Multi-Objective MDPs with Lexicographic Reward Preferences

Kyle Hollins Wray and **Shlomo Zilberstein**
University of Massachusetts
Amherst, MA 01003, USA
{wray, shlomo}@cs.umass.edu

Abdel-Illah Mouaddib
GREYC - Univeaite de Caen
Bd Marechal Juin, BP 5186
F14032 Caen Cedex, France
mouaddib@info.unicaen.fr

Introduction

This document describes work-in-progress for lexicographic value iteration in MOMDPs. Essentially, the idea is that for each state, lvmax starts with all actions available for the first value function. It then computes the optimal actions to take, but also keeps actions which yield values that are close to optimal (within some tolerance δ_1). Then, it uses this restricted action set to compute the optimal value for the next value function (which has a tolerance δ_2). This continues until the end, when a collection of actions remains after this pruning. The final action is selected from this set by going through the value functions one more time, but this time selecting only the optimal action (or set of actions if a tie occurs) until one remains. If after all this there is *still* a tie, we will just assume a simple preference ordering over actions breaks the final tie.

Note that if all $\delta_i = 0$, it reduces to a pure argmax selection at each pruning step. This is our original lexicographic preference idea.

Problem Definition

We present a formal model for multi-objective MDPs (MOMDPs) with a lexicographic ordering over the rewards' value functions. An MDP is a stochastic control process in which an agent exists in a set of states. The actions the agent performs in particular state each have a distribution over potential successor states. This transition results in a reward. The process operates over the state space for a finite or (virtually) infinite number of discrete time steps. The goal is to maximize an expected reward over the collection stages. Definition 1 formally states this process.

Definition 1. A **Multi-Objective Markov Decision Process (MOMDP) with lexicographic reward preferences** is a represented by a 4-tuple $\langle S, A, T, \mathbf{R} \rangle$ with:

- S is a finite set of n states, with initial state $s_0 \in S$
- A is a finite set of m actions
- $T : S \times A \times S \to [0, 1]$ is a state transition function which specifies the probability of transitioning from a state $s \in S$ to state $s' \in S$, given action $a \in A$ was performed; equivalently, we may write $T(s, a, s') = Pr(s'|s, a)$
- $\mathbf{R} = [R_1, \ldots, R_k]^T$ is a vector of reward functions $R_i : S \times A \times S \to \mathbb{R}$, $\forall i \in K$, with $K = \{1, \ldots, k\}$; each specifies the reward for being in a state $s \in S$, performing action $a \in A$, and transitioning to a state $s' \in S$, often written as $\mathbf{R}(s, a, s') = [R_1(s, a, s'), \ldots, R_k(s, a, s')]^T$

We call $h \in \mathbb{N} \cup \{\infty\}$ the *horizon*, i.e., number of steps until the process terminates. The process may be defined as finite ($h < \infty$) or infinite ($h = \infty$). Over the iterations, rewards are discounted by *discount factor* γ. For finite horizon problems, typically $\gamma = 1$; for infinite horizon problems, $\gamma \in [0, 1)$.

A *policy* $\pi : S \to A$ is a mapping from each state $s \in S$ to an action $a \in A$. In finite horizon problems, we have a sequence of policies $\langle \pi_1, \ldots, \pi_h \rangle$ representing a policy for each stage.

Let $\mathbf{V} = [V_1, \ldots, V_k]^T$ be a set of *value functions*. Let each function $V_i^\pi : S \to \mathbb{R}$, $\forall i \in K$, represent the value of states S following policy π. The stochastic process of MDPs enable us to represent this using the expected value over the reward for following the policy at each stage.

$$\mathbf{V}^\pi(s) = \mathbb{E}\Big[\sum_{t=0}^{h-1} \gamma^t \mathbf{R}(s^t, \pi(s^t), s^{t+1}) \Big| s^0 = s, \pi \Big]$$

This allows us to recursively write the value of the state $s \in S$, given a particular policy π, in the following manner.

$$\mathbf{V}^\pi(s) = \sum_{s' \in S} T(s, \pi(s), s')(\mathbf{R}(s, \pi(s), s') + \gamma \mathbf{V}^\pi(s'))$$

Lexicographic Value Iteration

We lexicographically prefer V_i^π to V_{i+1}^π for each $i \in K$. We are able to rewrite value iteration to solve multi-objective MDPs in this manner. In particular, the $\max$ operator needs to be replaced with a *lexicographic vector maximization* operator, denoted lvmax, which is defined in its general form in Definition 2 below.

Definition 2. The **lexicographic vector maximization** operator lvmax is defined in Equation 2 below. Let X be a set and $f : X \to \mathbb{R}^k$ be a vector function. Let $\delta \in \mathbb{R}_+^k$ be a tuple of non-negative slack variables. Let $X_1 = X$ and increasingly smaller subsets of X as

$$X_{i+1} = \{x \in X_i | \max_{x' \in X_i} f_i(x') - f_i(x) \leqslant \delta_i\} \qquad (1)$$

for all $i \in \{1, \ldots, k-1\}$. To break ties, we define the final subsets as $X_1^* = X_k$ and $X_{i+1}^* = \text{argmax}_{x \in X_i^*} f_i(x)$.

$$\underset{x \in X}{\text{lvmax}}\, \mathbf{f}(x) = \begin{bmatrix} \max_{x \in X_1^*} f_1(x) \\ \vdots \\ \max_{x \in X_k^*} f_k(x) \end{bmatrix} \tag{2}$$

With this definition, the *Bellman update equation* for MOMDPs with lexicographic reward preferences may be written as Equation 3 below for all states $s \in S$.

$$\mathbf{V}(s) = \underset{a \in A}{\text{lvmax}}\, \mathbf{Q}(s, a) \tag{3}$$

$$\mathbf{Q}(s, a) = \sum_{s' \in S} T(s, a, s')(\mathbf{R}(s, a, s') + \gamma \mathbf{V}(s')) \tag{4}$$

This operator will only converge to within δ of the actual value of the states.

Theoretical Analysis

We provide a proof of convergence for lvmax value iteration in Proposition 1.

Proposition 1. Lexicographic vector max (lvmax) value iteration converges to a unique fixed point of value functions, given discount factors $\gamma_i \in [0, 1)$ for all $i \in K$ and *Lipschitz constant* $\nu \in [0, 1)$ defined as

$$\nu = \max_{i \in K} \left(\gamma_i + (1 - \gamma_i) \frac{\delta_i}{|R_i^+ - R_i^-|} \right) \tag{5}$$

with $R_i^+ = \max_{s \in S} \max_{a \in A} \max_{s' \in S} R_i(s, a, s')$ and $R_i^- = \min_{s \in S} \min_{a \in A} \min_{s' \in S} R_i(s, a, s')$.

Proof. To prove convergence, we must show that the Bellman optimality equation converges to a unique fixed point. We will do this by induction on $i \in K$. First, we must state some definitions.

For a metric space $\langle X, d \rangle$, where X is a set and d is a distance metric, a map $f : X \to X$ is called a *contraction map* if there exists an α such that $d(f(x), f(y)) = \alpha d(x, y)$, for all $x, y \in X$.

Let the space X_i be the *space of value functions for* $i \in K$, i.e., we have $V_i = [V_i(s_1), \ldots, V_i(s_n)]^T \in X_i$. Let the distance metric d_i be the *max norm*, i.e., $\|V_i\|_\infty = \max_{s \in S} |V_i(s)|$. Since $\gamma_i \in [0, 1)$, the metric space $M_i = \langle X_i, d_i \rangle$ is a *complete metric space*; every Cauchy sequence of M_i converges to a point in M_i.

Let the lvmax Bellman optimality equation's (Equation 3) be defined as an operator B, i.e., $V^{t+1} = BV^t$, for $V^t, V^{t+1} \in X$ with $t \geqslant 0$. Let element $i \in K$ of the operator be B_i, such that $V_i^{t+1} = (BV^t)_i = B_i V_i^t$. We prove the operator B_i is a contraction map in M_i for all $i \in K$, given either that $i = 1$ or that the previous $i - 1$ has converged to within ϵ of its fixed point.

Let $V_{1i}, V_{2i} \in X_i$ be any two value function vectors, and $\gamma_i \in [0, 1)$. We first apply the definition of lvmax from Equation 2.

$$\|B_i V_{1i} - B_i V_{2i}\|_\infty = \max_{s \in S} |\max_{a \in A_{1i}^*} Q_{1i}(s, a) - \max_{a \in A_{2i}^*} Q_{2i}(s, a)|$$

By Definition 2, $A_{1i}^* \subseteq A_{1i}$. Therefore, for all $s \in S$, $\max_{a \in A_{1i}^*} Q_{1i}(s, a) \leqslant \max_{a' \in A_{1i}} Q_{1i}(s, a)$. Also by Definition 2, for all $a^* \in A_{2i}^* \subseteq A_{2,i+1} = \{a \in A_{2i} | \max_{a' \in A_{2i}} Q_{2i}(s, a') - Q_{2i}(s, a) \leqslant \delta_i\}$. Thus,

$$\max_{a' \in A_{2i}} Q_{2i}(s, a') - \delta_i \leqslant Q_{2i}(s, a^*) \leqslant \max_{a' \in A_{2i}^*} Q_{2i}(s, a')$$
$$- \max_{a' \in A_{2i}^*} Q_{2i}(s, a') \leqslant \delta_i - \max_{a' \in A_{2i}} Q_{2i}(s, a')$$

Without loss of generality, assume that $B_i V_{1i} \geqslant B_i V_{2i}$, making the absolute value optional in the first equation below. If the opposite is true, then we may simply switch the logic for deriving the two above upper bounds and apply those instead throughout.

Combine these three facts, and we obtain the following.

$$\begin{aligned} &\|B_i V_{1i} - B_i V_{2i}\|_\infty \\ &\leqslant \max_{s \in S} |\max_{a \in A_{1i}} Q_{1i}(s, a) - \max_{a \in A_{2i}} Q_{2i}(s, a) + \delta_i| \\ &\leqslant \max_{s \in S} |\max_{a \in A_{1i}} Q_{1i}(s, a) - \max_{a \in A_{2i}} Q_{2i}(s, a)| + |\delta_i| \end{aligned}$$

By Definition 2, when $i = 1$ we have $A_{1i} = A_{2i} = A$. Similarly, when $i \in \{2, \ldots, k\}$ given that $i-1$ has converged to within ϵ of its fixed point, it yields a unique fixed set of actions A', with $A_{1i} = A_{2i} = A' \subseteq A$. Let us denote this fixed actions set as $\bar{A}_i$ for all $i \in K$. Also, as part of the $Q(\cdot)$ values, we distribute $T(\cdot)$ to each $R(\cdot)$ and $V(\cdot)$ in the summations, then apply the property: $\max_x f(x) + g(x) \leqslant \max_x f(x) + \max_x g(x)$, twice.

$$\begin{aligned} &\|B_i V_{1i} - B_i V_{2i}\|_\infty \\ &\leqslant \max_{s \in S} \Big| \max_{a \in \bar{A}_i} \Big(\sum_{s' \in S} T(s, a, s') R_i(s, a, s') \\ &\qquad + \gamma_i \sum_{s' \in S} T(s, a, s') V_{1i}(s') \Big) \\ &\qquad - \max_{a \in \bar{A}_i} \Big(\sum_{s' \in S} T(s, a, s') R_i(s, a, s') \\ &\qquad - \gamma_i \sum_{s' \in S} T(s, a, s') V_{2i}(s') \Big) \Big| + \delta_i \\ &\leqslant \max_{s \in S} \Big| \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') R_i(s, a, s') \\ &\qquad + \gamma_i \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') V_{1i}(s') \\ &\qquad - \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') R_i(s, a, s') \\ &\qquad - \gamma_i \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') V_{2i}(s') \Big| + \delta_i \\ &\leqslant \max_{s \in S} \Big| \gamma_i \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') V_{1i}(s') \\ &\qquad - \gamma_i \max_{a \in \bar{A}_i} \sum_{s' \in S} T(s, a, s') V_{2i}(s') \Big| + \delta_i \end{aligned}$$

Note that we can pull out $\gamma_i \in [0, 1)$. Recall, that for any two functions f and g, $|\max_x f(x) - \max_x g(x)| \leqslant$

$\max_x |f(x) - g(x)|$.

$$\|B_i V_{1i} - B_i V_{2i}\|_\infty$$
$$\leqslant \gamma_i \max_{s\in S} \max_{a\in \bar{A}_i} \Big| \sum_{s'\in S} T(s,a,s')(V_{1i}(s') - V_{2i}(s')) \Big| + \delta_i$$

Since $\sum_{s'\in S} T(s,a,s') = 1$ and for all $s' \in S$, $T(s,a,s') \in [0,1]$, it is defined on the n-simplex. It then scales the vertices by the values $R(\cdot)$ or $V(\cdot)$. This forms simple convex polytope. Convex polytopes obtain their maximum value at the vertices (or on an entire edge or face, which includes the vertices). Therefore, we may exclusively maximize over these vertices ($R(\cdot)$ and $V(\cdot)$), and may simply drop both maximizations which select the weights (i.e., maximization over $s \in S$ and $a \in A$).

$$\|B_i V_{1i} - B_i V_{2i}\|_\infty \leqslant \gamma_i \max_{s'\in S} \Big| V_{1i}(s') - V_{2i}(s') \Big| + \delta_i$$
$$\leqslant \gamma_i \|V_{1i} - V_{2i}\|_\infty + \delta_i \frac{\|V_{1i} - V_{2i}\|_\infty}{\|V_{1i} - V_{2i}\|_\infty}$$
$$\leqslant \Big(\gamma_i + \frac{\delta_i}{\|V_{1i} - V_{2i}\|_\infty}\Big) \|V_{1i} - V_{2i}\|_\infty$$

We can place a constant upper bound on this value to remove the denominator $\|V_{1i} - V_{2i}\|_\infty$. Consider any finite or finite sequence of states and actions performed by the agent: $z = \langle s^0, a^0, s^1, a^1, \ldots \rangle$. The utility $u_i^h(z)$ at horizon $h \in \mathbb{N} \cup \{\infty\}$ is bounded:

$$u_i^h(\langle s^0, a^0, s^1, a^1, \ldots \rangle) = \sum_{t=0}^{h} \gamma_i^t R_i(s^t, a^t, s^{t+1})$$
$$\leqslant \sum_{t=0}^{\infty} \gamma_i^t R_i^+ \leqslant \frac{R_i^+}{1-\gamma_i}$$

Similarly,

$$-u_i^h(\langle s^0, a^0, s^1, a^1, \ldots \rangle) \geqslant \frac{R_i^-}{1-\gamma_i}$$

The value function is therefore bounded by these values, because including state transitions would only decrease the values.

$$\|V_{1i} - V_{2i}\|_\infty \leqslant \Big| \frac{R_i^+}{1-\gamma_i} - \frac{R_i^-}{1-\gamma_i} \Big| \leqslant \frac{|R_i^+ - R_i^-|}{1-\gamma_i}$$
$$\Rightarrow \qquad \frac{-1}{\|V_{1i} - V_{2i}\|_\infty} \leqslant -\frac{1-\gamma_i}{|R_i^+ - R_i^-|}$$

Now we may obtain final result, proving that the Bellman operator B_i is a contraction map on metric space M_i, for all $i \in K$.

$$\|B_i V_{1i} - B_i V_{2i}\|_\infty \leqslant \Big(\gamma_i - \delta_i \frac{-1}{\|V_{1i} - V_{2i}\|_\infty}\Big) \|V_{1i} - V_{2i}\|_\infty$$
$$\leqslant \Big(\gamma_i - \delta_i \Big(- \frac{1-\gamma_i}{|R_i^+ - R_i^-|}\Big)\Big) \|V_{1i} - V_{2i}\|_\infty$$
$$\leqslant \Big(\gamma_i + (1-\gamma_i) \frac{\delta_i}{|R_i^+ - R_i^-|}\Big) \|V_{1i} - V_{2i}\|_\infty$$
$$\leqslant \nu \|V_{1i} - V_{2i}\|_\infty$$

Thus, for all $i \in K$, by definition of a contraction map, B_i admits at most one fixed point. Additionally, since M_i is a complete metric space, we can guarantee convergence to a unique fixed point. *Banach's fixed point theorem* states that if $M_i = \langle X_i, d_i \rangle$ is a complete metric space and $B_i : X_i \to X_i$ is a contraction map, then B_i admits a unique fixed point $V_i^* \in X_i$, i.e., $B_i V_i^* = V_i^*$. Thus, we will have convergence of value iteration to a unique fixed point $V_i^* \in X_i$, for all $i \in K$. □

We may also derive an equation which guarantees convergence to within $\epsilon > 0$ of the fixed point, as shown in Proposition 2.

Proposition 2. Lexicographic vector max (lvmax) value iteration converges to within $\epsilon > 0$ of its unique fixed point once $\|V^{t+1} - V^t\|_\infty < \epsilon \frac{1-\nu}{\nu}$.

Proof. From Proposition 1, for all $i \in K$, a corollary of Banach's fixed point theorem is that the speed of convergence to within $\epsilon > 0$ of the fixed point x^* is known (using the generic notation from above for a metric space).

$$d(x^*, x_{t+1}) \leqslant \frac{\alpha}{1-\alpha} d(x_{t+1}, x_t)$$
$$\|V_i^* - V_i^{t+1}\|_\infty \leqslant \frac{\nu}{1-\nu} \|V_i^{t+1} - V_i^t\|_\infty$$

Since we want the distance from the fixed point $V_i^* \in X_i$ to be ϵ, we may rewrite the equation accordingly.

$$\epsilon \leqslant \frac{\gamma_i}{1-\gamma_i} \|V_i^{t+1} - V_i^t\|_\infty$$
$$\epsilon \frac{1-\gamma_i}{\gamma_i} \leqslant \|V_i^{t+1} - V_i^t\|_\infty$$

The above equation states that we are at least ϵ (or more) away from the fixed point when the maximum difference (over the states) between iterations satisfies the inequality. Therefore, we flip the inequality to create a convergence criterion, which ensures that we are ϵ or less from the fixed point. Finally, this must be true for all $i \in K$, so we may rewrite it with the infinity norm defined over both K and S.

$$\|V^{t+1} - V^t\|_\infty < \epsilon \frac{1-\nu}{\nu}$$

□

It is straightforward to show that lvmax value iteration generalizes value iteration, as shown in Proposition 3.

Proposition 3. Lexicographic vector max (lvmax) value iteration generalizes value iteration.

Proof. Let $M = \langle S, A, T, \mathbf{R} \rangle$ with $\mathbf{R} = \langle R_1 \rangle$ and $\delta_1 = 0$. By Definition 2, for all $s \in S$, $\mathbf{V}(s) = \text{lvmax}_{a\in A} \mathbf{Q}(s,a) = \max_{a\in A_1^*} Q_1(s,a)$. Since $A_1^* = A_k = A_1 = A$, we have $V_1(s) = \max_{a\in A} Q_1(s,a)$. This is value iteration on $M' = \langle S, A, T, R_1 \rangle$. □

Typically, a multi-objective optimization problem converts the problem into a objective function by summing the value function and weighting them in a particular manner. Our lvmax value iteration returns a different solution from the linearly weighted function approaches used to solve MOMDPs. Proposition 4 proves this fact.

Proposition 4. Let the normal Bellman's equation, with linear weights $\mathbf{w} \in \triangle^k$, and lvmax's value iteration's resulting value functions be denoted as the n-by-k matrices V^* and V^*_{lv}, respectively. There exist a class of MOMDPs $\mathcal{M}$ such that $V^* \neq V^*_{lv}$.

Proof. Assume by contradiction that for all $M \in \mathcal{M}$, $V^* = V^*_{lv}$, i.e., there always exists a set of weights $\mathbf{w} \in \triangle^k$ which makes this so. We know three things for all $s \in S$: $V^*_{\mathbf{w}}(s) = BV^*_{\mathbf{w}}(s)$, $V^*_{\mathbf{w}}(s) = \mathbf{w}\mathbf{V}^*(s)$, and $\mathbf{V}^*_{lv}(s) = B_{lv}\mathbf{V}^*_{lv}(s)$. Therefore, we have:

$$f(\mathbf{V}^*_{lv}(s), \mathbf{w}) = f(\mathbf{V}^*(s), \mathbf{w}) = V^*_{\mathbf{w}}(s) = BV^*_{\mathbf{w}}(s)$$
$$= B(f(\mathbf{V}^*(s), \mathbf{w})) = B(f(B_{lv}\mathbf{V}^*_{lv}(s), \mathbf{w}))$$

We now apply the Bellman optimality operator to the right side, as well as the lvmax operator.

$$f(\mathbf{V}^*_{lv}(s), \mathbf{w}) = \max_{a \in A} \Big(\sum_{s' \in S} T(s,a,s') f(\mathbf{R}(s,a,s'), \mathbf{w}) + \gamma \sum_{s' \in S} T(s,a,s') f(\max_{a' \in A^*_i} \mathbf{Q}^*_{lv}(s',a'), \mathbf{w}) \Big)$$

Apply the properties the linearity of f.

$$0 = \max_{a \in A} f\Big(\sum_{s' \in S} T(s,a,s')\mathbf{R}(s,a,s') + \gamma \sum_{s' \in S} T(s,a,s') \max_{a' \in A^*_i} \mathbf{Q}^*_{lv}(s',a') - \mathbf{V}^*_{lv}(s), \mathbf{w} \Big)$$

$$0 = \max_{a \in A} f\Big(\mathbf{x}_a, \mathbf{w}\Big)$$

Since $\mathbf{w} \in \triangle^k$, and f is a linear sum of weights and components, we will show that $\exists M \in \mathcal{M}$ such that $\exists s \in S$ such that $\forall i \in K, \mathbf{x}_a > 0$. □

Proposition 5. Let $M = \langle S, A, T, \mathbf{R} \rangle$, with $\mathbf{R} = \langle R_1, R_2 \rangle$, be a MOMDP with dead ends for R_1 and goal states for R_2. Assume there exists a proper policy. With $\gamma = 1$, lvmax value iteration converges to a policy which strictly avoids dead ends, and otherwise strictly prefers goal states.

Experimentation

This section is just to provide some initial experiments.

Grid World with Dead Ends

First, we implemented a model similar to the classic grid world problem. The agent may move north, south, east, and west in a w-by-h grid world. The agent moves successfully with a 0.8 probability, and fails by moving right or left, each with a 0.1 probability. At the edges, if the agent cannot move, it remains still.

Throughout the area, dead ends are placed (denoted by "-") as well as goal states (denoted by "+"). In the normal MDP version, dead ends have a reward of $-\infty$, and only have a positive weight of 1.0 on the transition probability for remaining in the dead end. Goal states have a reward of 1.0, and all other states have a small penalty of -0.03.

In our MOMDP with lexicographic reward function model, we separate the dead ends and goal states into two reward functions. The first reward R_1 provides a 0.0 reward for all states, and a -1.0 reward for transitioning to a dead end. The second reward R_2 yields a 1.0 for transitioning to a goal state, and a -0.03 for all other states.

We implemented this initial test in Python 2.7. Figures 1, 2, and 3 show some example output from the model in this domain. Interestingly, it appears to work very well at strictly avoiding dead ends, and then optimizing around the remaining action possibilities.

```
- e s w w w w - s -
s s w w w w w w s s
+ w w w w w w - e s
n w w w n n w s n +
n n + w - e w w - e
n n n w s w w s s s
n n n w n n n w w w
n n n w - n - e n n
w - n e s s s n n n
w w - e e e e n n n
```

Figure 1: First example's policy.

```
w - e e e e s w - + w - n - e e n n n n
s s e e e e e s w - e + w - e n n n w - n
e e e e e e e + s n w s s w - n - n w -
```

Figure 2: Second example's policy.

There are additional scenarios for which lvmax is useful. If there are primary and secondary goals, they can be represented by different rewards, with primary goal states strictly favored over secondary goal states.

- If there exists a perfect policy, then it seems to always converge with $\gamma = 1$. The convergent policy always strictly avoids the dead ends, as desired.
- Without a perfect policy, it converges but the policy can often be off if a lot of dead ends are close to one another. This is probably one example where the slack variables $\delta_1, \ldots, \delta_k$ could be useful.
- These results do not utilize the slack variables: $\delta_1, \ldots, \delta_k$.

Multi-Objective Autonomous Driving

To demonstrate the usefulness of both the lexicographic ordering as well as the slack variables, we are considering the autonomous driving domain. We will consider a path planning agent acting over an OpenStreetMap graph, which

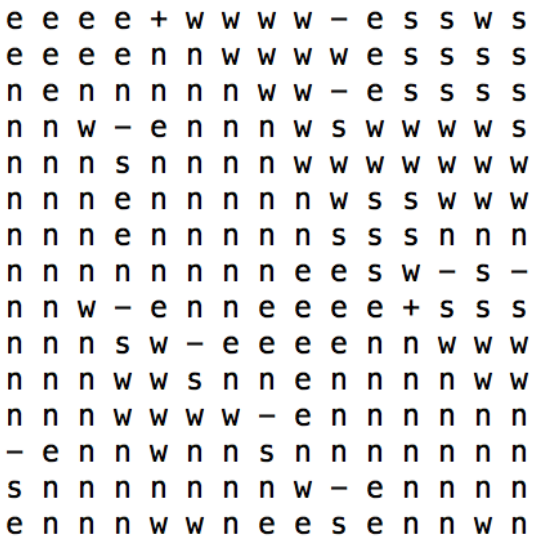

Figure 3: Third example's policy.

strictly prefers: shorter time/distance $>$ saving fuel/money $>$ stopping at optional amenities.

Our agent has some slack in optimizing the distance traveled. This allows for it to take an extra 5 minutes to go a bit out of the way in order to save some fuel, or stop at an optional amenity. This same slack holds for the other value functions as well.

We also can model dead ends in this domain. Consider an agent that can choose to drive faster. This might result in a ticket if they drive too fast an get caught, which could be considered a dead end.

This is currently being developed in C++, with supporting scripts to read the OpenStreetMap (OSM) file format written in Python.

Related Work

Recent work by Mouaddib used a strict lexicographic preference ordering for MOMDPs (Mouaddib 2004). Their work formed the foundation on which we built our lvmax value iteration, which generalizes their work through the introduction of slack variables. Other work by Perny et al. used Lorenz dominance with a weighted value functions favor more "fair" values (Perny et al. 2013).

In the past, others have used lexicographic ordering over value functions, calling this technique *ordinal dynamic programming* (Mitten 1974; Sobel 1975). Within the context of an MDP, Mitten assumed a specific preference ordering over outcomes in the finite horizon case. Sobel extended this model to infinite horizon MDPs, but did not fully capture our lvmax formulation under value iteration, slack variables generalizes their approach, and we present experiments showing its efficacy in an applied setting. Ordinal dynamic programming has also been explored within the reinforcement learning community (Gábor, Kalmár, and Szepesvári 1998; Natarajan and Tadepalli 2005), even with a similar notion of slack variables, but has not been represented in the general form we present.

Barbara and Jackson characterized an operator called leximin within an economics context, which is closely related to lvmax, except that the ordering is slightly different and does not include slack variables (Barbara and Jackson 1988). Since its inception, it has enjoyed use outside the domain of MDPs by other economics researchers (Bossert, Pattanaik, and Xu 1994; Fargier and Sabbadin 2005; Arlegi et al. 2005).

Our algorithm can capture the strict avoidance of dead ends as well as loops. Interestingly, Kolobov, Mausam, and Weld showed that unavoidable dead ends can be represented by a "price" (Kolobov, Mausam, and Weld 2012). The resulting formulation resembles a specific, non-slack variable version of lvmax value iteration.

A solid survey of the approaches used to solve MOMDPs is provided by Roijers et al. (2013), but other models exist. Constrained MDPs (CMDPs) are another formulation of MOMDPs, but to our knowledge no one has explored lexicographic preferences over rewards in this domain (Altman 1999). Additionally, Gonzales, Perny, and Dubus used Generalized Additive Decomposable (GAI) networks to capture the decomposability of the utility functions to model preferences (Gonzales, Perny, and Dubus 2011).

References

Altman, E. 1999. *Constrained Markov Decision Processes*, volume 7. CRC Press.

Arlegi, R.; Besada, M.; Nieto, J.; and Vzquez, C. 2005. Freedom of choice: the leximax criterion in the infinite case. *Mathematical Social Sciences* 49(1):1 – 15.

Barbara, S., and Jackson, M. 1988. Maximin, leximin, and the protective criterion: Characterizations and comparisons. *Journal of Economic Theory* 46(1):34 – 44.

Bossert, W.; Pattanaik, P. K.; and Xu, Y. 1994. Ranking opportunity sets: An axiomatic approach. *Journal of Economic Theory* 63(2):326 – 345.

Fargier, H., and Sabbadin, R. 2005. Qualitative decision under uncertainty: back to expected utility. *Artificial Intelligence* 164(12):245 – 280.

Gábor, Z.; Kalmár, Z.; and Szepesvári, C. 1998. Multi-criteria reinforcement learning. In *ICML*, volume 98, 197–205.

Gonzales, C.; Perny, P.; and Dubus, J. 2011. Decision making with multiple objectives using {GAI} networks. *Artificial Intelligence* 175(78):1153 – 1179. Representing, Processing, and Learning Preferences: Theoretical and Practical Challenges.

Kolobov, A.; Mausam; and Weld, D. S. 2012. A theory of goal-oriented mdps with dead ends. *CoRR* abs/1210.4875.

Mitten, L. G. 1974. Preference order dynamic programming. *Management Science* 21(1):43–46.

Mouaddib, A.-I. 2004. Multi-objective decision-theoretic path planning. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, 2814–2819 Vol.3.

Natarajan, S., and Tadepalli, P. 2005. Dynamic preferences in multi-criteria reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, 601–608. ACM.

Perny, P.; Weng, P.; Goldsmith, J.; and Hanna, J. 2013. Approximation of lorenz-optimal solutions in multiobjective markov decision processes. *CoRR* abs/1309.6856.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48:67–113.

Sobel, M. J. 1975. Ordinal dynamic programming. *Management Science* 21(9):967–975.