Finding Options that Minimize Planning Time

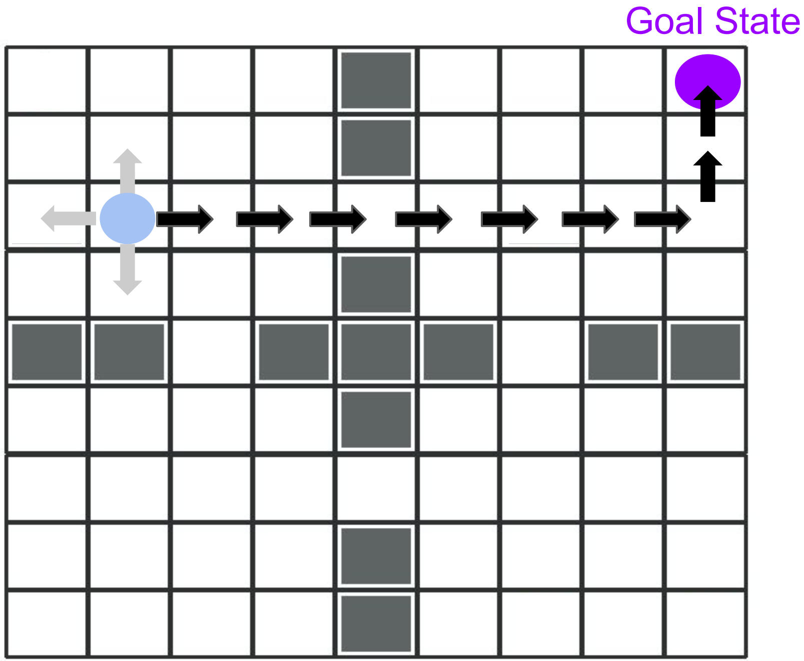
Yuu Jinnai[1], David Abel[1], D Ellis Hershkowitz[2], Michael L. Littman[1], George Konidaris[1]

Brown University[1], Carnegie Mellon University[2]

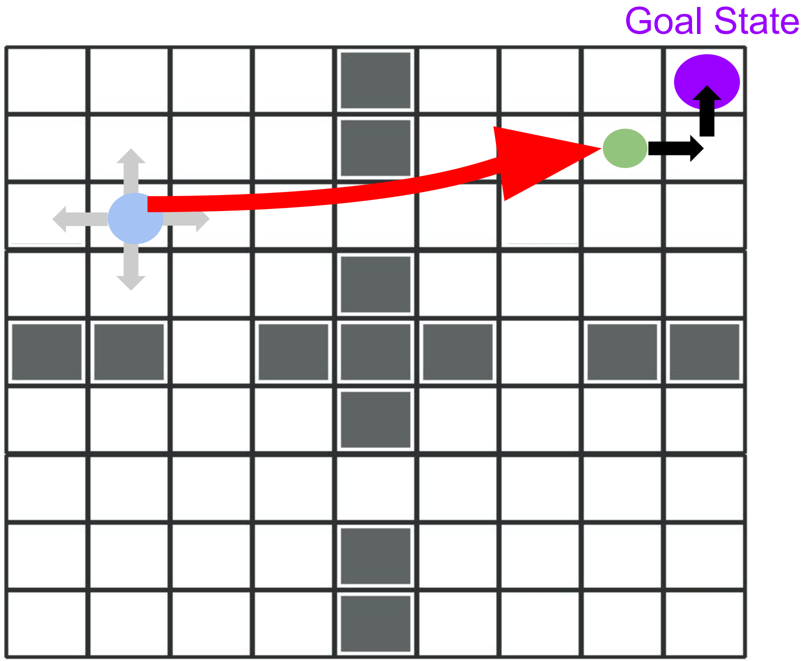
The problem of finding an optimal set of options that minimize planning time is NP-hard

Options (Sutton et al. 1999)

Primitive Actions

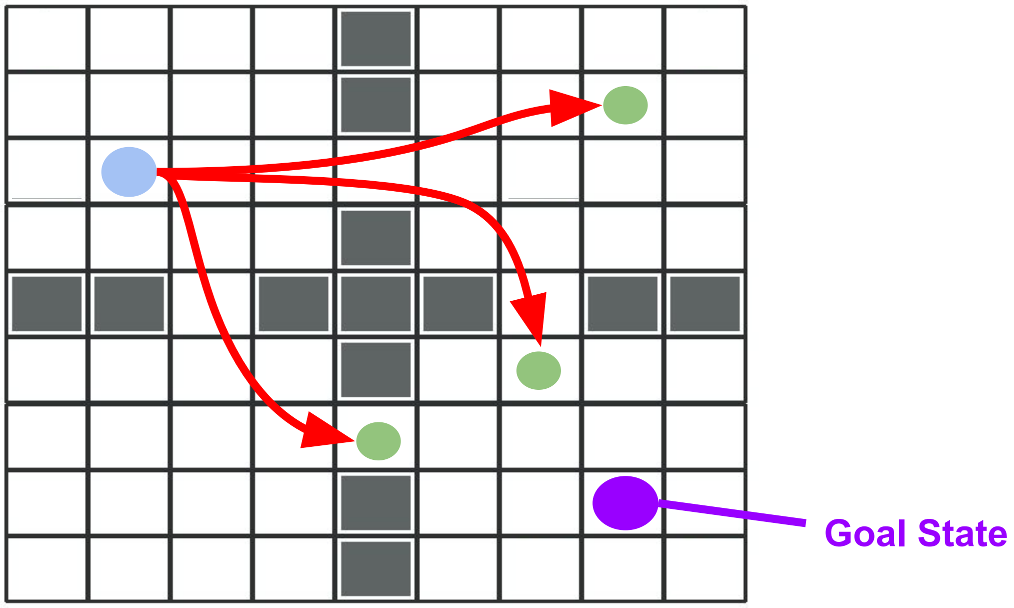


Using Options



Research Question: Which Options are the Best?

Using Options



: Initiation State: *I*(s)

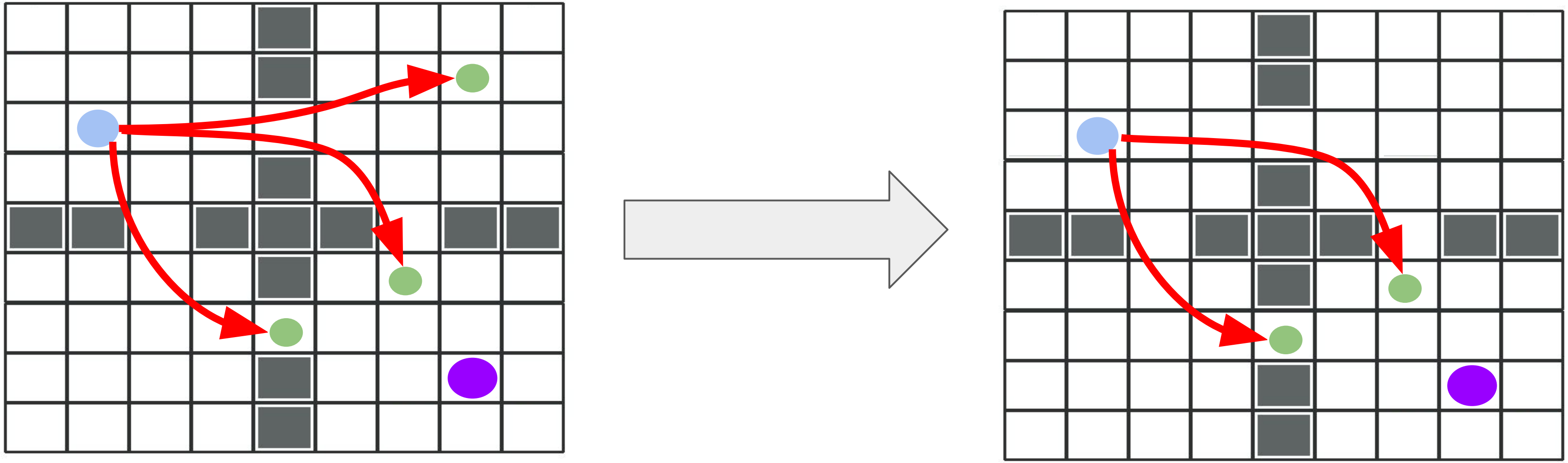
: Termination State: β(s)

Contributions

1. **Formally define** the problem of finding an optimal set of options for planning (value iteration algorithm)

Given: an **MDP,** a set of options, and an integer k

Return: an optimal set of options of size k

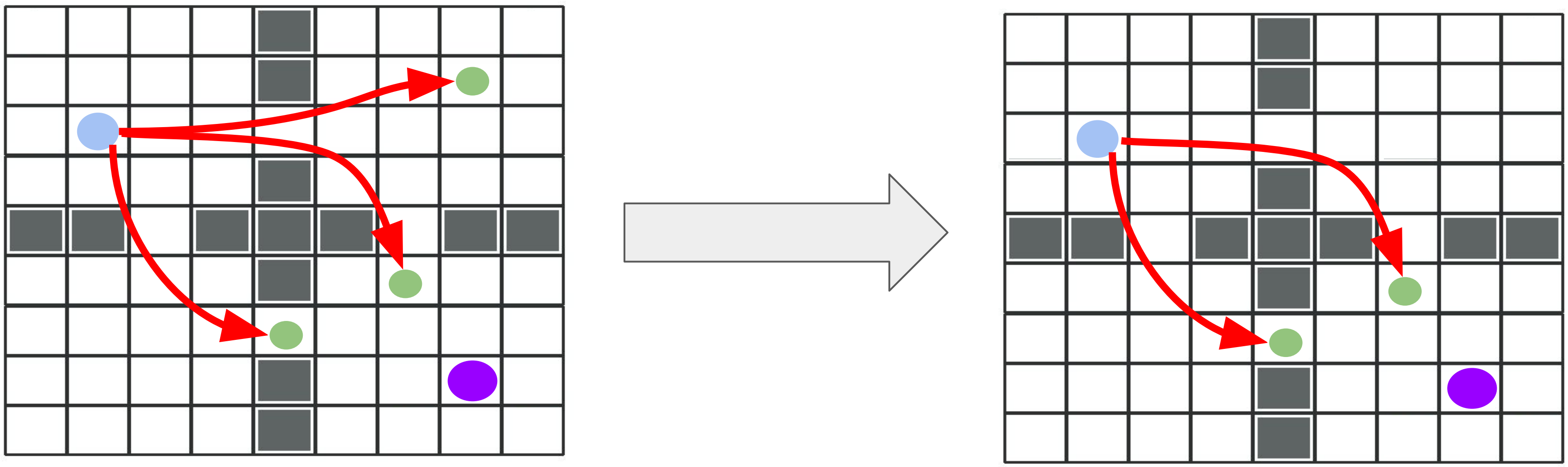


Contributions

1. **Formally define** the problem of finding an optimal set of options for planning
2. The complexity of computing an optimal set of options is **NP-hard**

Given: an **MDP,** a set of options, and an integer k

Return: an optimal set of options of size k



Contributions

1. **Formally define** the problem of finding an optimal set of options for planning
2. The complexity of computing an optimal set of options is **NP-hard**

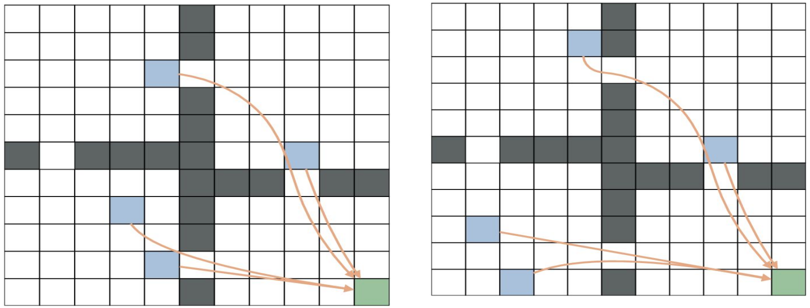
The problem:

1. is $2^{\log^{1-\epsilon} n}$-hard to approximate for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\text{poly} \log n})$, where n is the input size;
2. is $\Omega(\log n)$-hard to approximate even for deterministic MDPs unless $P = NP$;
3. has an $O(n)$-approximation algorithm;
4. has an $O(\log n)$-approximation algorithm for deterministic MDPs.

Contributions

1. **Formally define** the problem of finding an optimal set of options for planning
2. The complexity of computing an optimal set of options is **NP-hard**
3. **Approximation algorithm** for computing optimal options (under conditions)

Optimal Options | Approximation Algorithm



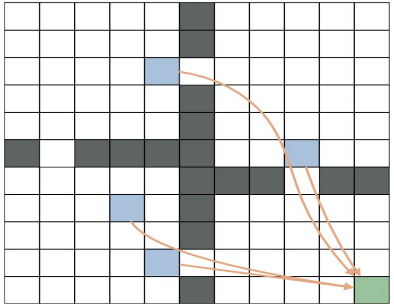
: Initiation State: *I*(s)

: Termination State: *β*(s)

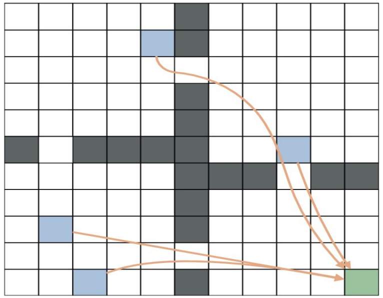
Contributions

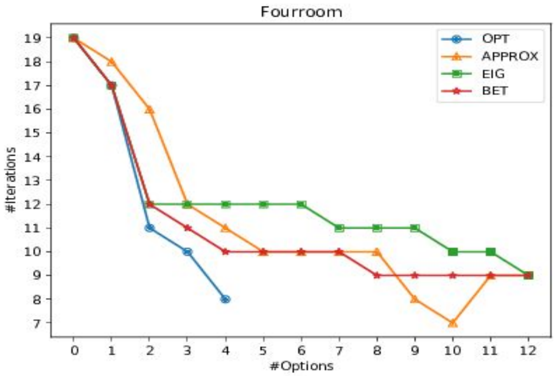
1. **Formally define** the problem of finding an optimal set of options for planning
2. The complexity of computing an optimal set of options is **NP-hard**
3. **Approximation algorithm** for computing optimal options (under conditions)
4. **Experimental evaluation** to compare with existing heuristic algorithms

Optimal Options



Approximation Algorithm





: Initiation State: *I*(s)

: Termination State: *β*(s)

Message

Finding options that minimize planning time is NP-hard

Option discovery is useful for planning
if and only if we have
structures, priors, or assumptions

Poster at Ballroom #40