Domain	# Actions	# Facts	Best Cost	Expansions × 10ˆ4	Messages × 10ˆ4	# trajectories + restarts	Total Time (sec)
blocks-2-2	26	26	(4.58 / 4.52)	(0.025 / 0.045)	(0.135 / 0.038)	(40 / 40 + 41)	(0.068 / 0.062)
blocks-3-3	129	63	(7.48 / 7.44)	(0.120 / 0.202)	(1.274 / 0.455)	(60 / 90 + 127)	(0.516 / 0.473)
blocks-4-3	315	99	(9.7 / 9.72)	(0.713 / 0.840)	(7.636 / 2.394)	(80 / 130 + 224)	(4.1 / 3.5)
blocks-5-2	422	107	(11.98 / 12.36)	(5.247 / 5.412)	(30.434 / 9.366)	(150 / 380 + 691)	(35.3 / 29.8)
blocks-6-2	746	146	(14.8 / 16.5)	(68.322 / 68.977)	(399.718 / 117.096)	(900 / 2140 + 7057)	(497.8 / 427.7)
depot-2-3	69	49	(11.2 / 11.4)	(0.276 / 0.973)	(2.844 / 3.219)	(90 / 650 + 62)	(1.2 / 3.0)
depot-3-3	149	74	(13.48 / 13.22)	(2.894 / 3.152)	(29.708 / 8.126)	(220 / 160 + 483)	(22.7 / 15.6)
depot-3-4	334	91	(16.76 / 16.76)	(9.273 / 11.765)	(141.352 / 44.033)	(410 / 240 + 2515)	(79.8 / 55.8)
depot-2-5	245	71	(11.4 / 11.14)	(4.342 / 5.050)	(89.343 / 23.417)	(350 / 170 + 1389)	(55.7 / 23.9)
depot-3-5	407	113	(16.78 / 16.78)	(55.045 / 59.236)	(1136.297 / 289.422)	(2330 / 740 + 10437)	(715.4 / 344.8)
logistics-1-3	49	27	(12.78 / 12.62)	(0.371 / 0.191)	(4.208 / 0.164)	(110 / 40 + 102)	(1.1 / 0.217)
logistics-2-3	69	34	(20.38 / 20.78)	(4.203 / 2.516)	(47.815 / 3.494)	(440 / 140 + 792)	(13.9 / 3.5)
logistics-2-4	80	40	(20.82 / 20.84)	(13.437 / 7.174)	(224.140 / 14.711)	(1130 / 170 + 2440)	(54.4 / 10.9)
logistics-2-5	203	70	(27.0 / 21.58)	(122.934 / 40.523)	(2705.937 / 103.567)	(2134 / 220 + 14162)	(848.5 / 81.7)
logistics-3-4	104	48	(26.8 / 27.22)	(152.786 / 109.649)	(2570.205 / 250.775)	(8010 / 770 + 27091)	(720.6 / 186.0)

Table 1: Empirical comparison of the DRTDP and PS-RTDP. Cell format is DRTDP / PS-RTDP.

own Q-values (line 7), and remains responsible for the trajectory. If a^* is a public action, however, the agent uses the value function update mechanism, and selects the next agent to take responsibility as in Algorithm 2 (line 27). As all messages are sent by the global update mechanism and the next agent selection process, reducing calls to these phases reduces the number of messages significantly.

PS-RTDP is no longer guaranteed to converge. One reason is that an agent chosen to advance the trajectory may get stuck in a loop of private actions, without being able to reach the goal, or execute any public action. For example, in the Logistics domain, agent 1 may unload a package at location A. As all agents currently have Q-values of 0, the tie breaking mechanism decided that an agent with no access to A takes the trajectory. That agent has no public actions available, and can hence only advance the trajectory by private driving actions, never releasing responsibility to other agents. We therefore add a private cycle detection mechanism (line 4). If a cycle is detected, we restart the trajectory from s_0. The sensitivity of the cycle detection is important, as cycles may arise due to stochastic effects. Detecting a cycle too soon may make the algorithm ignorant to a possible path towards the goal. Detecting a cycle too late may cause the algorithm to be much slower than the complete DRTDP.

4 Empirical Analysis

We now provide empirical analysis on domains adapted from the CPPP literature, comparing the complete DRTDP and the approximate DRTDP algorithms, implemented in Python. All experiments were run on Google cloud running a Xeon CPU with two 2GHz cores, and 1.8GB user RAM. The domains were adapted by adding stochastic effects. For example, in the blocks world domain, when picking a block it may either be successfully picked, stay where it was, or fall on the table. We also added different capabilities and success probabilities for different agents for completing different tasks, requiring smarter policies for optimization. For example, we added block types, where each arm can lift only a subset of block types, with varying success probabilities. On each problem we stop the algorithms after 10 RTDP trajectories, and estimate the average cost over 50 policy executions. We terminate once the policy stops improving.

Table 1 shows the results. In many domains the approximate DRTDP generated orders of magnitude fewer messages than the complete version, while converging to policies of similar quality. In all domains both methods expanded a similar number of states, but often the approximate version requires fewer goal reaching trajectories, not counting restarts after loops, that can be much shorter.

In the blocks domain, the approximate RTDP generated about one third of the messages. On the other hand, it often required many more trajectories to converge, especially for the larger problems, and hence resulted in only slightly lower runtime. Only in the largest blocks problem, PS-RTDP resulted in significantly higher average cost. This is because in blocks world there is a relatively small number of private actions, and hence the public version is not very different.

In Logistics, the PS-RTDP sends in many cases fewer than $\frac{1}{10}$ of the messages, and is hence considerably faster on many problems. In Logistics, PS-RTDP also required much fewer trajectories to converge. In Depot, as in blocks world, the number of messages sent by PS-RTDP is about one third of the messages sent by the complete DRTDP. In this domain, however, the approximate version required about half the time before converging, especially on larger problems.

5 Conclusion and Future Work

We presented two algorithms for CPPP in stochastic environments. DRTDP is a distributed, privacy preserving adaptation of RTDP, which is identical in execution to RTDP on the joint problem. PS-RTDP is an approximation of DRTDP that shares only public changes between agents to reduce the amount of messages during planning. Experiments show that PS-RTDP sends significantly less messages than DRTDP, sending an order of magnitude fewer messages in some cases. Future research can advance multiple trajectories simultaneously, allowing agents that do not currently advance a trajectory to start a new one. sequential action execution. We will investigate an extension to concurrent execution.

ea fuga nihil illo excepturi similique enim alias obcaecati voluptatum, laborum itaque qui saepe rerum?