Probabilistic Planning with Non-Linear Utility Functions and Worst-Case Guarantees

Stefano Ermon, Carla Gomes, Bart Selman
Department of Computer Science

Alexander Vladimirsky
Department of Mathematics

Cornell University
Introduction

- Markov Decision Processes used for many probabilistic planning problems
- Non-linear utility functions used to describe preferences among all possible outcomes
- Risk sensitive planners want to maximize utility subject to worst case constraints
Contributions of this paper

- This paper contributes a dynamic programming solution to finding policy that maximizes reward subject to worst case constraints
- Stochastic shortest path problem on real world road networks is used to demonstrate practicality of the method
Markov Decision Processes

An MDP is a tuple \((S,A,P,r)\)

\(S\): set of states

\(A\): set of actions

\(P\): transition probabilities for next action given current state and action

\(r\): reward amount given state, action, and next state

This paper only considers finite MDPs
Policies

- Planning horizon $T$ is number of time steps that the agent plans for
- A *history* at time step $t$ is a sequence of states and actions that lead from initial state to state at time $t$
- Set of all histories at time step $t$ is denoted $H_t = (S \times A)^t \times S$
- Policy is a sequence of decision rules, one for each time step in planning horizon
- Most general decision rules are *randomized history-dependent* which are mappings $d_t: H_t \rightarrow P(A)$ where $P(A)$ is set of probability distributions over the actions
Utility Functions

Let $w_T$ be total reward received by agent

$$w_T = \sum_{t=0}^{T-1} r_t(s_t, a_t, s_{t+1})$$

Let $U$ be monotonically non-decreasing utility function which maps rewards to utility values
Finite Horizon Problems

Start with a finite planning horizon

We define the value of a policy \( \pi \in \Pi^{HR} \) from an initial state \( s \in S \) as

\[
v_{U,T}^\pi(s) = \mathbb{E}^{s,\pi} \left[ U \left( \sum_{t=0}^{T} r_t \right) \right] = \mathbb{E}^{s,\pi} [ U (w_T) ]
\]

which is the expected utility of the total reward \( w_T \).

Expected Utilities exist and are finite due to finite number of trajectories (Lui 2006)

\[
v_{U,T}^*(s) = \sup_{\pi \in \Pi} v_{U,T}^\pi(s)
\]
Worst-case constraints

\[ d_T^*(s) = \sup_{\pi \in \Pi} d_T^\pi(s) \]

where

\[ d_T^\pi(s) = \min \{ k | \mathbb{P}[w_T = k] > 0 \} \]

is the worst-case realization of the total reward \( w_T \)

We consider linear worst-case constraints of form \( w_T > L \) \( (6) \). Want to find:

\[ v_{U,T,L}^*(s) = \sup_{\pi \in \Pi(L)} v_{U,T}^\pi(s) \]

\( \Pi(L) \) is set of policies where reward satisfies \( (6) \)

\( \Pi(L) \) is not empty iff \( L < d_T^*(s) \)
Extended value utility functions and optimality conditions

- Extended value utility functions are used in algorithm
- Defined as monotonically non-decreasing functions $U: \mathbb{R} \rightarrow \mathbb{R} \cup \{-\infty\}$
- The optimal policy with these utility functions is deterministic and depends on the history.
- By augmenting the state space it turns out it depends only on the history through the total reward accumulated thus far
State space augmentation

For computation, we will augment MDP with sum of accumulated rewards

\[
W^0 = \{0\}, \quad W^{t+1} = \{r + w | r \in R, w \in W_t\}
\]

We consider an extended MDP where the augmented states space is \(S = (S \times W^0) \cup (S \times W^1) \cup \cdots \cup (S \times W^T)\). The actions available in an augmented state \(\langle s \rangle = (s, w)\) are \(A_s = A_{(s, w)} = A_s\) for all wealth levels \(w\). The transition probability from a state \(\langle s \rangle = (s, w)\) to \(\langle s' \rangle = (s', w')\) is

\[
P(\langle s' \rangle | \langle s \rangle, a) = \begin{cases} P(s' | s, a) & \text{if } w' = w + r(s, a, s') \\ 0 & \text{otherwise} \end{cases}
\]

Augmented Rewards are zeros

Terminal Augmented reward \(J(\langle s \rangle) = U(w)\) applicable at time \(T\) for augmented state \(\langle s \rangle\)
Policy Computation and Pruning

**Algorithm 1** Dynamic Programming equations for the augmented problem

\[ t \leftarrow T \]
\[
\text{for all } s \in S \text{ do} \]
\[
\text{Initialize } (z)_T^{*, T}(s, w) = U(w) \]
\[
\text{for } t = T - 1 \rightarrow 0 \text{ do} \]
\[
\text{for all } s \in S \text{ do} \]
\[
\text{for all } w \in W^t \text{ do} \]
\[
(z)_T^{*, t-1}(s, w) = \]
\[
\max_{a \in A_s} \sum_{s' \in S} P(s'|s, a) [(z)_T^{*, t}(s', w + r(s, a, s'))] \]

You can prune the search space by eliminating cases whenever we cannot meet the worst case requirement.
Stochastic shortest paths in road networks

- Given vertices and edges
- There are starting vertices and goal vertices.
- Want to reach goal vertex from initial vertex with least cost
- Cost of traversing edge is stochastic
- Real world road network is used as underlying graph
- Edge costs model travel times
  - Assumed to be discretized Beta-distributed random variables
  - Expected value is edge length
- Rewards are discretized negative costs, hence assumption of finite MDP holds
Utility Functions used

\[ U(x) = x \]

\[ U_L(x) = \begin{cases} 
-\infty & x \leq L \\
1 & x > L 
\end{cases} \]

\[ U_{K,D,L}(x) = \begin{cases} 
1 & K \leq x \\
(x - D)/(K - D) & D \leq x < K \\
0 & L < x < D \\
-\infty & x \leq L 
\end{cases} \]

Finally, we consider a worst-case constrained exponential utility function \( U_{\gamma,L}(x) \), by introducing a worst-case lower bound \( L \) in the standard utility function \( U_{\gamma}(x) = e^{\gamma x} \).
Road Network Data

San Joaquin County Road Network graph used

18263 nodes

23874 edges

Results on following slide show distribution of times to reach destination node from starting node
Red: Optimize worst case Markovian policy (depends only on current state)

Black: Minimize expected travel time $U(x)$ via our algorithm

Green: Optimize utility function via our algorithm

Resulting probability distributions and worst-case bounds (dashed lines)
Conclusions

- When planner cares about different objectives, augmented policies can achieve significant improvement over standard ones
- Non-linear utility functions allow a richer set of planning features yet increase the complexity due to augmentation of the state space
- This formulation can be useful for time-dependent problems where augmentation of state space is unavoidable
- Example is Green Driver App which faces SSPs whose edge cost probability distributions depend on current time since traffic lights are modeled
- Paper’s approach scales to real world road networks and leads to significantly different plans than traditional optimization criteria