BLP) was run on a 2.8GHz Quad-Core Intel Mac with 2GB of RAM with a time limit of 3 hours. We used the best available version of the bilinear program approach which was the iterative best response version with standard parameters. This is a generic solution method which does not perform as well as the more specialized approaches in [22], but we do not expect results to differ by more than a single order of magnitude. We do not compare to the coverage set algorithm because the bilinear programming methods have been shown to be more efficient for all available test problems.

We provide values for the exhaustive variant, exh, on small problems and constraint optimization formulation, COP, for all problems. We tested our algorithms on six benchmarks: recycling robot, meeting-in-a-grid 3x3 and 8x8; and navigation problems\(^2\). These are the largest and hardest benchmarks we could find in the literature. We compare our algorithms with: GMAA\(^*\)-ICE [25], IPG [1], and BLP. The GMAA\(^*\)-ICE heuristic search variant outperforms other generic exact solvers such as (G)MAA\(^*\) [25]. The IPG algorithm is a competitive alternative to the GMAA\(^*\) approach and performs well on problems with reduced reachability [1]. Results for GMAA\(^*\)-ICE were provided by Matthijs Spaan and as such were conducted on a different machine. Similarly, results for IPG were collected on different machines. As a result, the timing results for GMAA\(^*\)-ICE and IPG are not directly comparable to the other methods, but are likely to only differ by a small constant factor from those that would be obtained on our test machine.

The results can be seen in Table 1. In all benchmarks, the COP variant of MPS outperforms the other algorithms. The results show that the COP variant produces the optimal policies in much less time for all tested benchmarks. For example, in the meeting in a 3x3 grid problem for \(T = 5\): the COP variant computed the optimal policies approximately 33, 4558 and 4580 times faster than the GMAA\(^*\)-ICE, BLP and IPG algorithms, respectively. We also note that the COP variant is very useful for the medium and large domains. For example, in all large domains, the exh. variant ran out of memory while the COP variant computed the optimal solutions for horizons up to 100. Yet, the exh. variant can compute the optimal solution of small problems faster than the COP variant. For instance, in the recycling robot for horizon \(T = 1000\), the exh. variant computed the optimal solution in about 5 times faster than the COP variant of the MPS algorithm due to overhead in the constraint optimization formulation and a lack of structure that can be utilized.

There are many different reasons for these results. The MPS algorithm outperforms GMAA\(^*\)-ICE and IPG mainly because they perform a policy search in the space of decentralized history-dependent policies. Instead, the MPS algorithm performs its policy search in the space of decentralized Markov policies, which is exponentially smaller than that of the decentralized history-dependent policies. The MPS outperforms the BLP algorithm mainly because of the dimension of its solution representation. More specifically, the number of bilinear terms in the BLP approach grows polynomially in the horizon of the problem, causing it to not perform well for large problems and large horizons with tightly coupled reward values.

We continue the evaluation of the MPS algorithm on randomly generated instances with multiple agents. The random instances were built upon the recycling robot problem described in Sutton and Barto [26]. Given \(n\) such models, each of which is associated with a single agent, we choose a number of interaction events. An interaction event is a pair of joint states and actions \((s,a)\) where the reward \(r(s,a)\)

---

\(^2\)All problem definitions are available at the following website: http://users.isr.ist.utl.pt/~mitspaan/deepomdp/

Table 1: Experimental results for the COP and exh. variants of MPS as well as GMAA\(^*\)-ICE (labeled ICE), IPG, and BLP.