Domain	GPPP	MAPR -p	PMR	MAPlan/ FF+DTG	PSM- VRD	DPP
blocksworld	12	**20**	**20**	**20**	**20**	**20**
depot	11	0	0	13	17	**19**
driverlog	14	**20**	19	17	**20**	**20**
elevators	**20**	19	19	11	12	**20**
logistics	**20**	19	0	18	18	**20**
rovers	19	19	**20**	**20**	12	**20**
satellites	18	**20**	19	**20**	18	**20**
sokoban	9	0	6	**18**	**18**	17
taxi	**20**	0	19	**20**	0	**20**
wireless	3	2	7	4	0	**9**
woodworking	18	0	0	16	**19**	**19**
zenotravel	**20**	**20**	18	**20**	13	**20**
sum	184	139	147	197	167	**224**

Table 1: Coverage results for a timeout of 30 minutes over the CoDMAP instances.

Domain	Action count Public	Private	Projection	Depth	Time
blocks	2448	0	23256	2	0.057
depot	6134	0	74174	3	0.142
driverlog	56448	1248	1110720	2	2.66
elevators	768	112	447747	11	498.2
logistics	264	52	480	3	0.019
rover	1196	1642	1196	3	1.53
satellite	3139	42099	3139	2	19.9
sokoban	1856	0	4172	2	0.665
taxi	153	0	153	2	0.001
wireless	5814	0	113414	3	7.98
woodworking	8226	0	8226	2	0.168
zeno-travel	288	1098	1008	3	0.266

Table 2: Projection computation metrics over a large problem from each domain.

Experimental Results

We experimented with benchmarks from the 2015 CoDMAP competition (**?**).[2] We run DPP on a 2.66 GHz machine with 4 cores and 8 GB of memory. Note taht while the computer had multiple cores, we implemented DPP to run on a single core. DPP was implemented using the Fast-Downward (FD) planner (**?**) for the high-level planning and the FastForward (FF) planner (**?**) for extending high-level plans. FD was configured to use preferred operators, deferred heuristic evaluation, and two heuristics: FF and a landmark-based heuristic. This is a common configuration used also by the LAMA planner (**?**). We compare DPP with the best performing and most relevant privacy-preserving planners: GPPP (**?**), MAPlan/FF+DTG (an extension of the MAFS algorithm (**?**) using the FF and DTG heuristics together), MAPR-p, PMR (**?**), and PSM-VRD (**?**; **?**). Details on these planners can be found in the CoDMAP website.

the domains in the competition. Our projection approach – DPP – solves more problems than all other approaches. In 7 of the 12 domains, DPP solved all instances, and in 2 more it solved 19 out of 20 problems. In all domains DPP is either the best performing algorithm (in terms of coverage) or within 3 problems of the best performing algorithm. This is in contrast to its closest rivals – GPPP and MAPlan – where GPPP is non-competitive in 4 domains and MAPlan in 2 domains. Thus, DPP shows robustness across all domains. wireless. This is because the wireless domain has many dead-ends that are difficult to escape. Both FF and FD do not have a sufficiently strong mechanism to detect and avoid dead-ends. Still, in this domain too, DPP is the best-preforming planner.

DPP and all other planners is that once the projection was constructed we use an off-the-shelf classical planner, rather than a coordinated joint search mechanism. Even the construction of the DP projection can be easily distributed, having each agent add its projected actions and dependency facts. Our results point to the strengths of the mature classical planners over the rather new joint search mechanisms, such as GPPP (**?**) and MAFS (**?**), which is used by MAPlan.

[2] `http://agents.fel.cvut.cz/codmap/results/`

We further analyze the actual size of the DP projection and runtime of creating it (using Algorithm 2). We chose a challenging problem on each domain, and compute various metrics during the projection construction. Table 2 reports the size of the original problem, in terms of the number of public and private actions of all agents, and the number of actions in the projection. We further report the maximal depth of a regression tree in our experiments, as well as the time it took to compute the projection in seconds.

Obviously, in domains where there are no private actions, our method is extremely fast. Satellite instances took much time, mainly due to the large number of private actions. Still, there is minor dependencies between the actions, resulting in shallow regression trees with a large branching factor.

more difficult instances of this domain contain private floors, that only a single elevator can reach. As boarding and exiting various quantities of passengers in floors requires different actions, there are many possible combinations for achieving the preconditions of actions, as reflected by the maximal depth of the regression trees in this domain. Hence, this is the only domain in the set of benchmarks that poses a true challenge to the regression tree computation.

ignore the identity of passengers, and focus only on their number, we can learn more general regression trees faster. Another method for scaling up would be to use an approximate regression mechanism, by, e.g., regressing only one precondition at a time.

Conclusion

process in which some information about private dependencies is shared while privacy is preserved. We showed that the DP projection also preserves a strong form of privacy in which the cardinality of private objects of a given type cannot be inferred by any adversary agent. The benefit of our DP projection is demonstrated in a planner that effectively uses the DP projection to solve more benchmark problems than any other state-of-the-art privacy preserving planner.

the DP projection as a heuristic for guiding the search of a complete solver, such as MAFS.

Acknowledgments: We thank the reviewers for their useful comments. This work was supported by ISF Grant 933/13, and by the Helmsley Charitable Trust through the Agri-

cultural, Biological and Cognitive Robotics Center of Ben-Gurion University of the Negev.