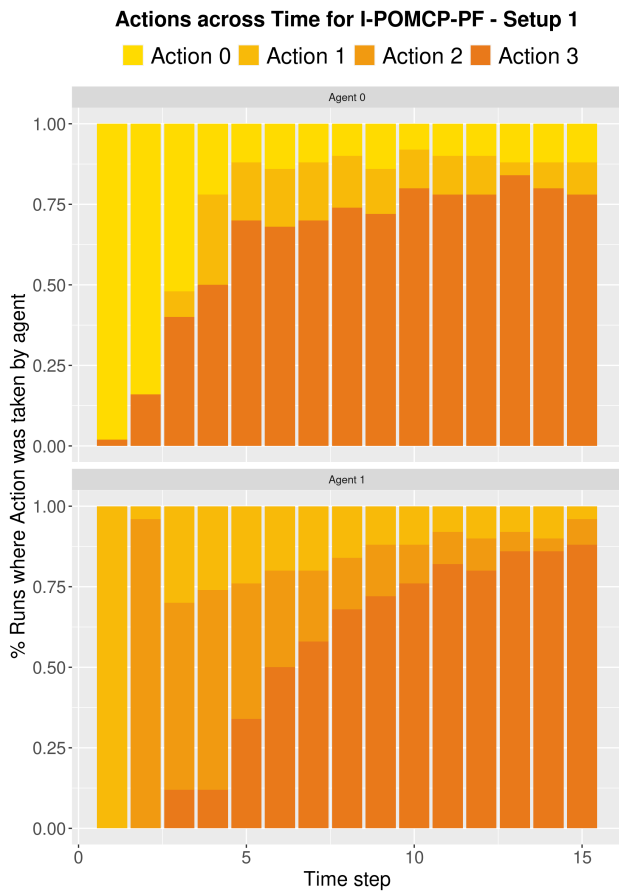
Figure 6: Agent actions using I-POMCP-PF in Setup 1

References

David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *23rd International Conference on Neural Information Processing Systems (NIPS)*, NIPS'10, pages 2164–2172, 2010.
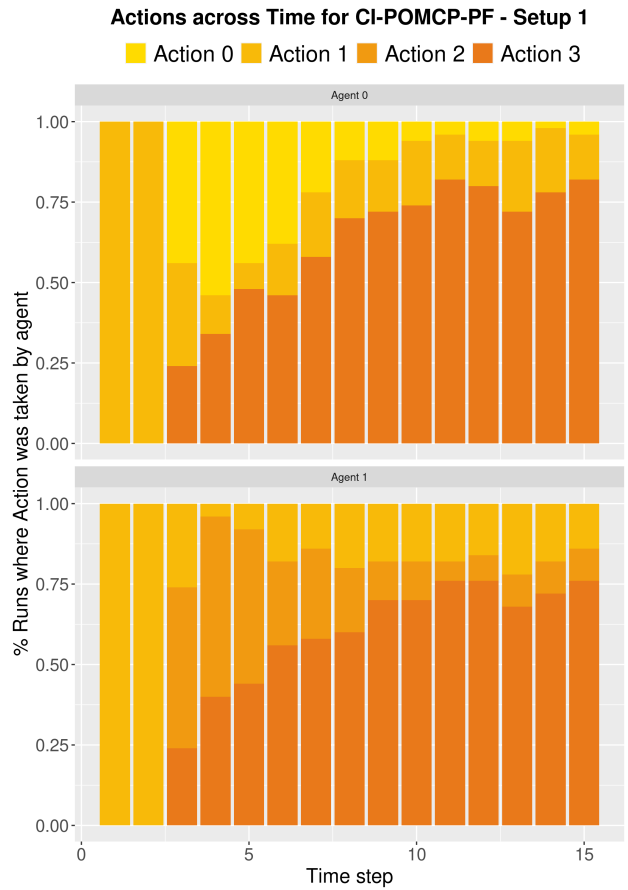
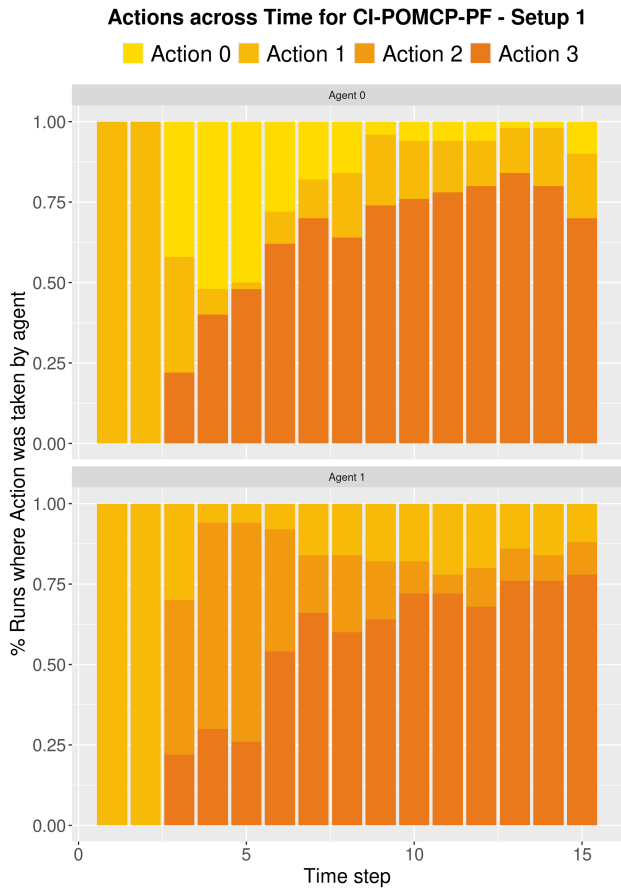Figure 7: Agent actions using CI-POMCP-PF (cost = 0) in Setup 1

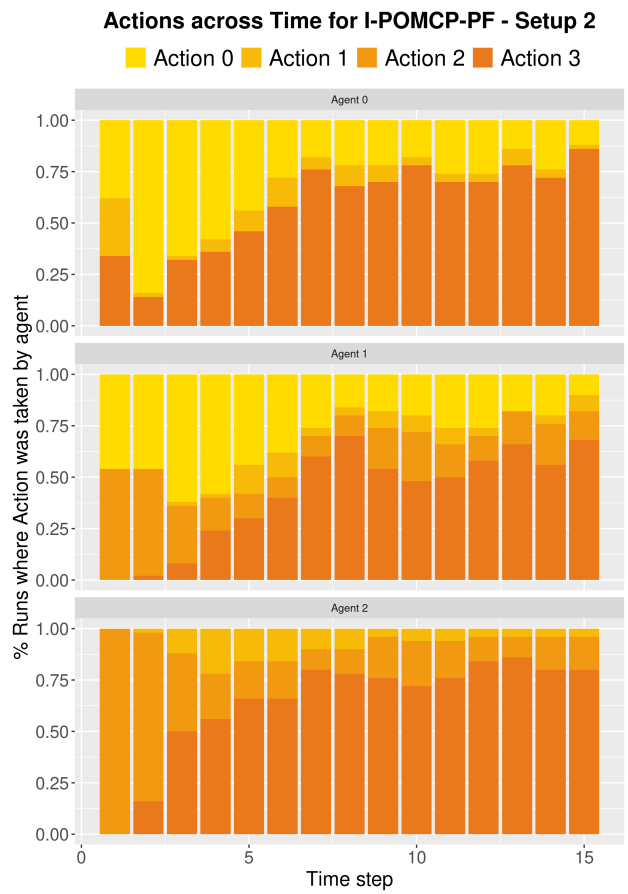Figure 8: Agent actions using CI-POMCP-PF (cost = 1) in Setup 1
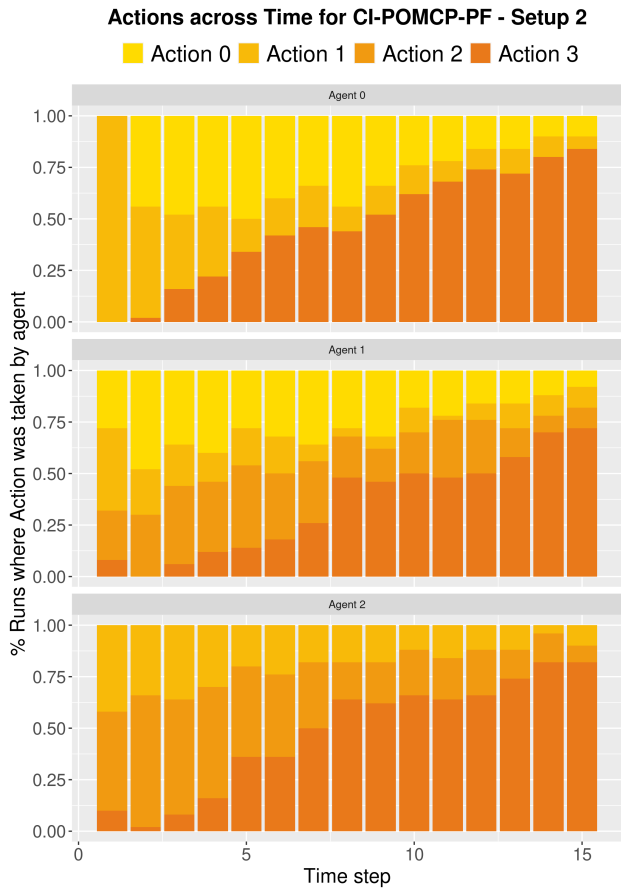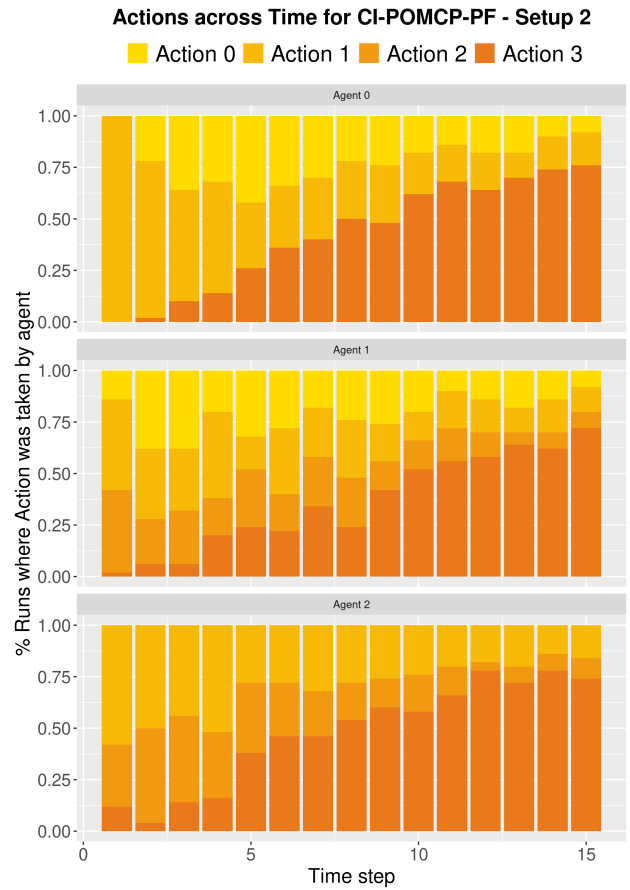

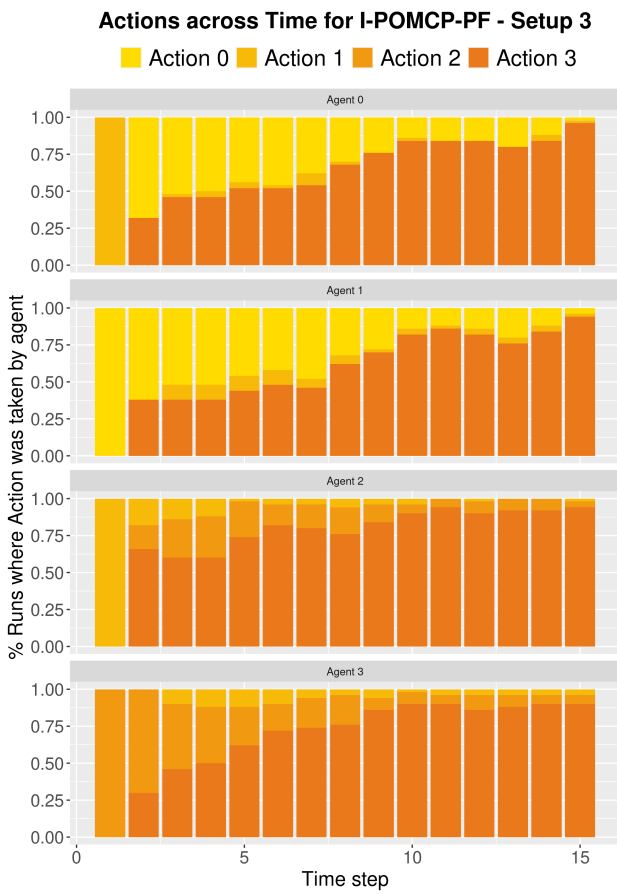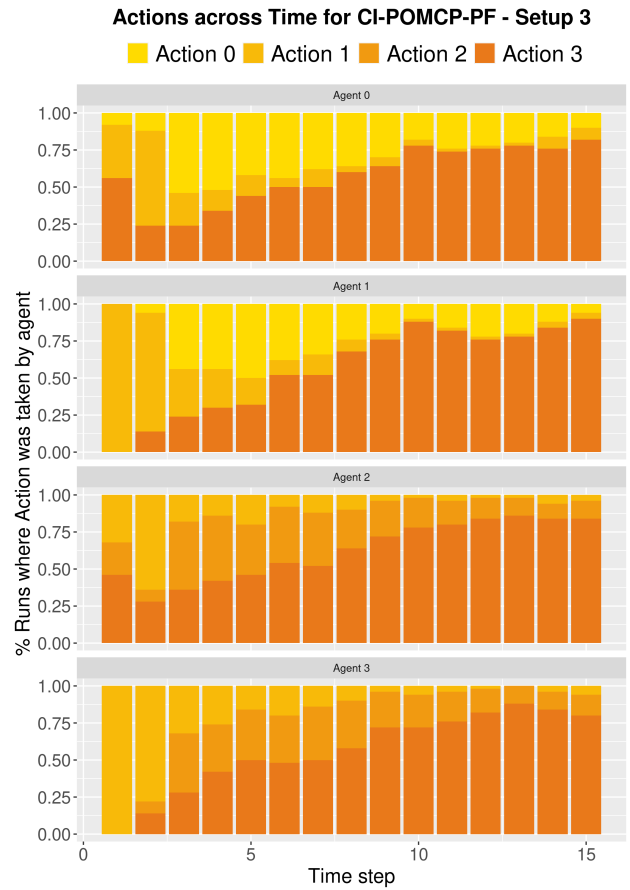Figure 9: Agent actions using I-POMCP-PF in Setup 2

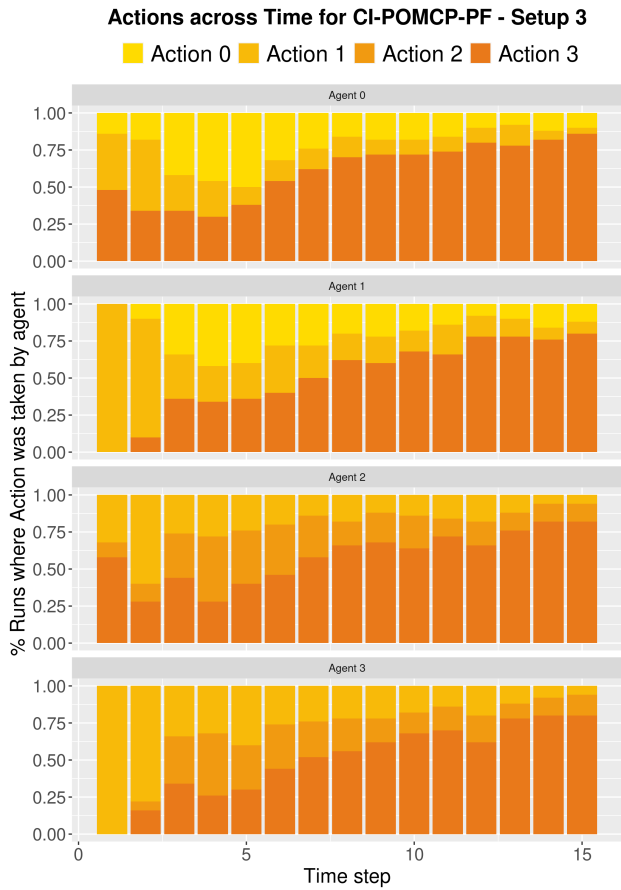Figure 10: Agent actions using CI-POMCP-PF (cost = 0) in Setup 2

Figure 11: Agent actions using CI-POMCP-PF (cost = 1) in Setup 2

Figure 12: Agent actions using I-POMCP-PF in Setup 3

Figure 13: Agent actions using CI-POMCP-PF (cost = 0) in Setup 3

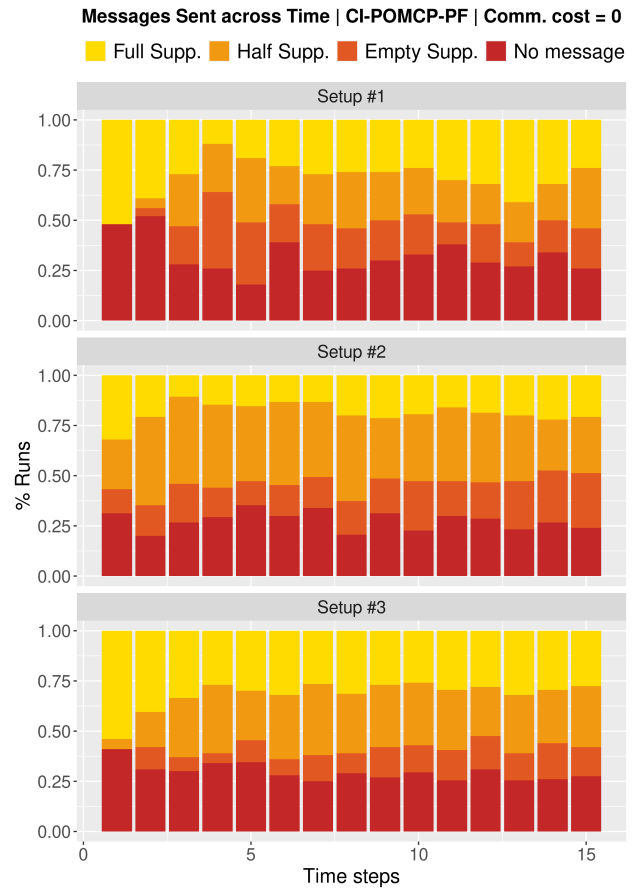Figure 14: Agent actions using CI-POMCP-PF (cost = 1) in Setup 3
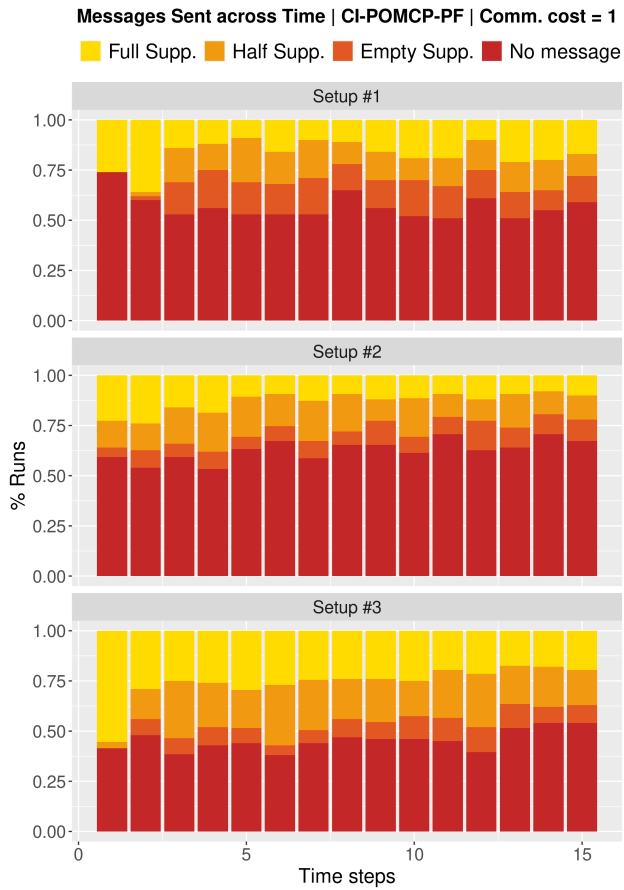

Figure 15: Messages sent using CI-POMCP-PF (cost = 0) in Setups 1-3

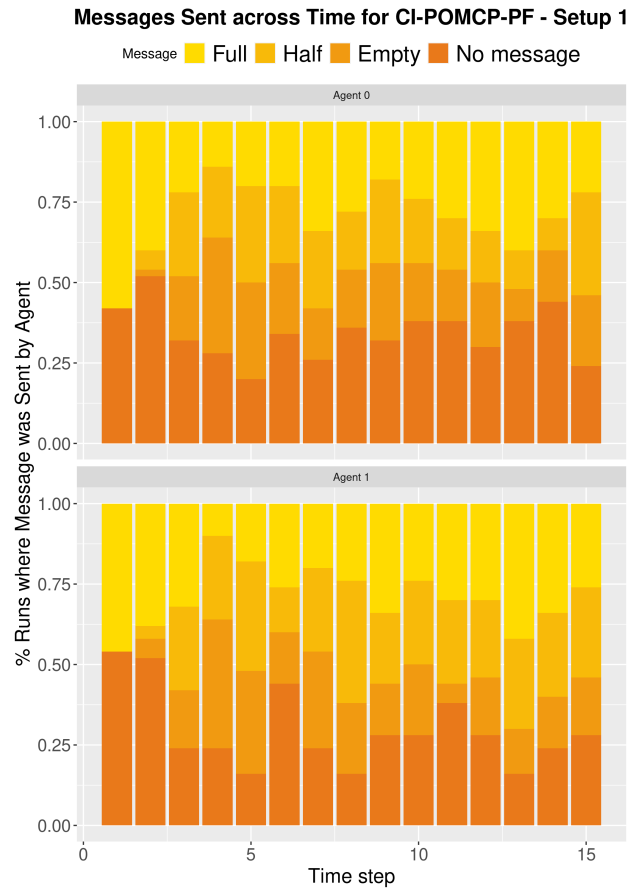Figure 16: Messages sent using CI-POMCP-PF (cost = 1) in Setups 1-3

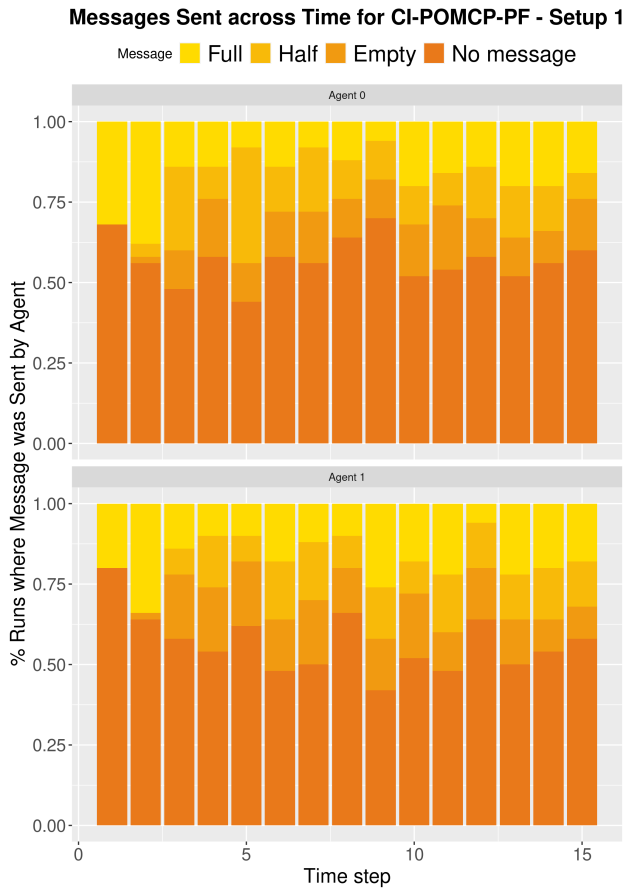Figure 17: Messages sent using CI-POMCP-PF (cost = 0) in Setup 1

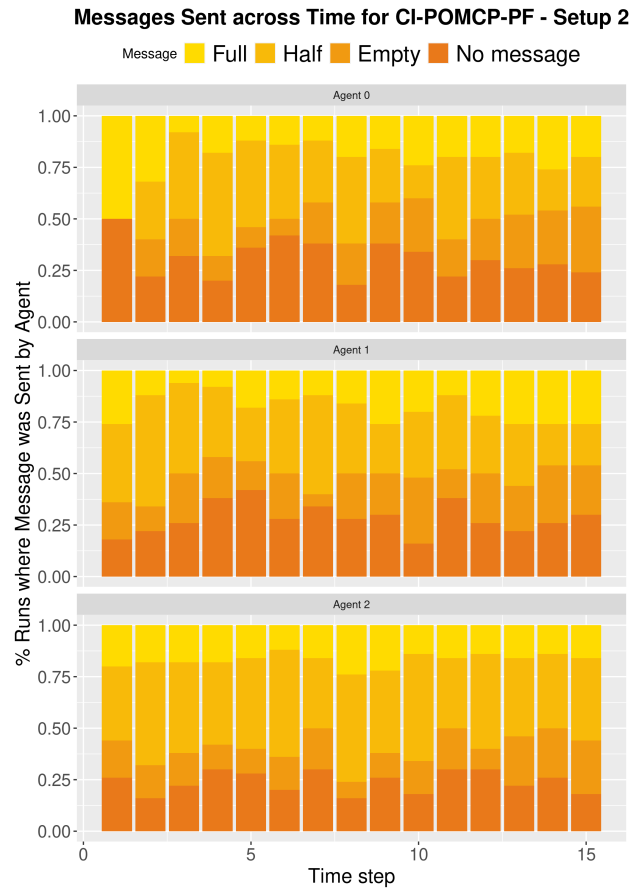Figure 18: Messages sent using CI-POMCP-PF (cost = 1) in Setup 1

Figure 19: Messages sent using CI-POMCP-PF (cost = 0) in Setup 2
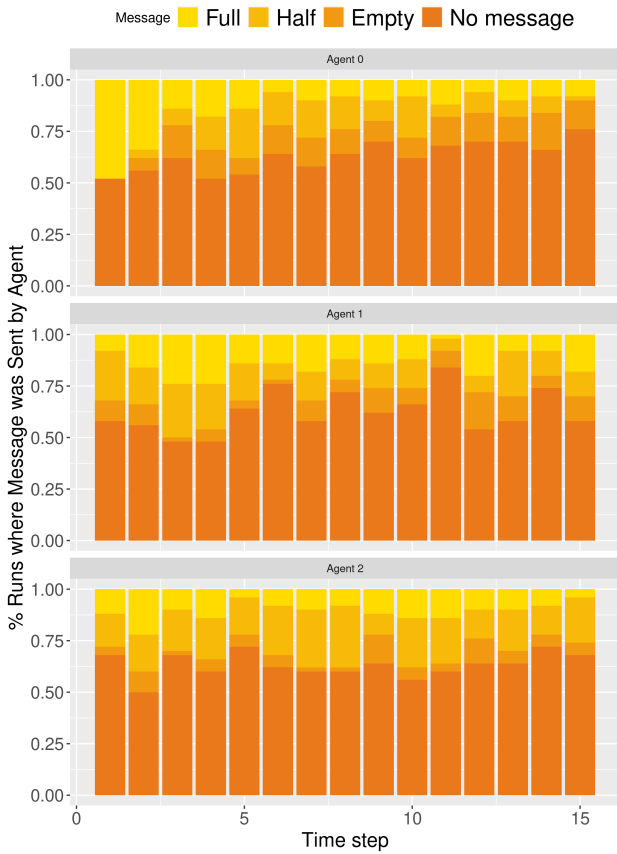
Figure 20: Messages sent using CI-POMCP-PF (cost = 1) in Setup 2
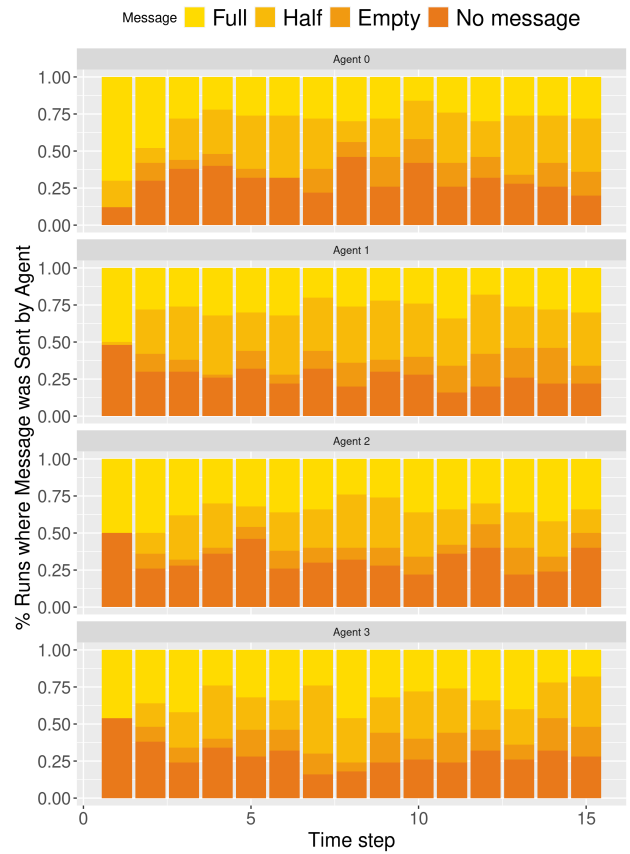
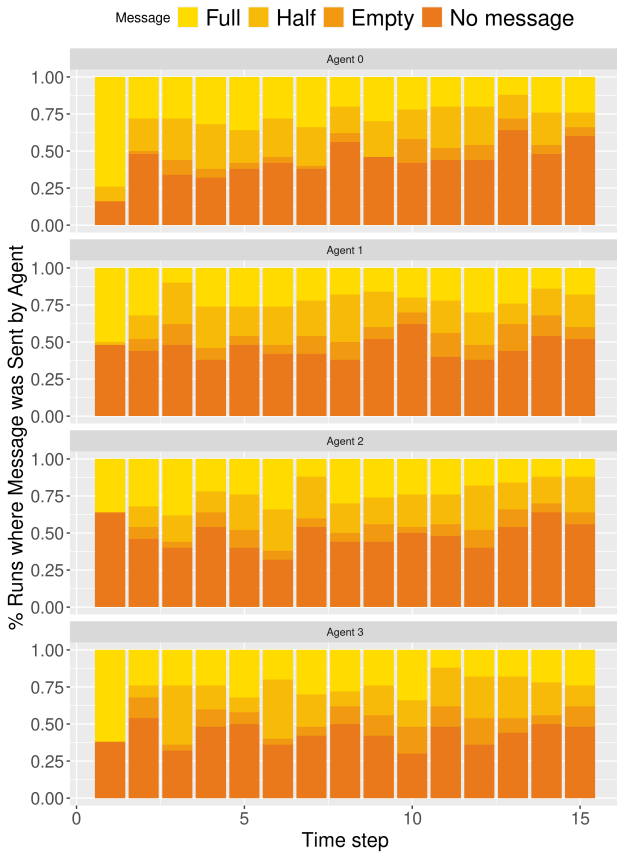Figure 21: Messages sent using CI-POMCP-PF (cost = 0) in Setup 3

Figure 22: Messages sent using CI-POMCP-PF (cost = 1) in Setup 3