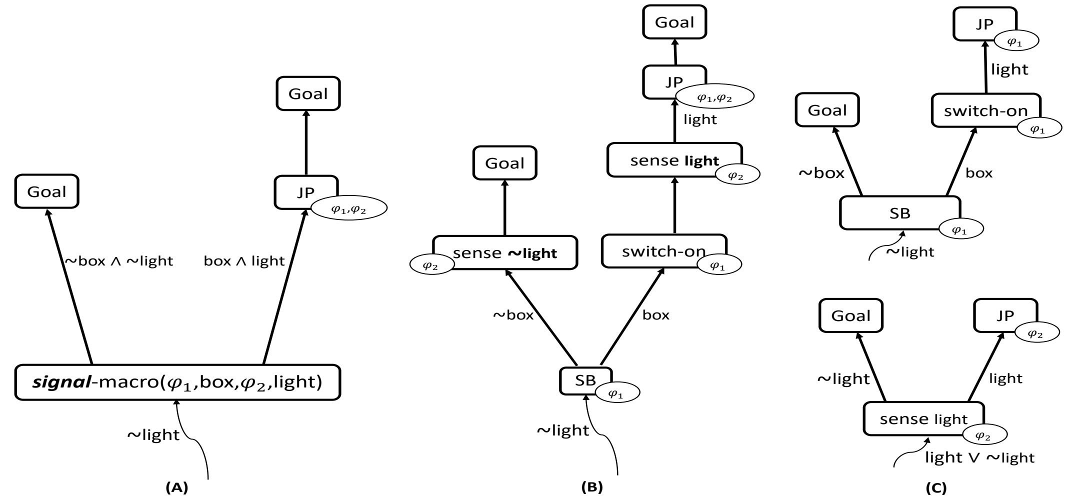

Figure 3: *Signaling* in the QDec-FPS planner. (A) A team plan with the signaling macro action. (B) The team plan following the macro expansion. (C) Projected trees for φ_1 (top) and φ_2 (bottom).

able to deduce that the preconditions of some actions are known to the acting agent, even though they are.

5 Empirical Evaluation

We now examine the applicability and scalability of QDec-FPS by comparing it with QDec-FP on 3 multi-agent planning domains. Some problems in each domain were modified to require *signaling*. Both QDec-FP and QDec-FPS are implemented in C#, and were run on a Windows 10, 64 bit machine with *i*7 processor, 2.8GHz CPU, and 16Gb RAM. IMAP was not considered, as QDec-FP was shown to scale better than IMAP (**?**).

We experiment with the following three domains:

Box-Pushing (BP): Boxes located in a grid-like structure. Each box is to be pushed to its destination location outside the column the box appears in (**?**). A light box can be pushed by a single agent co-located with the box while a heavy box requires two co-located agents. Agents can be non-homogeneous, i.e., different agents can observe and push different boxes.

Table-Mover (TM): TM consists of a number of tables and rooms, and agents that can move between connected rooms (**?**). The tables' locations are uncertain initially, and agents must move them to their goal locations. Agents can be non-homogeneous in their sensing and manipulation abilities. All table manipulation actions are collaborative: *2move-table*, *2lift-table*, *2drop-table*.

Rovers: Multiple rovers navigate a planet surface, finding samples and communicating them back to a lander (**?**). Two rovers must simultaneously collect the rock sample, while a single agent can sample soil as well as take images of certain objectives on its own. Coordination points include locations (waypoints) which are accessible to multiple rovers. Rovers communicate sampled soil/image/rock data to the lander that exists at a certain waypoint. A rover navigates between two waypoints and must be present at the corresponding to sample. Availability of data to sample at a waypoint is unknown to the rovers initially. In this modified domain, our schema requires two rovers working jointly to collect rock samples. After taking measurements, the rovers must broadcast them back to the lander. In this domain in general, a rover has fewer public actions (approx. 46% on average), but a relatively complex internal planning problem, including navigation, soil and rock sampling, and image capturing.

Table 1 compares QDec-FPS and QDec-FP based on policy quality (*max-width*, *max-height*), runtime (*time*), and the number of *backtracks* required. *max-width* and *max-height* refer to maximum number of branches and the maximum height of all individual solution trees obtained for the agents. The number of branches is also indicative of the number of sensing actions performed, as branching occurs following an observation. The planner backtracks when at least one of the single-agent problems, obtained by decomposing τ_{team}, is unsolved by CPOR. Within each domain in the table, dashed lines separate three problem classes: homogeneous agents, non-homogeneous agents, and non-homogeneous agents that require signaling. To handle signaling, QDec-FPS adds macros, which may have a significant overhead. Therefore, these macros were only added in problems that require signaling. The decision whether to add macros was done *manually*. In the future, we will automatically detect whether signaling is needed.

In BP, QDec-FPS scales much better than QDec-FP in problems with three to five agents. Increasing the number of objects had minor impact on QDec-FPS running time, as opposed to QDec-FP. For many problems, QDec-FPS needs to backtrack fewer times than QDec-FP, and as a result, it finds solutions faster. On the other hand, increasing the number of agents has an adverse effect on QDec-FPS. In fact, instance B3 in the BP domain with nine *identical* agents, was quite rapidly solved by QDec-FP, while QDec-FPS times out. This is because the new translation makes the team problem much harder to solve. Thus, QDec-FP finds a team plan quickly, and when agents are identical, it is usually very easy to fix the team problem by adding any needed sensing action. For that reason, in the case of homogenous agents, we see no backtracks in any of the problems. On the other hand, prob-

Domain	Ins (#agt)	Objects	Max-width		Max-height		Time (sec)		BT	
			fp	*fps*	*fp*	*fps*	*fp*	*fps*	*fp*	*fps*
BP	B1 (3)	16	8	5	23	18	3.59	**2.91**	0	0
	B2 (4)	16	12	10	19	19	**5.3**	6.1	0	0
	B3 (9)	36	64	*	24	*	**25.3**	*	0	*
	B4 (3)	11	4	4	14	11	16.39	**1.17**	9	0
	B5 (3)	12	6	8	16	15	13.65	**2.9**	4	0
	B6 (3)	12	-	6	-	12	-	**13.58**	47^+	4
	B7 (3)	12	8	8	18	19	158.89	**3.87**	41	0
	B8 (3)	12	8	8	17	21	111.6	**4.05**	26	0
	B9 (3)	13	16	14	21	19	121.42	**5.6**	19	0
	B10 (3)	16	16	15	26	29	155.83	**9.69**	33	0
	B11 (5)	20	*	24	*	32	*	**75.21**	*	1
	B12 (5)	20	*	24	*	37	*	**365.9**	*	6
	B13 (2)	10	*na*	2	*na*	6	*na*	**1.06**	*na*	0
	B14 (2)	12	*na*	2	*na*	2	*na*	**1.20**	*na*	1
	B15 (3)	12	*na*	4	*na*	12	*na*	**2.49**	*na*	0
	B16 (3)	13	*na*	4	*na*	7	*na*	**5.5**	*na*	1
	B17 (3)	13	*na*	4	*na*	6	*na*	**8.9**	*na*	2
	B18 (3)	14	*na*	6	*na*	16	*na*	**4.04**	*na*	0
	B19 (4)	15	*na*	8	*na*	14	*na*	**17.6**	*na*	2
TM	T1 (3)	10	8	7	20	16	**2.68**	2.78	0	0
	T2 (4)	12	16	13	21	17	**7.84**	8.26	0	0
	T3 (4)	15	12	16	34	26	**8.74**	12.39	0	0
	T4 (5)	14	19	22	22	24	**20.5**	43.58	0	0
	T5 (5)	16	12	14	32	25	**9.7**	11.2	0	0
	T6 (3)	8	2	2	8	8	11.01	**0.61**	7	0
	T7 (3)	10	8	7	20	16	**34.5**	41.8	8	8
	T8 (3)	10	8	8	24	22	37.87	**22.45**	9	4
	T9 (3)	10	-	-	-	-	140.32	**29.73**	31	6
	T10 (4)	12	16	16	23	22	**6.68**	152.18	0	14
	T11 (5)	16	16	16	30	35	274.5	**56.15**	27	3
	T12 (2)	10	*na*	2	*na*	9	*na*	**1.7**	*na*	0
	T13 (2)	12	*na*	4	*na*	17	*na*	**6.81**	*na*	0
	T14 (3)	12	2	2	16	11	17.59	**2.32**	9	0
	T15 (3)	13	*na*	4	*na*	14	*na*	**7.80**	*na*	0
	T16 (3)	14	*na*	4	*na*	19	*na*	**11.68**	*na*	1
Rovers	R1 (1)	12	2	2	8	8	3.8	3.8	0	0
	R2 (2)	14	2	2	9	8	5.4	**5.3**	0	0
	R3 (2)	13	4	4	15	15	9.3	**8.9**	0	0
	R4 (2)	17	12	12	34	23	**35.8**	38.6	0	0
	R5 (3)	18	12	12	28	32	**51.5**	54.1	0	0
	R6 (3)	21	30	30	36	29	136.6	**111.9**	0	0
	R7 (3)	23	98	81	50	45	567.9	**233.2**	0	0
	R8 (3)	13	2	2	7	6	57.9	**5.9**	3	0
	R9 (3)	14	4	4	12	12	113.8	**10.2**	4	0
	R10 (3)	14	4	4	11	11	314.5	**89.3**	11	3
	R11 (3)	19	-	12	-	12	325.1	**41.1**	6+	0
	R12 (4)	15	4	4	12	11	47.2	**16.0**	1	0
	R13 (4)	20	12	12	25	19	241.8	**49.1**	3	0
	R14 (2)	16	*na*	2	*na*	9	*na*	**3.6**	*na*	0
	R15 (2)	18	*na*	2	*na*	10	*na*	**5.89**	*na*	0
	R16 (2)	16	*na*	4	*na*	13	*na*	**34.17**	*na*	0
	R17 (2)	20	*na*	4	*na*	14	*na*	**17.68**	*na*	0
	R18 (3)	17	*na*	2	*na*	7	*na*	**10.17**	*na*	0
	R19 (3)	18	*na*	4	*na*	13	*na*	**12.93**	*na*	0
	R19 (3)	14	*na*	2	*na*	7	*na*	**6.62**	*na*	0
	R20 (3)	19	*na*	8	*na*	16	*na*	**36.74**	*na*	0
	R21 (4)	20	*na*	6	*na*	11	*na*	**67.79**	*na*	0
	R22 (4)	22	*na*	6	*na*	12	*na*	**68.21**	*na*	0

Table 1: Comparison of QDec-FP and QDec-FPS planners. *Ins (#agt)*: instance number and number of acting agents. *Object*: the number of objects in each problem. *BT*: number of planner backtracks. *: time out. '-': planner could not solve or breaks down. *na*: problem not applicable to a solver. *fp*: QDec-FP. *fps*: QDec-FPS. The best approach, based on time only, is shown in bold.

lems B11 and B12 were solved by the QDec-FPS planner but were unsolved by QDec-FP. This is most likely due to the high number of required backtracks. We can conclude that for simpler problems with identical agents, QDec-FP scales better with the number of agents, but for more complex problems, QDec-FPS is required.

Unlike BP, in TM each public action is a collaborative action. Similar to the BP domain, when backtracking is not required to solve an instance, for example, the simpler instances T1 to T5, the QDec-FP solver takes less time to solve it than that of QDec-FPS. When problems are more complex and backtracking is required, e.g., for the instances T6 and T11, QDec-FPS is faster.

In Rovers, we see a similar trend as BP and TM. For the simple instances like R1 to R7, the performance of the planners is mixed. When backtracking maybe required since the rovers are non-homogeneous (instances R8 to R13), QDec-FPS outperforms QDec-FP on all instances. Instances with more objects are solved relatively easily by QDec-FPS.

Problem T9 is interesting because it is unsolvable. This requires the planner to rule out all possible solutions. Because the team problem QDec-FPS generates has fewer solutions, it was able to rule them out much faster using six backtracks, compared to 31 for QDec-FP.

The table clearly shows that QDec-FPS generates smaller trees across all domains. A closer inspection of the policies also shows that the number of *noops* in its solution is smaller.

To test signaling, we added new instances to all domains that cannot be solved without signaling (B13-B19, T12-T16, R14-R22). These instances cannot be solved by QDec-FP (shown as *na*). In the BP instances, moving from 2-3 agents to 4 increased runtime by an order of magnitude. This is likely due to the number of optional pairwise signaling added with each agent. However, adding objects did not much impact runtime. In the TM domain, we see a more mixed picture. Adding agents or objects can increase runtime, although not always (e.g. T14 vs. T12 and T13). T14 is particularly interesting because it can be solved without signaling (hence QDec-FP solves it with 9 backtracks), but it is solved even faster with signaling. For this instance there are three agents, φ_1, φ_2, and φ_3 such that φ_1 and φ_3 together are capable of solving this problem without signaling. As we purposely placed φ_3 farther from the table's location, QDec-FPS generated a plan where φ_1 signaled to φ_2 the table's location and they achieved the goal together, without φ_3. This solution was generated much quicker and with no backtracks. Signaling in Rovers is similar to BP. Adding more agents makes the problem harder, especially when we move to 4 agents, whereas the effect of objects is less clear. The number of instances of signaling macros in the team solutions range from 1 to 4 on larger problem instances.

6 Conclusion

QDec-FPS uses a factored approach that solves a MAP problem by reducing it to multiple single-agent planning problems. By reasoning about individual agents' knowledge, it generates more informed team plans that lead to fewer backtracks and better scalability. This ability allows it to also support signaling, a form of implicit communication through the state of the world, and thereby to solve problems unsolvable by previous planners.

Acknowledgements. This work was supported by ISF Grants 1651/19 and 1210/18, by the Israel Ministry of Science and Technology Grant 54178, and by the Lynn and William Frankel Center for Computer Science. Quod nisi nihil ipsam, vero accusamus doloribus ab quia, tempora velit tempore iste quae voluptatum possimus, voluptates perferendis sed nulla aliquid a dignissimos saepe blanditiis, cumque architecto molestiae tenetur omnis totam laborum impedit eum itaque ipsa?Rem incidunt minima provident explicabo magni eius saepe rerum aspernatur dolore, fugit dolorum amet architecto, quo velit similique culpa molestiae, rem eveniet saepe vero sapiente laboriosam quos illo id, veniam tempore deserunt in alias.