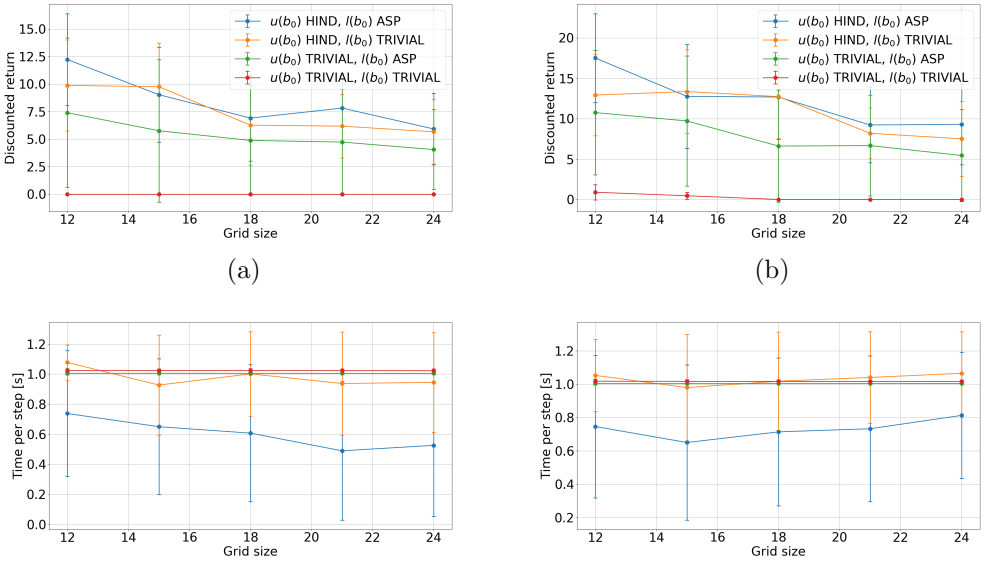

(a) (b)

(c) $M = 4$ rocks. (d) $M = 8$ rocks.

Figure 1: Discounted return (**top**) and computational time per step (**bottom**) with DESPOT **Julia** (mean $\pm$ standard deviation) for rocksample in larger grids and with more rocks (**EXP-2**).

Generalization tests in rocksample with Julia DESPOT (EXP-2)

In Figure 1, we report the results obtained with Julia implementation of DESPOT. They confirm the results achieved with the C++ implementation, proving the generality of our methodology *to different software implementations* of the solver. In particular, in the 20×20 grid with 20 rocks, *ASP + HIND* achieves slightly better performance than *TRIVIAL + HIND* in terms of average discounted return (17.68 ± 7.15 vs. 14.45 ± 5.75) and computational time per step (0.81 ± 0.47 s vs. 1.20 ± 0.27 s). With *TRIVIAL* upper bound, a strong improvement is observed in the discounted return (*ASP + TRIVIAL* achieves 13.19 ± 8.30, while *TRIVIAL + TRIVIAL* achieves 1.70 ± 2.64). It is interesting to note that DESPOT with *ASP + TRIVIAL* outperforms AdaOPS with the same bounds, in this scenario.

We do not report results with handcrafted (*preferred*) heuristics in the lower bound, since they are not implemented in Julia version of DESPOT.