Multi-Objective MDPs with Lexicographic Reward Preferences

AAAI Submission #

Abstract

Sequential decision problems that involve multiple objectives are prevalent. Consider for example a driver of a semi-autonomous car who may want to minimize both travel time and the effort associated with manual driving. We introduce a rich model called lexicographic MDP (LMDP) and a corresponding planning algorithm called LVI that generalize previous work by allowing for conditional lexicographic preferences with slack. We analyze the convergence characteristics of LVI and establish its game theoretic properties. The performance of LVI in practice is tested within a realistic benchmark problem in the domain of semi-autonomous driving. Finally, we demonstrate how GPU-based optimization can improve the scalability of LVI and other value iteration algorithms for MDPs.

1 Introduction

Stochastic planning problems designed to optimize multiple objectives are widespread within numerous domains such as smart homes and commercial buildings (Kwak et al. 2012), reservoir water control (Castelletti, Pianosi, and Soncini-Sessa 2008), and autonomous robotics (Mouaddib 2004; Calisi et al. 2007). Current approaches often use a scalarization function and a weight vector to project the multi-objective problem to a single-objective problem (Roijers et al. 2013; Natarajan and Tadepalli 2005; Perny and Weng 2010; Perny et al. 2013). While these approaches leverage effectively the vast existing work on single-objective optimization, they have several drawbacks. First, choosing a projection is often too onerous to use in practice since there are many viable Pareto optimal solutions to the original multi-objective problem, making it hard to visualize and analyze alternative solutions. Often there is no clear way to prefer one over another. In some cases, a simple lexicographic order exists among the objectives, for example using plan safety as primary criterion and cost as secondary. But lexicographic order of objectives can be too rigid, not allowing any tradeoffs between objectives (e.g., a large reduction in costs for a minimal reduction in safety).

Recent work by Mouaddib used a strict lexicographic preference ordering for multi-objective MDPs (Mouaddib 2004). Others have also developed lexicographic orderings over value functions, calling this technique *ordinal dynamic programming* (Mitten 1974; Sobel 1975). Mitten assumed a specific preference ordering over outcomes for a finite horizon MDP; Sobel extended this model to infinite horizon MDPs. Ordinal dynamic programming has been explored under reinforcement learning (Gábor, Kalmár, and Szepesvári 1998; Natarajan and Tadepalli 2005), with the notion of a minimum criterion value, distinct from slack.

We propose a natural extension of sequential decision making with lexicographic order by introducing two added model components: conditioning and slack. Conditioning allows the lexicographic order to depend on certain state variables. Slack allows a small deviation from the optimal value of a primary variable so as to improve secondary value functions. The added flexibility is essential to cature practical preferences in many domains. For example, in manufacturing, there is always a tradeoff among cost, quality, and time. In critical states of the manufacturing process, one may prefer to optimize quality with no slack, whereas in less important states one may optimize cost, but allow for some slack in order to improve time and quality.

Our interest in developing this model stems from work on planning for semi-autonomous driving. Consider a car that can operate autonomously under certain conditions, for example maintaining safe speed and distance from other vehicles on a highway. All other roiad conditions require manual driving. A driver may want to minimize both the time needed to reach the destination and the effort associated with manual driving. The concepts of conditional preference and slack are quite valuable in defining the overall objective. To ensure safety, if the driver is tired, then selecting roads which are autonomous-capable are preferred without any margin of slack; however, if the driver is not tired, then roads which optimize travel time are preferred, perhaps with some slack to allow for long-distance, autonomous-capable highways. We focus on this domain for the remainder of the paper.

The general use of preference decomposition is popular within the field, as found in Generalized Additive Decomposable (GAI) networks (Gonzales, Perny, and Dubus 2011) or Conditional Preference Networks (CP-Nets) (Boutilier et al. 2004). Constrained MDPs (CMDPs) can also capture this preference structure, as well as slack, and are potentially a more general representation than LMDPs (Altman 1999); however, to our knowledge lexicographic preferences have not been explored within the model. Various other forms of slack are also commonly found in the literature (Gábor, Kalmár, and Szepesvári 1998). Combining these ideas, LMDP and LVI naturally encapsulates many problem domains.

Our primary contributions include formulating the Lexicographic MDP (LMDP) model and the corresponding Lexicographic Value Iteration (LVI) algorithm. They generalize the previous methods mentioned above with our formula-

tion of slack variables and conditional state-based preferences. We also provide an interesting connection between decision theory and game theory, in addition to a statement for the uniqueness of LMDP policies with respect to linearly weighted scalarization function policies. Furthermore, we introduce a new benchmark problem for multi-objective research: semi-autonomous driving. We implement general tools to experiment in this domain, leveraging the OpenStreetMap (OSM) package–a collaborative project to create a free editable map of the world. Finally, we employ GPU-based optimization to implement our algorithm and show its general benefits for Value Iteration (VI) in MDPs.

Section 2 states the LMDP problem definition. Section 3 presents our main convergence results, bound on slack, and an interesting relation to game theory. Section 4 discusses our experimentats within the context of semi-autonomous driving. Finally, Section 6 concludes with a final discussion of LMDPs and LVI.

2 Problem Definition

A Multi-Objective Markov Decision Process (MOMDP) is a sequential decision process in which an agent controls a domain with a finite set of states. The actions the agent can perform in each state cause a stochastic transition to a successor state. This transition results in a reward, which consistes of a vector of values, each of which depends on the state transition and action. The process unfolds over a finite or infinite number of discrete time steps. In a standard MDP, there is a single reward function and the goal is to maximize the expected cummulative discounted reward over the sequence of stages. MOMDPs present a more general model with multiple reward functions. We define below a variant of MOMDPs that we call Lexicographic MDP (LMDP), which extends MOMDPs with lexicographic preferences to also include conditional preferences and slack. We then introduce a Lexicographic Value Iteration (LVI) algorithm (Algorithm 1) which solves LMDPs.

Definition 1. A **Lexicographic Markov Decision Process (LMDP)** is a represented by a 7-tuple $\langle S, A, T, \mathbf{R}, \delta, \mathcal{S}, o \rangle$:

- S is a finite set of n states, with initial state $s_0 \in S$
- A is a finite set of m actions
- $T : S \times A \times S \rightarrow [0, 1]$ 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, $T(s, a, s') = Pr(s'|s, a)$
- $\mathbf{R} = [R_1, \ldots, R_k]^T$ is a vector of reward functions such that $\forall i \in K = \{1, \ldots, k\}$, $R_i : S \times A \times S \rightarrow \mathbb{R}$; 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 $\mathbf{R}(s, a, s') = [R_1(s, a, s'), \ldots, R_k(s, a, s')]$
- $\delta = \langle \delta_1, \ldots, \delta_k \rangle$ is a tuple of slack variables such that $\forall i \in K, \delta_i \geqslant 0$
- $\mathcal{S} = \{S_1, \ldots, S_\ell\}$ is a set which forms an ℓ-partition over the state space S
- $o = \langle o_1, \ldots, o_\ell \rangle$ is a tuple of strict preference orderings such that $L = \{1, \ldots, \ell\}$, $\forall j \in L$, o_j is a k-tuple ordering elements of K

Algorithm 1 Lexicographic Value Iteration (LVI)

```
V ← 0
V' ← 0
while ‖V − V'‖_∞^S > ε(1−γ)/γ do
    V' ← V
    V^fixed ← V
    for j = 1, ..., ℓ do
        for i = o_j(1), ..., o_j(k) do
            while ‖V_i' − V_i‖_∞^{S_j} > ε(1−γ)/γ do
                V_i'(s) ← V_i(s), ∀s ∈ S_j
                V_i(s) ← B_i V_i'(s), ∀s ∈ S_j
            end while
        end for
    end for
end while
return V'
```

We consider *infinite horizon* LMDPs (i.e., $h = \infty$), with a *discount factor* $\gamma \in [0, 1)$, due to space considerations. The finite horizon case follows in the natural way. A *policy* $\pi : S \rightarrow A$ maps each state $s \in S$ to an action $a \in A$.

Let $\mathbf{V} = [V_1, \ldots, V_k]^T$ be a set of *value functions*. Let each function $V_i^\pi : S \rightarrow \mathbb{R}$, $\forall i \in K$, 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.

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

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

LMDPs lexicographically maximize $V_{o_j(i)}(s)$ over $V_{o_j(i+1)}(s)$, for all $i \in \{1, \ldots, k-1\}$, $j \in L$, and $s \in S$, using V_{oji+1} to break ties. The model allows for slack as defined by $\eta_{o_j(i)} \geqslant 0$ (deviation from optimal for a single action change) and $\delta_{o_j(i)} \geqslant 0$ (deviation from the overall optimal value). As we show below, the classical value iteration algorithm (Bellman 1957) can be easily modified to solve MOMDPs with this preference characterization.

For the sake of readability, we use the following convention: Always assume that the ordering is present, unless otherwise stated. This allows us to omit the explicit ordering $o_j(\cdot)$ for subscripts, sets, etc. For example, $V_{i+1} \equiv V_{o_j(i+1)}$, and $\{1, \ldots, i-1\} \equiv \{o_j(1), \ldots, o_j(i-1)\}$.

First, Equation 1 defines $Q_i(s, a)$, the value of taking an action $a \in A$ in a state $s \in S$ according to objective $i \in K$.

$$Q_i(s, a) = \sum_{s' \in S} T(s, a, s')(R_i(s, a, s') + \gamma V_i(s')) \quad (1)$$

With this definition in place, we may define the aforementioned restricted set of actions for each state $s \in S$. For

$i = 1$, let $A_1(s) = A$ and for all $i \in \{1, \ldots, k-1\}$ let $A_{i+1}(s)$ be defined following Equation 2.

$$A_{i+1}(s) = \{a \in A_i(s) | \max_{a' \in A_i(s)} Q_i(s, a') - Q_i(s, a) \leqslant \eta_i\} \tag{2}$$

For reasons explained in Section 3, we let $\eta_i = (1-\gamma)\delta_i$.

Finally, let Equation 3 below be the *Bellman update equation* for MOMDPs with lexicographic reward preferences for $i \in K$, using slack $\delta_i \geqslant 0$, for all states $s \in S$. If $i > 1$, then we require V_{i-1} to have converged for all states.

$$V_i(s) = \max_{a \in A_i(s)} Q_i(s, a) \tag{3}$$

Within the algorithm, we leverage a modified value iteration with slack Bellman update equation (from Equation 3) denoted as B_i. We either use V_i for $s \in S_j \subseteq S$ or $V_i^{fixed}(s)$ for $s \in S \backslash S_j$, as shown in Equation 4 below, with $[\cdot]$ denoting Iverson brackets.

$$B_i V_i'(s) = \max_{a \in A_i(s)} \sum_{s' \in S} T(s, a, s')(R_i(s, a, s') + \gamma \bar{V}_i(s')) \tag{4}$$

$$\bar{V}_i(s') = V_i'(s')[s \in S_j] + V_i^{fixed}(s')[s \notin S_j] \tag{5}$$

Also, in each infinity norm we denote the domain of the maximization such that $\| \cdot \|_\infty^Z = \max_{z \in Z} | \cdot |$.

3 Theoretical Analysis

First we show in Proposition 1 proves that η_i from Equation 2 may be defined as $(1-\gamma)\delta_i$ to bound the final deviation from the optimal value of a state by δ_i, for $i \in K$. This is designed to be a worst-case guarantee that considers each state selects an action as far from optimal as it can, given the slack allocated to it. The accumulation of error over all states is bounded by δ; in practice, this very strong constraint can be relaxed, if desired.

Proposition 1. For all $j \in L$, for $i \in K$, assume $1, \ldots, i-1$ has converged. Let V^η be the value functions returned following Equation 4; Lines 7-10. Let V^π be the value functions returned by value iteration, following the resulting optimal policy π, starting at V^η. If $\eta_i = (1-\gamma)\delta_i$ then $\forall s \in S_j$, $V_i^\eta(s) - V_i^\pi(s) \leqslant \delta_i$.

Proof. For any $i \in K$, the full (infinite) expansion of value iteration for V_i^η is as follows ($t \to \infty$).

$$V_i^\pi(s) = \sum_{s^t \in S} T(s, \pi(s), s^t)\Big(R_i(s, \pi(s), s^t) + \gamma\Big(\cdots + \gamma\Big(\sum_{s^1 \in S} T(s^2, \pi(s^2), s^1)\Big(R_i(s^2, \pi(s^2), s^1) \tag{6}$$

$$+ \gamma\Big(V_i^0(s^1)\Big)\Big)\Big)\cdots\Big)\Big) \tag{7}$$

Since value iteration admits exactly one unique fixed point, any initial value of V_i^0 is allowed; we let $V_i^0 = V_i^\eta$. From this, $Q_i^\eta(s, \pi(s))$ (Equation 1) exists within lines 6 and 7. Also, by Equation 2, $V_i^\eta(s) - Q_i^\eta(s, \pi(s)) \leqslant \eta_i$ since $\pi(s) \in A_k(s) \subseteq \cdots \subseteq A_{i+1}(s)$. Equivalently, $Q_i^\eta(s, \pi(s)) \geqslant V_i^\eta(s) - \eta_i$. Combine all of these facts and bound it from below.

$$V_i^\pi(s) \geqslant \sum_{s^t \in S} T(s, \pi(s), s^t)\Big(R_i(s, \pi(s), s^t) + \gamma\Big(\cdots + \gamma\Big(\sum_{s^2 \in S} T(s^3, \pi(s^3), s^2)\Big(R_i(s^3, \pi(s^3), s^2) + \gamma\Big(V_i^\eta(s^2) - \eta_i\Big)\Big)\Big)\cdots\Big)\Big)$$

The η_i falls out of the inner equation. Also, recall that $\sum_{s^2 \in S} T(s^3, \pi(s^3), s^2) = 1$ and $\gamma\eta_i$ is a constant.

$$\geqslant \sum_{s^t \in S} T(s, \pi(s), s^t)\Big(R_i(s, \pi(s), s^t) + \gamma\Big(\cdots + \gamma\Big(\sum_{s^2 \in S} T(s^3, \pi(s^3), s^2)\Big(R_i(s^3, \pi(s^3), s^2) + \gamma\Big(V_i^\eta(s^2)\Big)\Big) - \gamma\eta_i\Big)\cdots\Big)\Big)$$

We may again recognize the existence of $Q_i^\eta(s^3, \pi(s^3))$ and place a lower bound on the next one with $V_i^\eta(s^3) - \eta_i$. This process repeats, each time introducing a new η_i, with one less γ multiplied in front of it, until we reach the final equation. We obtain the following inequality, and note that if $\eta_i \geqslant 0$, then we may subtract another η_i in order to obtain a geometric series (i.e., the sum may begin at $t = 0$).

$$V_i^\pi(s) \geqslant V_i^\eta(s) - \sum_{t=0}^{\infty} \gamma^t \eta_i \geqslant V_i^\eta(s) - \frac{\eta_i}{1-\gamma}$$

$$V_i^\eta(s) - V_i^\pi(s) \leqslant \frac{\eta_i}{1-\gamma}$$

Therefore, let $\eta_i = (1-\gamma)\delta_i$. This guarantees that error *for all* states $s \in S$ is bounded by δ_i. □

In order to prove convergence of Algorithm 1, we must first prove Proposition 2. It states that the value iteration component over a state partition with slack is a contraction map. The proof itself follows from value iteration (Bellman 1957) and from Russel and Norvig (2010). We include it here since there are a few important modifications required, for completeness, and it explains exactly why we must make an assumption about convergence of LVI later in this section.

Proposition 2. For all $j \in L$, for $i \in K$, assume $1, \ldots, i-1$ has converged, with discount factor $\gamma \in [0, 1)$. B_i (Equation 4) is a contraction map in the space of value functions over $s \in S_j$, i.e., $\|B_i V_1 - B_i V_2\|_\infty^{S_j} \leqslant \gamma \|V_1 - V_2\|_\infty^{S_j}$.

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

Let the space $Y_i = \mathbb{R}^z$ be the *space of value functions for* i for $z = |S_j|$, i.e., we have $V_i = [V_i(s_{j1}), \ldots, V_i(s_{jz})]^T \in Y_i$. Let the distance metric d_i be the *max norm*, i.e., $\|V_i\|_\infty = \max_{s \in S_j} |V_i(s)|$. Since $\gamma \in [0, 1)$, the metric space $M_i = \langle Y_i, d_i \rangle$ is a *complete normed metric space* (i.e., *Banach space*).

Let the lexicographic Bellman optimality equation for i (Equation 4) be defined as an operator B_i. Must show 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 $V_1, V_2 \in Y_i$ be any two value function vectors. Apply Equation 4. For $s \in S_j$, if $i = 1$ then $A_i(s) = A$; otherwise, $A_i(s)$ is defined using $A_{i-1}(s)$ (Equation 2) which by construction have converged.

$$\|B_iV_1 - B_iV_2\|_\infty^{S_j} = \max_{s \in S_j} |\max_{a \in A_i(s)} Q_1(s,a) - \max_{a \in A_i(s)} Q_2(s,a)|$$

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.

$$\begin{aligned}
&\leqslant \max_{s \in S_j} \Big| \max_{a \in A_i(s)} \Big(\sum_{s' \in S} T(s,a,s') R_i(s,a,s') \\
&\quad + \gamma \sum_{s' \in S} T(s,a,s') \bar{V}_1(s') \Big) \\
&\quad - \max_{a \in A_i(s)} \Big(\sum_{s' \in S} T(s,a,s') R_i(s,a,s') \\
&\quad - \gamma \sum_{s' \in S} T(s,a,s') \bar{V}_2(s') \Big) \Big| \\
&\leqslant \max_{s \in S_j} \Big| \gamma \max_{a \in A_i(s)} \sum_{s' \in S} T(s,a,s') \bar{V}_1(s') \\
&\quad - \gamma \max_{a \in A_i(s)} \sum_{s' \in S} T(s,a,s') \bar{V}_2(s') \Big|
\end{aligned}$$

First, we can pull out γ. Recall, also that for any two functions f and g, $|\max_x f(x) - \max_x g(x)| \leqslant \max_x |f(x) - g(x)|$. After applying this property, we note that $T(\cdot)$ lies on an n-simplex, which forms a simple convex polytope after scaling the value function difference. Convex polytopes obtain their maximum values at the vertices, allowing us to select the maximal state $s \in S$ instead.

$$\begin{aligned}
&\leqslant \gamma \max_{s \in S_j} \max_{a \in A_i(s)} \Big| \sum_{s' \in S} T(s,a,s') (\bar{V}_1(s') - \bar{V}_2(s')) \Big| \\
&\leqslant \gamma \max_{s \in S_j} \max_{a \in A_i(s)} \sum_{s' \in S} T(s,a,s') \Big| \bar{V}_1(s') - \bar{V}_2(s') \Big|
\end{aligned}$$

Now, apply Equation 5 and notice that the $V_{1i}^{fixed}(s') = V_{2i}^{fixed}(s')$, $\forall s' \in S \backslash S_j$. These terms cancel, and we are left with the difference of V_1 and V_2 over S_j.

$$\begin{aligned}
&\leqslant \gamma \max_{s \in S_j} \max_{a \in A_i(s)} \sum_{s' \in S_j} T(s,a,s') \Big| V_1(s') - V_2(s') \Big| \\
&\leqslant \gamma \max_{s \in S_j} \Big| V_1(s) - V_2(s) \Big| \\
&\leqslant \gamma \|V_1 - V_2\|_\infty^{S_j}
\end{aligned}$$

This proves that the operator B_i is a contraction map on metric space M_i, for all $i \in K$. □

Following the same logic as Bellman's optimality equation, we may guarantee convergence to within $\epsilon > 0$ of the fixed point (Corollary 3).

Proposition 3. For all $j \in L$, for $i \in K$, assume $1, \ldots, i-1$ has converged. Following Equation 4; Lines 7-10, for any $i \in K$, B_i converges to within $\epsilon > 0$ of a unique fixed point once $\|V_i^{t+1} - V_i^t\|_\infty^{S_j} < \epsilon \frac{1-\gamma}{\gamma}$ for iteration $t > 0$.

Proof. Expanding upon Proposition 2, for all $i \in K$, by definition of a contraction map, B_i admits at most one fixed point. By *Banach's fixed point theorem*, since M_i is a complete metric space and B_i is a contraction map on Y_i, B_i admits a unique fixed point $V_i^* \in Y_i$. Therefore, the final V^* is a unique fixed point in the space over all value functions over $i \in K$ and $s \in S_j$.

Finally, 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).

$$\begin{aligned}
d(x^*, x_{t+1}) &\leqslant \frac{\alpha}{1-\alpha} d(x_{t+1}, x_t) \\
\|V_i^* - V_i^{t+1}\|_\infty^{S_j} &\leqslant \frac{\gamma}{1-\gamma} \|V_i^{t+1} - V_i^t\|_\infty^{S_j}
\end{aligned}$$

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

$$\begin{aligned}
\epsilon &\leqslant \frac{\gamma}{1-\gamma} \|V_i^{t+1} - V_i^t\|_\infty^{S_j} \\
\epsilon \frac{1-\gamma}{\gamma} &\leqslant \|V_i^{t+1} - V_i^t\|_\infty^{S_j}
\end{aligned}$$

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.

$$\|V_i^{t+1} - V_i^t\|_\infty^{S_j} < \epsilon \frac{1-\gamma}{\gamma}$$

□

With our propositions for LVI with slack and fixed states in place, we now prove that LVI itself converges in Proposition 4.

Proposition 4. LVI (Algorithm 1) converges to a unique fixed point V^* in the space of value functions given that for each $j \in L, 1, \ldots, i-1$ has converged over $s \in S_j$, with discount factor $\gamma \in [0, 1)$.

Proof. Let $Y = \mathbb{R}^{k \times n}$ be the space of all value functions over $i \in K$ and $s \in S$, i.e., for all $V \in Y$ we have the following.

$$V = \begin{bmatrix} V_1(s_1) & \cdots & V_1(s_n) \\ \vdots & \ddots & \vdots \\ V_k(s_1) & \cdots & V_k(s_n) \end{bmatrix}$$

Let distance metric d be the *max norm*, i.e., $\|V\|_\infty = \max_{i \in K} \max_{s \in S} |V_i(s)|$. Thus, the metric space $M = \langle Y, d \rangle$ is a *Banach space*.

Let G be an operator in M following Lines 4-13 in Algorithm 1, i.e., $V' = GV$ using the algorithm's variables: V and V'. Must show that G is a contraction map. Let $V_1, V_2 \in Y$ be any two value function vectors.

$$\|GV_1 - GV_2\|_\infty = \max_{i \in K} \max_{s \in S} |(GV_1)_i(s) - (GV_2)_i(s)|$$

Since we have partitioned S into $S_1, \ldots, S_\ell$, we may break the $\max$ apart. Then, apply the fact that for all $i \in K$, Lines 7-10 each modify a different partition's states $s \in S$ of $V_i(s)$.

$$\begin{aligned}\|GV_1 - GV_2\|_\infty &= \max_{i \in K} \max_{j \in L} \max_{s \in S_j} |(GV_1)_i(s) - (GV_2)_i(s)| \\ &= \max_{i \in K} \max_{j \in L} \max_{s \in S_j} |B_i V_{1i}(s) - B_i V_{2i}(s)|\end{aligned}$$

This is equivalent to the infinity norm over S_j, which enables us to apply Proposition 2.

$$\begin{aligned}\|GV_1 - GV_2\|_\infty &= \max_{i \in K} \max_{j \in L} \|B_i V_{1i} - B_i V_{2i}\|_\infty^{S_j} \\ &\leqslant \gamma \max_{i \in K} \max_{j \in L} \|V_{1i} - V_{2i}\|_\infty^{S_j}\end{aligned}$$

Next, we may write the infinity norm definition over S_j, recognizing that if $s \notin S_j$, then $|V_{1i}^{fixed}(s) - V_{2i}^{fixed}(s)| = 0$. Thus, we can simply maximize over all states S. After this, we apply the definition of our *max norm* distance metric and obtain our desired result.

$$\begin{aligned}\|GV_1 - GV_2\|_\infty &\leqslant \gamma \max_{i \in K} \max_{j \in L} \max_{s \in S_j} |V_{1i}(s) - V_{2i}(s)| \\ &\leqslant \gamma \max_{i \in K} \max_{s \in S} |V_{1i}(s) - V_{2i}(s)| \\ &\leqslant \gamma \|V_1 - V_2\|_\infty\end{aligned}$$

Therefore, G is a contraction map in the space of both K and S, so G admits at most one fixed point. By *Banach's fixed point theorem*, since M is a complete metric space and G is a contraction map on Y, G admits a unique fixed point $V^* \in Y$. Therefore, the final V^* is a unique fixed point in the space over all value functions over $i \in K$ and $s \in S$. □

We have guaranteed convergence of LVI to the optimal policy. Interestingly, conditioning the value function preference on one particular subset of the states, introduces a connection between LVI for LMDPs and game theory. First, we recognize a mapping from the LMDP to a *normal form game* in Definition 2. Then, we show in Proposition 5 that the resulting LMDP's optimal policy computed from LVI is, in fact, a Nash equilibrium of the normal form.

Definition 2. Let LMDP $\langle S, A, T, \mathbf{R}, \delta, \mathcal{S}, o \rangle$ have value functions V^π for corresponding optimal policy π, the optimal value functions V^η, computed via LVI (Algorithm 1).

Let $\bar{s} = \langle s_1, \ldots, s_\ell \rangle$ be any tuple of states such that $\forall z \in L$, $s_z \in S_z$, and let $\bar{i} = \langle i_1, \ldots, i_\ell \rangle$ be any tuple of indices $i_z \in K$.

The **LMDP's Normal Form Game** $\langle L, \mathcal{A}, U \rangle$ is:

- $L = \{1, \ldots, \ell\}$ is a finite set of agents, one for each partition in X
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_\ell$ is a finite set of joint actions, such that $\forall z \in L, x = |S_z|, S_z = \{s_{z1}, \ldots, s_{zx}\}, \mathcal{A}_z = A_{i_z}(s_{z1}) \times \cdots \times A_{i_z}(s_{zx})$
- $U = \langle u_1, \ldots, u_\ell \rangle$ is a set of utility functions, such that $\forall z \in L$, $\forall a \in \mathcal{A}$, $s_z \in \bar{s}$, $u_z(a_z, a_{-z}) = \min\{V_{i_z}^{\pi, a_z}(s_z), V_{i_z}^\eta(s_z) - \delta_{i_z}\}$.

Note that we only consider pure strategy profiles, and thus a player's *strategy set* will be used synonymously with its *action set*. Similarly, since we also consider one-stage games, *strategy profile* will be used synonymously with *action profile*.

Proposition 5. For LMDP $\langle S, A, T, \mathbf{R}, \delta, \mathcal{S}, o \rangle$, let π be the optimal policy computed using LVI (Algorithm 1). Let $f(\pi) = \langle \omega_1, \ldots, \omega_\ell \rangle$ so that $\forall z \in L$, $x = |S_z|$, $S_z = \{s_{z1}, \ldots, s_{zx}\}$, $\omega_z = \langle \pi(s_{z1}), \ldots, \pi(s_{zx}) \rangle$. Applying the transformation in Definition 2, the strategy (action) profile $f(\pi) = a = (a_z, a_{-z}) \in \mathcal{A}$ is a weak pure strategy Nash equilibrium.

Proof. By the definition of a (weak) Nash equilibrium, must show that $\forall z \in L$, $\forall a' \in \mathcal{A}_z$, $u_z(a_z, a_{-z}) \geqslant u_z(a', a_{-z})$.

Let π' be the corresponding policy for $f(\pi') = \langle a_1, \ldots, a_{z-1}, a', a_{z+1}, \ldots, a_\ell \rangle \in \mathcal{A}$. Recall that $a' = \langle a'_1, \ldots, a'_x \rangle$, $x = |S_z|$.

Let $V^{\pi'}$ be the value functions after value iteration has converged for each value function in K following policy π'. Note that $\mathbf{V}_x^\pi = \mathbf{V}_x^{\pi'}$, $\forall x \in \{1, \ldots, i_z - 1\}$, and therefore by Equation 2, $A_{i_z}^\pi = A_{i_z}^{\pi'}$. This ensures that we may use Proposition 1, by simply considering a MOMDP with a reduced number of rewards up to i_z.

By Proposition 1, $V_{i_z}^\pi(s_z) \geqslant V_{i_z}^\eta(s_z) - \delta_i$. Apply the fact that π is defined following action a_z, so $V_{i_z}^{\pi, a_z}(s_z) = V_{i_z}^\pi(s_z)$. Thus, by Definition 2:

$$u_z(a_z, a_{-z}) = \min\{V_{i_z}^\pi(s_z), V_{i_z}^\eta(s_z) - \delta_i\} = V_{i_z}^\eta(s_z) - \delta_i$$

Similarly, there are two cases for $V_{i_z}^{\pi'}(s_z)$.

Case 1: $V_{i_z}^{\pi'}(s_z) \geqslant V_{i_z}^\eta(s_z) - \delta_i$. Which implies:

$$u_z(a', a_{-z}) = \min\{V_{i_z}^{\pi'}(s_z), V_{i_z}^\eta(s_z) - \delta_i\} = V_{i_z}^\eta(s_z) - \delta_i$$

Therefore, $u_z(a_z, a_{-z}) = u_z(a', a_{-z})$.

Case 2: $V_{i_z}^{\pi'}(s_z) < V_{i_z}^\eta(s_z) - \delta_i$. Which implies:

$$u_z(a', a_{-z}) = \min\{V_{i_z}^{\pi'}(s_z), V_{i_z}^\eta(s_z) - \delta_i\} = V_{i_z}^{\pi'}(s_z)$$

Therefore, $u_z(a_z, a_{-z}) > u_z(a', a_{-z})$.

In both cases, the inequality $u_z(a_z, a_{-z}) \geqslant u_z(a', a_{-z})$ is true. Therefore, the strategy (action) profile π is a Nash equilibrium.

□

So far, we have shown some convergence properties of LVI in an LMDP, and a connection to game theory. We will now prove a uniqueness property in relation of linearly weighted scalarization functions; in particular, there exists LMDPs such that for the corresponding MOMDP, no weight exists which would allow VI to return the same policy as LVI.

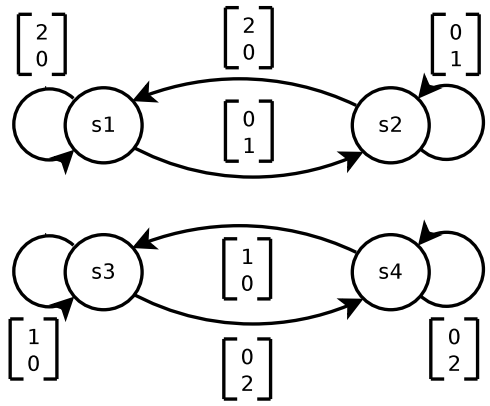

Figure 1: The example MOMDP as described in Proposition 6.

Proposition 6. There exist LMDPs, with a policy following LVI π_{lvi}, such that for all weights $\mathbf{w}$, the corresponding scalarized MOMDP's policy $\pi_{\mathbf{w}}$ is not equal to π_{lvi}.

Proof. Consider the MOMDP depicted in Figure 1, with states $S = \{s_1, s_2, s_3, s_4\}$, $A = \{stay, leave\}$, $T(s, stay, s') = 1$ if $s = s'$, $T(s, leave, s') = 1$ if $s \neq s'$, $T(s, a, s') = 0$ otherwise, and rewards $\mathbf{R} = [R_1, R_2]^T$ as shown.

(a) We now prove that for $w_1 < 1/3$ and $w_2 > 2/3$, $\pi_{\mathbf{w}}(s_1) = leave$ and $\pi_{\mathbf{w}}(s_2) = stay$. Proof by induction on t.

Base Case: $t = 1$.

$$\begin{aligned} Q^1_{\mathbf{w}}(s_1, stay) &= 2w_1 + \gamma V^0_{\mathbf{w}}(s_1) \\ &< w_2 + \gamma V^0_{\mathbf{w}}(s_2) = Q^1_{\mathbf{w}}(s_1, leave) \\ Q^1_{\mathbf{w}}(s_2, stay) &= w_2 + \gamma V^0_{\mathbf{w}}(s_2) \\ &> 2w_1 + \gamma V^0_{\mathbf{w}}(s_1) = Q^1_{\mathbf{w}}(s_2, leave) \\ \Rightarrow \quad & \pi^1_{\mathbf{w}}(s_1) = leave \qquad \pi^1_{\mathbf{w}}(s_2) = stay \\ \Rightarrow \quad & V^1_{\mathbf{w}}(s_1) = w_2 \qquad V^1_{\mathbf{w}}(s_2) = w_2 \end{aligned}$$

Thus, the base case holds true. Note: Both value functions have the same value.

Induction Step: Assume true for $t = T$, must show for $t = T + 1$.

$$\begin{aligned} Q^{T+1}_{\mathbf{w}}(s_1, stay) &= 2w_1 + \gamma V^T_{\mathbf{w}}(s_1) = 2w_1 \sum_{t=0}^{T} \gamma^t \\ &< w_2 + \gamma V^T_{\mathbf{w}}(s_2) = w_2 \sum_{t=0}^{T} \gamma^t = Q^{T+1}_{\mathbf{w}}(s_1, leave) \\ \Rightarrow \quad & \pi^{T+1}_{\mathbf{w}}(s_1) = leave \qquad \pi^{T+1}_{\mathbf{w}}(s_2) = stay \\ \Rightarrow \quad & V^{T+1}_{\mathbf{w}}(s_1) = w_2 \sum_{t=0}^{T} \gamma^t \qquad V^{T+1}_{\mathbf{w}}(s_2) = w_2 \sum_{t=0}^{T} \gamma^t \end{aligned}$$

By induction, $\forall t \in \mathbb{N}$, $\pi_{\mathbf{w}}(s_1) = leave$ and $\pi_{\mathbf{w}}(s_2) = stay$ for weights $w_1 < 1/3$ and $w_2 > 2/3$.

(b) Apply the same logic for $w_1 > 1/3$ and $w_2 < 2/3$ to obtain the reverse policy: $\pi_{\mathbf{w}}(s_1) = stay$ and $\pi_{\mathbf{w}}(s_2) = leave$.

(c) Ambiguity exists at $w_1 = 1/3$ and $w_2 = 1/3$, wherein we must employ a tie-breaking rule. Under this scenario, we can construct a policy such that $\pi_{\mathbf{w}}(s_1) = \pi_{\mathbf{w}}(s_2) = stay$.

(d) Note that (a) and (b) are reversed for states s_3 and s_4 by applying the exact same logic again for these states. This means that for $w_1 < 2/3$ and $w_2 < 1/3$ it implies $\pi_{\mathbf{w}}(s_3) = leave$ and $\pi_{\mathbf{w}}(s_4) = stay$; for $w_1 > 2/3$ and $w_2 < 1/3$ it implies $\pi_{\mathbf{w}}(s_3) = stay$ and $\pi_{\mathbf{w}}(s_4) = leave$. Similar to (c), only with the weights defined as $w_1 = 2/3$ and $w_2 = 1/3$, ambiguity allows for a tie-breaking rule to produce $\pi_{\mathbf{w}}(s_3) = \pi_{\mathbf{w}}(s_3) = stay$.

(e) Combine (a)-(d), realizing that for states $S = \{s_1, s_2, s_3, s_4\}$, no weight exists to produce the policy $\pi_{\mathbf{w}}(s) = stay$ for all $s \in S$.

(f) Must show that LVI in an LMDP using these states, actions, transitions, and rewards can return the policy $\pi_{lvi}(s) = stay$ for all $s \in S$.

□

Finally, we note that it is easy to show that Algorithm 1 generalizes value iteration (VI) and other forms of LVI (Gábor, Kalmár, and Szepesvári 1998).

4 Experimentation

Semi-autonomous systems explore efficient policy execution involving the optimal transfer of control between human and agent. Our focus is on semi-autonomous driving, namely the optimal policy to adapt to a human's level of fatigue.

The LMDP consists of states formed by a 4-tuple: current intersection, previous intersection, driver tired (true/false), and autonomy (enabled/disabled). Actions are taken at intersections, as found in path planning in graphs (e.g., GPS). Due to real world limitations, autonomy can only be enabled on main roads and highways; in our case, this means a speed limit greater than or equal to 30 miles per hour.

Actions are simply which road to take at each intersection and, if the road allows, whether or not to enable autonomy. We aim to follow the extensive body of engineering and psychological research on driver fatigue (Ji, Zhu, and Lan 2004). As such, the stochasticity within state transitions considers the probability that the driver moves from awake to tired, with a probability of 0.9.

There are two reward functions: time and autonomy. The time reward (cost) is proportional to the time spent on the road (in seconds), plus a small constant expected value of 5 (seconds) to model time spent at a traffic light, slowing down turning, or waiting for others. The autonomy reward (cost) is proportional to the time spent on the road, but is only applied if the driver is tired; otherwise, there is an epsilon penalty. For both rewards, the goal state is an absorbing state which awards zero.

It is natural to define the problem in terms of strictly optimizing time when the driver is awake, and strictly optimizing autonomy when the driver is tired. We also allow for a 10

Figure 2: The policy for awake (above) versus tired (below).

second slack in the expected time to reach the goal (i.e., time reward), in order to favor the ease of autonomy if available.

Figure 2 shows an example optimal policy returned by LVI for a few roads north of Boston Commons. Each road at an intersection has an action, denoted by the arrows. Green arrows denote that autonomy was disabled; purple arrows denote that autonomy was enabled. The blue colored roads show autonomy-capability.

5 Discussion

6 Conclusion

7 Acknowledgements

References

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

Bellman, R. E. 1957. *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135–191.

Calisi, D.; Farinelli, A.; Iocchi, L.; and Nardi, D. 2007. Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics* 24(8-9):763–777.

Castelletti, A.; Pianosi, F.; and Soncini-Sessa, R. 2008. Water reservoir control under economic, social and environmental constraints. *Automatica* 44(6):1595 – 1607.

Gábor, Z.; Kalmár, Z.; and Szepesvári, C. 1998. Multi-criteria reinforcement learning. In *Proceedings of the International Conference on machine Learning*, volume 98, 197–205.

Gonzales, C.; Perny, P.; and Dubus, J. P. 2011. Decision making with multiple objectives using GAI networks. *Artificial Intelligence* 175(78):1153 – 1179.

Ji, Q.; Zhu, Z.; and Lan, P. 2004. Real-time nonintrusive monitoring and prediction of driver fatigue. *Vehicular Technology, IEEE Transactions on* 53(4):1052–1068.

Kwak, J.-Y.; Varakantham, P.; Maheswaran, R.; Tambe, M.; Jazizadeh, F.; Kavulya, G.; Klein, L.; Becerik-Gerber, B.; Hayes, T.; and Wood, W. 2012. SAVES: A sustainable multiagent application to conserve building energy considering occupants. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 21–28.

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 *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, 2814–2819.

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.

Perny, P., and Weng, P. 2010. On finding compromise solutions in multiobjective Markov decision processes. In *Proceedings of the European Conference on Artificial Intelligence*, 969–970.

Perny, P.; Weng, P.; Goldsmith, J.; and Hanna, J. 2013. Approximation of Lorenz-optimal solutions in multiobjective Markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 92–94.

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.

Russell, S., and Norvig, P. 2010. *Artificial Intelligence: A modern approach*. Upper Saddle River, New Jersey: Prentice Hall.

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

Supplement to Submission

This section includes supplemental matterial for the paper