Anytime-Constrained Reinforcement Learning

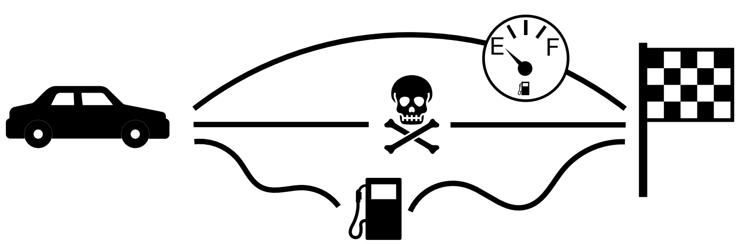
Jeremy McMahan and Xiaojin Zhu

Computer Sciences
SCHOOL OF COMPUTER, DATA & INFORMATION SCIENCES
UNIVERSITY OF WISCONSIN-MADISON

Optimal policies that obey a cost constraint at *every* point in time can be computed in polynomial time via approximate state augmentation!

Motivation

Self-driving cars must obey *safety* and *fuel* constraints.



- Cars must adaptively update their routes to refuel
- Safety must be guaranteed at all times
- These considerations cannot be captured by expectation or chance constraints

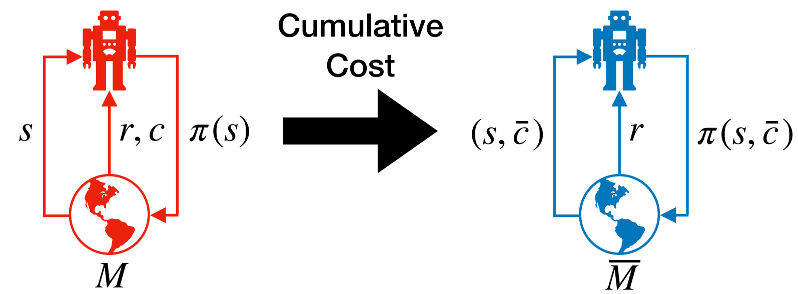
Anytime Constraints

Total cost NEVER exceeds the budget!

$$\text{(ANY)} \begin{cases} \max_{\pi} \mathbb{E}^{\pi}_{M}\left[\sum_{h=1}^{H} r_h(s_h, a_h)\right] \\ \text{s.t. } \mathbb{P}^{\pi}_{M}\left[\forall t \in [H],\ \sum_{h=1}^{t} c_h \leq B\right] = 1. \end{cases}$$

Theorem: Solving (ANY) is NP-hard. Determining Feasibility is even NP-hard with 2 constraints.

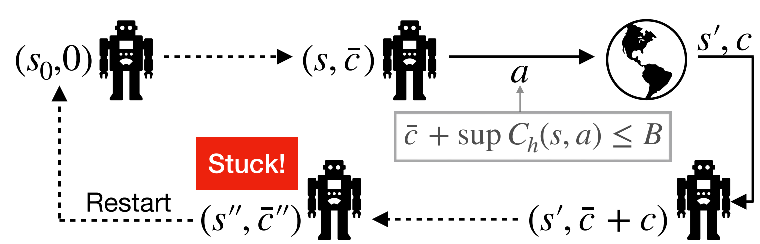
Reduction to RL



Theorem: Solving $\overline{M}$ bar solves (ANY), and can be done in polynomial time if the cost precision is small.

State Computation

All feasible (state, cost)-pairs are hard to compute, but we can compute a superset using *Safe Exploration:*



Add each pair to $\bar{S}$

Lemma: $\bar{S}$ contains all feasible $(s, \bar{c})$ pairs and can be computed in time $O(HSA2^{\text{cost precision}})$ using *forward induction.*

Approximation

Reduce complexity by projecting costs to a smaller space:

An *optimistic* projection guarantees optimality:

$$f_h(\hat{c}, c) = \begin{cases} \hat{c} + \left\lfloor \frac{c}{\ell} \right\rfloor \ell & \text{o.w.} \\ \left\lceil \frac{B-(H-h)c^{\max}}{\ell} \right\rceil \ell & \text{if SMALL} \end{cases}$$

Projecting leads to approximate feasibility:

$$\mathbb{P}^{\pi}_{M}\left[\forall t \in [H],\ \sum_{h=1}^{t} c_h \leq B(1+\epsilon)\right] = 1$$

Theorem: Using $\ell = \epsilon B/H$, solving $\hat{M}$ yields an optimal-value, approximately feasible policy in polynomial time so long as c^{max} is polynomial.

Conclusions

- Anytime constraints are more realistic for many applications, but are NP-hard to solve.
- Efficiently computing exact solutions is possible when the cost precision is low.
- Efficiently computing approximately-feasible solutions is possible when the cost distributions are upper bounded. *Best possible guarantees in theory!*