Solution methods for a class of finite-horizon vector-valued Markov decision processes

Anas Mifrani, Philippe Saint-Pierre, Nicolas Savy

▶ **To cite this version:**

Anas Mifrani, Philippe Saint-Pierre, Nicolas Savy. Solution methods for a class of finite-horizon vector-valued Markov decision processes. INFOR: Information Systems and Operational Research , 2025, ⟨10.1080/03155986.2025.2484050⟩. ⟨hal-04924721v2⟩

HAL Id: hal-04924721

https://hal.science/hal-04924721v2

Submitted on 29 Apr 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HAL Authorization

SOLUTION METHODS FOR A CLASS OF FINITE-HORIZON VECTOR-VALUED MARKOV DECISION PROCESSES

ANAS MIFRANI*, PHILIPPE SAINT-PIERRE, NICOLAS SAVY

Abstract. This paper investigates and develops solution methods for a class of finite-horizon Markov decision processes characterized by additive or multiplicative vector rewards. Two concepts of optimality are treated: (1) optimality in the space of return vectors, whereby a policy is optimal if it delivers a maximal total reward from any initial state; and (2) optimality in the space of return functions, whereby a policy is optimal if its total reward function is maximal among all total reward functions. The paper elucidates the relation between the two concepts, proposes a procedure for utilizing this relation to determine the set of optimal policies under concept (1), and formulates a dynamic programming approach to calculating optimal policies under concept (2). The paper demonstrates that dynamic programming yields all optimal policies under concept (2). The paper's results are illustrated with numerical experiments and a multi-objective stochastic inventory control problem.

Keywords: Multi-objective Markov decision processes; vector maximization; dynamic programming; multiple-criteria decision analysis.

The Version of Record of this manuscript was published on April 5, 2025, and is available in *INFOR: Information Systems and Operational Research*, `https://www.tandfonline.com/doi/full/10.1080/`, along with the supplemental material.

1. Introduction

Markov decision processes offer a mathematical framework for modeling and solving sequential decision-making problems where outcomes are uncertain. There are three components to such a process: a stochastic dynamical system to be controlled over a period of $N \geq 1$ epochs; real-valued rewards accrued between consecutive epochs as a result of decisions taken at epochs; and a control policy that prescribes actions such that the total expected reward (or cost) for (of) operating the system is maximized (minimized). Viewed in this way, a Markov decision process defines a single-objective, discrete-time optimal control problem.

*: Corresponding author.

Authors' affiliation: Université de Toulouse, Institut de Mathématiques de Toulouse, F-31062 Toulouse Cedex 9, France.

Email addresses: anas.mifrani@math.univ-toulouse.fr; philippe.saint-pierre@math.univ-toulouse.fr; nicolas.savy@math.univ-toulouse.fr

However, a number of decision-making problems are inherently multi-objective. In administering chemotherapy, for instance, an oncologist wants to maximize the probability of cure while minimizing damage to normal cells (Coldman & Murray, 2000). A logistics manager looks for measures that simultaneously minimize the costs of warehousing and transportation over the coming year (Burns, Hall, Blumenfeld, & Daganzo, 1985). And in planning periodic pavement rehabilitation, the local government wants to ensure the highest quality roads for its citizens with minimal maintenance expenditures (Golabi, Kulkarni, & Way, 1982). While the dynamics of such problems may lend themselves to Markov decision process formulation (Puterman, 2014), it is sometimes unclear how the various objectives involved – e.g., probability of cure versus damage to normal cells, warehousing versus transportation costs, and road quality versus maintenance expenses – can be condensed into a single scalar-valued reward (or cost) function (Brown & Strauch, 1965; Zadeh, 1963). Interest in overcoming this issue, and therefore in expanding the use of Markov decision processes to multi-objective decision-making, led to the introduction of vector-valued Markov decision processes.

In a vector-valued Markov decision process, rewards take values in $\mathbb{R}^m$, $m \geq 1$, with each reward component representing an optimization objective. The standard formulation is as follows. Let S denote the set of states the system can occupy throughout its lifetime. For any state s, let A_s be the set of actions available in s; $R_t(s,a) = (r_t(s,a)_1, ..., r_t(s,a)_m)$ the reward for choosing $a \in A_s$ in s at time $t = 1, ..., N-1$, and $R_N(s)$ the reward for occupying state s at the terminal epoch; $p_t(j|s,a)$ the transition probability from s to $j \in S$ if $a \in A_s$ was chosen at time t; and $u_t^\pi(s) = \mathbb{E}_\pi^s[\sum_{i=t}^{N-1} R_i(X_i, d_i(X_i)) + R_N(X_N)] \in \mathbb{R}^m$ the expected total reward for using a Markovian deterministic policy π from t onward given $X_t = s$, where X_i represents the (random) state at time i and $d_i(X_i)$ the action prescribed by π for X_i at time i. For any state s and epoch t, we call the vector $u_t^\pi(s)$ a policy return. In particular, $u_1^\pi(s)$ represents the return of a policy π over the entire decision-making horizon. When $m = 1$, this model reduces to a Markov decision process.

The reward structure just described induces a partial order on the set of policies whereby a policy π may be superior to a policy π' in some respects but inferior to π' in others. For example, taking $m = 2$ and a common initial state s, we may have $u_1^\pi(s)_1 \geq u_1^{\pi'}(s)_1$ yet $u_1^\pi(s)_2 < u_1^{\pi'}(s)_2$, so that, componentwise, we neither have $u_1^\pi(s) \geq u_1^{\pi'}(s)$ nor $u_1^{\pi'}(s) \geq u_1^\pi(s)$. In a standard Markov decision process ($m = 1$), this situation obviously never arises.

Though Brown and Strauch (1965) were the first to consider a Markov decision process with partially ordered rewards, namely rewards in multiplicative lattices, the chief theoretical developments concerning vector-valued Markov decision processes as presented here occurred in papers published between the 1970s and 1980s (Furukawa, 1980; Henig, 1983; White, 1982). To the best of our knowledge, D. J. White's seminal paper (White, 1982) contains the first attempt at formulating an exact dynamic programming approach to solving a class of

vector-valued Markov decision processes. This approach has been cited by a recent survey of multi-objective reinforcement learning (Hayes et al., 2022), and numerous authors have used it either to compare it experimentally with their own approaches (Roijers, Röpke, Nowe, & Radulescu, 2021; Wiering & De Jong, 2007) or as a point of departure for the development of new algorithms (Chen, Trevizan, & Thiébaux, 2023; Mandow, Pérez-de-la Cruz, & Pozas, 2022; Ruiz-Montiel, Mandow, & Pérez-de-la Cruz, 2017; Van Moffaert & Nowé, 2014). It is based on the following vector analogue of the Bellman equations of a finite-horizon Markov decision process (Puterman, 2014, Chapter 4):

$$U_t(s) = e\left(\bigcup_{a\in A_s}\left(\{R_t(s,a)\}\bigoplus\sum_{j\in S} p_t(j|s,a)U_{t+1}(j)\right)\right);\ \ t<N \tag{1}$$

$$U_t(s) = \{R_N(s)\};\ \ t=N \tag{2}$$

for all $s\in S$ and $t=1,...,N$, where $e(X)$ denotes the Pareto efficient subset of a set $X\subseteq\mathbb{R}^m$ (see Section 2 for a formal definition), $A\bigoplus B=\{a+b:\forall a\in A,\ \forall b\in B\}$ for any two nonempty sets A and B, and where the unknowns are the $U_t(s)$'s, $s\in S$, $t=1,...,N$.

White claims that the solutions of Equations (1) and (2) are the Pareto efficient sets of policy returns for all epochs and initial states, i.e., $U_t(s)=e\big(\bigcup_\pi\{u_t^\pi(s)\}\big)$ for all $t\le N$ and $s\in S$ (White, 1982, Theorem 2). In fact this claim is generally false (Mifrani, 2025), notwithstanding its coincidence, for $m=1$, with the correct observation that the solutions of the Bellman equations are the $\max_\pi u_t^\pi(s)$'s (Puterman, 2014, Proposition 4.3.3.).

We might note, in passing, that such issues do not arise in infinite horizon models ($N=\infty$). Furukawa (1980) has proved that the fixed-point characterization of a Markov decision process's optimal infinite horizon value (Puterman, 2014, Theorem 6.2.6.) extends *mutatis mutandis* to vector-valued processes. In short, the infinite horizon counterparts of White's equations are valid. Here we shall confine our analysis to finite horizon models.

In this paper, we take the position of a decision maker who has to select a *V-optimal* (V for vector-based) Markovian deterministic policy, that is, a policy which generates an efficient return from any initial state. We develop an approach to computing such a policy that does not involve dynamic programming on the space of return vectors. The key role in this approach is played by an auxiliary optimality criterion that we call *F-optimality* (F for function-based). The difference between the two criteria lies in that, to compare a pair of policies π and π', F-optimality focuses on the policies' return *functions*, u_t^π and $u_t^{\pi'}$, $t=1,...,N$, rather than on the return *vectors* $u_t^\pi(s)$, $u_t^{\pi'}(s)$ achieved in individual states s. The subtlety of this distinction will be illustrated in Example 1 of Section 3.

We establish the following: (1) V-optimality is subsumed under F-optimality; (2) F-optimality is susceptible to dynamic programming; (3) the solutions to the dynamic programming equations can be leveraged to construct F-optimal policies; and (4) provided there is a finite number

of F-optimal policies, a computationally useful characterization of V-optimal policies within the set of F-optimal policies can be implemented to find all policies of the former kind. Thus, in particular, we shall see that all V-optimal policies can be calculated without evaluating the $e\big(\bigcup_\pi\{u_1^\pi(s)\}\big)$'s, a potentially intractable task in the absence of a valid recurrence relation between $e\big(\bigcup_\pi\{u_t^\pi(s)\}\big)$ and $e\big(\bigcup_\pi\{u_{t+1}^\pi(j)\}\big)$ for all $s, j \in S$ and $t = 1, ..., N-1$.

The hypotheses and notation underpinning this paper are presented in greater detail in Section 2. In Section 3, we shall substantiate, and discuss the implications of, points (1)-(4) as outlined above. In particular, we shall devise algorithms for computing policies according to each criterion. A numerical analysis of the algorithms is undertaken in Section 3. Section 4 reports implementation results for a multi-objective stochastic inventory management problem. In Section 5, we consider the ramifications of our results for models with multiplicative – rather than additive – rewards, make some general comments on the algorithms, and close with a discussion of potential applications of these results.

2. Model Assumptions and Notation

At each epoch $t \leq N$, the system occupies a state s_t. The set of all states, S, is finite. The decision maker has at their disposal a set of actions, A, which they must choose from at each epoch. If only certain actions are allowed in a state, let A_s be the set of permissible actions in $s \in S$, from which it follows $A = \cup_{s\in S} A_s$. Suppose A_s is a compact subset of $\mathbb{R}$ for all $s \in S$. Assuming $a \in A$ was selected at time $t < N$, the probability that the system will occupy state $j \in S$ at $t+1$ depends only on the present state $s \in S$, and is denoted by $p_t(j|s,a)$. For choosing action $a \in A$ in state s at time $t < N$, the decision maker receives a vector reward $R_t(s,a) \in \mathbb{R}^m$, $m \geq 2$. Suppose that transition probabilities and rewards are continuous on A_s for all $s \in S$. A (Markovian, deterministic) decision rule d_t dictates the action to be taken in each state at epoch $t < N$, and is viewed therefore as a mapping from S to A. The set of all decision rules, D, is considered to be compact. For any $t < N$ and any $d_t \in D$, let $P^{d_t} = (p_t(j|s, d_t(s))_{s,j\in S}$ be the transition probability matrix induced by d_t. No decision is taken at epoch N, but a state-dependent reward $R_N(s)$ is generated. A policy specifies the decision rule that should be used at each epoch, and shall be identified with its corresponding sequence of decision rules $(d_1, ..., d_{N-1})$. Let $\Pi = D^{N-1}$ be the set of all policies. For any $\pi \in \Pi$ and $t < N$, $\bar{\pi}(t) = (d_t, ..., d_{N-1})$ shall denote the portion of decision rules used by π from t onward.

For any policy $\pi = (d_1, ..., d_{N-1})$ and any $t < N$, we have the recurrence relation

$$u_t^\pi(s) = R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s))u_{t+1}^\pi(j) \tag{3}$$

where we let

$$u_N^\pi(s) = R_N(s). \tag{4}$$

Expanding the sum in (3) over all future epochs and states yields the expression

$$u_t^\pi(s) = R_t(s, d_t(s)) + \sum_{i=t}^{N-2} \sum_{j \in S} \left(\prod_{k=t}^{i} P^{d_k} \right)_{s,j} R_{i+1}(j, d_{i+1}(j)) + \sum_{j \in S} \left(\prod_{k=t}^{N-1} P^{d_k} \right)_{s,j} R_N(j). \quad (5)$$

The terms "policy return" and "return", where time and state are omitted for brevity, shall refer to any vector $u \in \mathbb{R}^m$ for which there is a policy π, a time $t = 1, ..., N$ and an $s \in S$ with $u = u_t^\pi(s)$. When it is necessary to distinguish between the function u_t^π and the values $u_t^\pi(s)$ it takes at particular states $s \in S$, the phrases "(policy) return function" and "(policy) return vector" shall be used instead, with time and state also omitted for brevity.

For any partially ordered set $(X, \geq)$, let $e(X)$ be the efficient (or admissible, or noninferior, or Pareto optimal) subset of X, to wit:

$$e(X) = \{\, x \in X : \forall y \in X,\, y \geq x \implies y = x \,\}. \quad (6)$$

Let $F(S, \mathbb{R}^m)$ denote the set of all $\mathbb{R}^m$-valued functions on S. In Section 3 we shall be concerned with efficiency in subsets of $(\mathbb{R}^m, \geq)$ and $(F(S, \mathbb{R}^m), \succeq)$, where:

$$\forall x, y \in \mathbb{R}^m,\, x \geq y \iff \forall i = 1, ..., m,\, x_i \geq y_i, \quad (7)$$

$$\forall u, v \in F(S, \mathbb{R}^m),\, u \succeq v \iff \forall s \in S, u(s) \geq v(s). \quad (8)$$

The partial orders thus defined provide a means for comparing, respectively, return vectors and return functions. A strict partial order $>$ can also be defined on $(X, \geq)$ as $\forall x, y \in X, x > y \iff x \geq y \wedge x \neq y$. When $X \subseteq \mathbb{R}^m$ and $\mathbb{R}^m$ is equipped with (7), the elements of $e(X)$ are sometimes referred to as "vector maxima" (Geoffrion, 1968), though for consistency with previous work on vector-valued Markov decision processes the generic adjective "efficient" shall be used instead. When X has a maximum, such as is the case with $\bigcup_{\pi \in \Pi} \{u_t^\pi(s)\}$ for $m = 1$ (Puterman, 2014, Proposition 4.4.3), we have $e(X) = \{\max(X)\}$.

3. Theoretical Results

As stated in the Introduction, we shall study two related concepts of optimality as regards policies. In the first concept, a policy is optimal if, whatever the state in which it was first implemented, it delivers a maximal return over the N epochs:

Definition 1 (V-optimality)**.** *A policy π^V is V-optimal if and only if $u_1^{\pi^V}(s) \in e\big(\bigcup_\pi \{u_1^\pi(s)\}\big)$ for all states $s \in S$.*

Here "V" stands for "vector", and $\bigcup_\pi \{u_1^\pi(s)\}$ is endowed with $\geq$ as defined in (7). For $m = 1$, $\max_\pi u_1^\pi(s)$ exists for all $s \in S$ (Puterman, 2014, Proposition 4.3.3.), and Definition 1 reads "π^V is V-optimal if and only if $u_1^{\pi^V}(s) = \max_\pi u_1^\pi(s)$ for all $s \in S$", which is the standard optimality criterion across a wide range of Markov decision process applications (Borrero & Akhavan-Tabatabaei, 2013; Goedhart, Haijema, Akkerman, & de Leeuw, 2023; Mason, Denton,

Shah, & Smith, 2014; Puterman, 2014; Ramirez-Nafarrate, Hafizoglu, Gel, & Fowler, 2014; Schlosser & Gönsch, 2023).

One of the aims of this section is to supply a procedure for determining all V-optimal policies under the hypotheses of Section 2. This will be achieved by leveraging the connection between V-optimality and a neighboring optimality concept of which return functions, rather than return vectors, are the core ingredient.

Definition 2 (F-optimality)**.** *A policy π^F is F-optimal if and only if $u_1^{\pi^F} \in e\big(\bigcup_{\pi\in\Pi}\{u_1^\pi\}\big)$,*

where "F" stands for "function", and where it is implicit that $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$ is ordered by $\succeq$ as defined in (8).

In brief, a policy π^* is V-optimal if for each state s there exists no other policy $\pi_s \neq \pi^*$ with $u_1^{\pi_s}(s) \geq u_1^{\pi^*}(s)$, and is F-optimal if there is no other π such that $u_1^\pi \succeq u_1^{\pi^*}$.

Example 1. *The distinction is illustrated by the following situation. Suppose these policies were available in a two-state model with $m = 2$: a policy π_1 yielding $u_1^{\pi_1}(s_1) = (3, 1)$ and $u_1^{\pi_1}(s_2) = (5, -2)$, and a policy π_2 yielding $u_1^{\pi_2}(s_1) = (2, 1)$ and $u_1^{\pi_2}(s_2) = (\frac{1}{2}, 0)$. Then $u_1^{\pi_1}$ and $u_1^{\pi_2}$ are incomparable with respect to $\succeq$; $u_1^{\pi_1}(s_2)$ and $u_1^{\pi_2}(s_2)$ are incomparable with respect to $\geq$; and $u_1^{\pi_1}(s_1) > u_1^{\pi_2}(s_1)$. By definition, π_2 is not V-optimal, for $u_1^{\pi_1}(s_1) > u_1^{\pi_2}(s_1)$ implies the existence of a state s (s_1 here) for which there is a policy $\pi_s \neq \pi_2$ (π_1 here) with $u_1^{\pi_s}(s) \geq u_1^{\pi_2}(s)$. It may still be F-optimal, however, as we have $u_1^{\pi_1}(s_2) \not\geq u_1^{\pi_2}(s_2)$ and therefore $u_1^{\pi_1} \not\succeq u_1^{\pi_2}$. If some third policy π_3 satisfied $u_1^{\pi_3}(s_1) \geq u_1^{\pi_2}(s_1)$ and $u_1^{\pi_3}(s_2) \geq u_1^{\pi_2}(s_2)$, then π_2 would not be F-optimal.*

In the succeeding development, we will find it convenient to focus on the latter concept for four key reasons, all of which will be demonstrated in due course: (1) we are able to guarantee the existence of F-optimal policies; (2) the problem of finding F-optimal policies is susceptible to dynamic programming; (3) F-optimal policies satisfy the Principle of Optimality (Bellman, 1954); and (4) a V-optimal policy *must* be F-optimal, that is, given $\pi \in \Pi$, efficiency of u_1^π in $F(S, \mathbb{R}^m)$ is a necessary condition for efficiency of $u_1^\pi(s)$ in $\mathbb{R}^m$ for all $s \in S$. These observations have important practical implications. First, the fact that the Principle of Optimality holds means that all F-optimal – and therefore all V-optimal – policies will be found through dynamic programming. Second, if we can determine which policies are not F-optimal, we will immediately recognize those that are not V-optimal. Third, if $e(\bigcup_{\pi\in\Pi}\{u_1^\pi\})$ is finite and can be computed in a finite number of steps, it will be possible to obtain the set of V-optimal policies, and thereby solve both optimization problems simultaneously. This last point will be illustrated in Section 5.

Proposition 1. *Let $\pi^* \in \Pi$. If π^* is V-optimal, then it is F-optimal.*

Proof. Suppose that $\pi^* \in \Pi$ is V-optimal. Let $\pi' \in \Pi$ be a policy such that $u_1^{\pi'} \succeq u_1^{\pi^*}$. Then for any $s \in S$, $u_1^{\pi'}(s) \geq u_1^{\pi^*}(s)$. Therefore, since π^* is V-optimal, it follows that $u_1^{\pi'}(s) = u_1^{\pi^*}(s)$ for any $s \in S$. Thus, $u_1^{\pi'} = u_1^{\pi^*}$. This shows that $u_1^{\pi^*} \in e(\bigcup_\pi \{u_1^\pi\})$, and hence that π^* is F-optimal. □

Lemma 1 formalizes a useful intuition about return functions that will be invoked repeatedly throughout this section.

Lemma 1. *Let $\pi = (d_1, ..., d_{N-1})$, $\pi' = (d'_1, ..., d'_{N-1}) \in \Pi$, $d_t \in D$ and $t = 1, ..., N-2$. Suppose $u_{t+1}^{\pi} \succeq u_{t+1}^{\pi'}$. Then for any two policies π_1 and π_2 such that $\bar{\pi}_1(t) = (d_t, d_{t+1}, ..., d_{N-1})$ and $\bar{\pi}_2(t) = (d_t, d'_{t+1}, ..., d'_{N-1})$, $u_t^{\pi_1} \succeq u_t^{\pi_2}$.*

Proof. Suppose $u_{t+1}^{\pi} \succeq u_{t+1}^{\pi'}$, and let $s \in S$. Let $\pi_1, \pi_2 \in \Pi$ be policies such that $\bar{\pi}_1(t) = (d_t, d_{t+1}, ..., d_{N-1})$ and $\bar{\pi}_2(t) = (d_t, d'_{t+1}, ..., d'_{N-1})$. For all $j \in S$, $u_{t+1}^{\pi_1}(j) = u_{t+1}^{\pi}(j) \geq u_{t+1}^{\pi'}(j) = u_{t+1}^{\pi_2}(j)$, hence $\sum_{j \in S} p(j|s, d_t(s)) u_{t+1}^{\pi_1}(j) \geq \sum_{j \in S} p(j|s, d_t(s)) u_{t+1}^{\pi_2}(j)$ due to the nonnegativity of probabilities. Thus,

$$R_t(s, d_t(s)) + \sum_{j \in S} p(j|s, d_t(s)) u_{t+1}^{\pi_1}(j) \geq R_t(s, d_t(s)) + \sum_{j \in S} p(j|s, d_t(s)) u_{t+1}^{\pi_2}(j).$$

This establishes $u_t^{\pi_1}(s) \geq u_t^{\pi_2}(s)$ for each $s \in S$. Ergo, $u_t^{\pi_1} \succeq u_t^{\pi_2}$. □

Example 2. *Consider a vector-valued Markov decision process with $S = \{1, 2\}$, $A_1 = \{a, b\}$, and $A_2 = \{a\}$. Suppose that at a decision epoch t we had $p_t(1|1, a) = .75$; $p_t(2|1, a) = .25$; $p_t(1|1, b) = p_t(2|1, b) = .5$; $p_t(1|2, a) = 1$; $p_t(2|2, a) = 0$; $R_t(1, a) = (1, 0)$; $R_t(2, a) = (0, 0)$; and $R_t(1, b) = (0, 1)$.*

For the purposes of this example, let us assume that there exist policies π and π' with returns $u_{t+1}^{\pi}(1) = (0, 0)$, $u_{t+1}^{\pi}(2) = (-2, 2)$, $u_{t+1}^{\pi'}(1) = (-0.5, 0)$, $u_{t+1}^{\pi'}(2) = (-6, 1)$. Clearly, $u_{t+1}^{\pi} \succeq u_{t+1}^{\pi'}$.

Now let $d_t \in D$ denote the decision rule that chooses b in state 1, i.e., $d_t(1) = b$ and $d_t(2) = a$. Choose π_1 to be any policy that selects d_t at time t then pursues π from time $t + 1$ onward. Similarly, let π_2 select d_t at time t then pursue π' at all future epochs. In our notation, this translates to $\bar{\pi}_1(t) = (d_t, \pi)$ and $\bar{\pi}_2(t) = (d_t, \pi')$. Through simple calculations, we will demonstrate the assertion in Lemma 1 that $u_t^{\pi_1} \succeq u_t^{\pi_2}$. From Equation (3) we have that

$$\begin{aligned} u_t^{\pi_1}(1) &= R_t(1, d_t(1)) + \sum_{j \in S} p_t(j|1, d_t(1)) u_{t+1}^{\pi}(j) \\ &= (0, 1) + 0.5 \cdot (0, 0) + 0.5 \cdot (-2, 2) \\ &= (-1, 2), \end{aligned}$$

and

$$\begin{aligned} u_t^{\pi_2}(1) &= R_t(1, d_t(1)) + \sum_{j \in S} p_t(j|1, d_t(1)) u_{t+1}^{\pi'}(j) \\ &= (0,1) + 0.5 \cdot (-0.5, 0) + 0.5 \cdot (-6, 1) \\ &= (-3.25, 1.5). \end{aligned}$$

Thus, $u_t^{\pi_1}(1) \geq u_t^{\pi_2}(1)$*. The reader can easily replicate this method of calculation to verify that* $u_t^{\pi_1}(2) = (0,0)$ *and* $u_t^{\pi_2}(2) = (-0.5, 0)$*. This means that* $u_t^{\pi_1}(2) \geq u_t^{\pi_2}(2)$*, hence* $u_t^{\pi_1} \succeq u_t^{\pi_2}$*.*

Fundamental to the proof of Lemma 1 is the fact that for any $\pi = (d_1, ..., d_{N-1}) \in \Pi$, $s \in S$ and $t = 1, ..., N-1$, $u_t^{\pi}(s) = R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s)) u_{t+1}^{\pi}(j)$. That is, policy returns are separable and additive. The scope of the lemma, however, covers a broader category of separable returns. Following Morin (1982), we can make this generalization: any vector-valued Markov decision process such that $u_t^{\pi}(s) = R_t(s, d_t(s)) \circ \sum_{j \in S} p_t(j|s, d_t(s)) u_{t+1}^{\pi}(j)$, where $\circ$ is an isotonic symmetric binary operator, i.e., a symmetric binary operator that preserves inequalities (with respect to $\geq$), satisfies the lemma. To prove this generalization, we may proceed in exactly the same fashion as above, concluding from the isotonicity of $\circ$ that

$$R_t(s, d(s)) \circ \sum_{j \in S} p(j|s, d(s)) u_{t+1}^{\pi_1}(j) \geq R_t(s, d(s)) \circ \sum_{j \in S} p(j|s, d(s)) u_{t+1}^{\pi_2}(j)$$

for all $s \in S$, and therefore that $u_t^{\pi_1} \succeq u_t^{\pi_2}$. Incidentally, Morin (1982) points out that a strictly isotonic associative $\circ$, of which addition in $\mathbb{R}^m$ and componentwise multiplication in $(0, \infty)^m$ would be examples, ensures the validity of the Bellman equations in Markov decision processes. However, Mifrani (2025) has recently shown that this is not true for all vector-valued Markov decision processes with regard to the vector extension of those equations.

It will later prove desirable, especially for the purpose of justifying optimality equations, to have a property that enables us to assert that each inefficient point in $\bigcup_{\pi \in \Pi} \{u_t^{\pi}\}$ is dominated by an efficient one. Notice that this is not entailed by the definition of efficiency, because in general, all we can say about an inefficient point is that it is dominated by another point, which may or may not be efficient. Berge (1985) calls "absorbent" a partially ordered set $S \subseteq (X, \geq)$ such that for every $x \in X$, there exists $s \in S$ satisfying $s \geq x$. We wish then an absorbent $e(\bigcup_{\pi \in \Pi} \{u_t^{\pi}\})$ for all $t = 1, ..., N$, so that in addition to the aforementioned property, we may conclude that $e(\bigcup_{\pi \in \Pi} \{u_1^{\pi}\}) \neq \emptyset$, and therefore that F-optimal policies exist. The following lemma from Henig (1985) implies that a nonempty partially ordered set is absorbent if it meets the conditions of Zorn's lemma (Zorn, 1935).

Lemma 2. *(Henig, 1985) Let* $(U, \geq)$ *be a nonempty partially ordered set, and* K *a nonempty subset of* U*. Suppose that for every* $u \in U$ *and every* $v \in K$*,* $u \geq v$ *implies* $u \in K$*. Suppose further that every totally ordered subset (chain) of* U *has an upper bound in* U*. Then* $e(U) \cap K \neq \emptyset$*.*

For fixed $u \in \bigcup_{\pi\in\Pi}\{u_t^\pi\}$ and $t = 1, ..., N$, let $K(u)$ denote the set $\{v \in \bigcup_\pi\{u_t^\pi\} : v \succeq u\}$. We claim that there is an efficient $v \in e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$ such that $v \succeq u$. Our proof rests on Lemma 2. First of all, $K(u)$ is nonempty, as $u \succeq u$. Moreover, as will be shown below,

(1) for every v in $K(u)$ and every $v' \in \bigcup_\pi\{u_t^\pi\}$, $v' \succeq v$ implies $v \in K(u)$, and;
(2) every chain of $\bigcup_{\pi\in\Pi}\{u_t^\pi\}$ is bounded above in $\bigcup_{\pi\in\Pi}\{u_t^\pi\}$.

The proof of point (2) involves studying the convergence of certain sequences in D and in $\Pi = D^{N-1}$. For convergence to be meaningful on either set, a topology must be introduced. The most widely assumed topology in the analysis of Markov decision processes is that of uniform (or sup-norm) convergence. But because uniform convergence implies pointwise convergence, and because our proof does not use properties of the former which are not true of the latter, it suffices to endow D with the topology of pointwise convergence and, by extension, Π with the associated product topology.

Theorem 1. *Equip D with the topology of pointwise convergence and $\Pi = D^{N-1}$ with the product topology. Then points (1) and (2) as enunciated above are true.*

Proof. We divide the proof into two parts.

(1) Let $v \in K(u)$ and $v' \in \bigcup_\pi\{u_t^\pi\}$. If $v' \succeq v$, then, since $v \succeq u$ and $\succeq$ is transitive, we have that $v' \succeq u$, hence $v' \in K(u)$.

(2) Notice first that Π, being the product of compact sets, is compact. For all $\pi \in \Pi$, let $f_t(\pi) = u_t^\pi$. According to (Birkhoff, 1940, Theorem 16), it suffices to show that f_t, viewed as a mapping from Π to $F(S, \mathbb{R}^m)$, satisfies the following property: whenever $e \in F(S, \mathbb{R}^m)$ and for every sequence $(\pi_n)_n$ with values in Π, $\pi_n \to \pi^\circ$ and $f_t(\pi_n) \geq e$ for all n imply $f_t(\pi^\circ) \geq e$.

Accordingly, let $e \in F(S, \mathbb{R}^m)$ and $(\pi_n)_n$ a sequence of policies converging to a $\pi^0 \in \Pi$, with $f_t(\pi_n) \succeq e$ for all n. Let $s \in S$. From (5), we have that

$$f_t(\pi_n)(s) = R_t(s, d_t^{\pi_n}(s)) + \sum_{i=t}^{N-2}\sum_{j\in S}\left(\prod_{k=t}^{i} P^{d_k^{\pi_n}}\right)_{s,j} R_{i+1}(j, d_{i+1}^{\pi_n}(j)) + \sum_{j\in S}\left(\prod_{k=t}^{N-1} P^{d_k^{\pi_n}}\right)_{s,j} R_N(j) \geq e(s)$$

for all n. Now, in view of D's topology, we have that for all $i = 1, ..., N$, $d_i^{\pi_n}(s) \to d_i^{\pi^0}(s)$ in A. This, together with the continuity of the transition probabilities and of each reward component on A, yields $f_t(\pi^n)(s)_p \to f_t(\pi^0)(s)_p$ in $\mathbb{R}$ and hence $f_t(\pi^0)(s)_p \geq e(s)_p$ for all $p = 1, ..., m$. Thus, by definition, $f_t(\pi^0)(s) \geq e(s)$. Since s was chosen arbitrarily, it follows that $f_t(\pi^0) \succeq e$, again by definition of $\succeq$. By virtue of this and the compactness of Π, it follows from Theorem 1 that every chain in $f_t(P) = \bigcup_\pi\{u_t^\pi\}$ has an upper bound in $\bigcup_\pi\{u_t^\pi\}$. □

Theorem 1 relies on the fact that Π is compact, which in turn relies on the fact that D is compact. Considering that many Markov decision process applications use a finite A (Borrero & Akhavan-Tabatabaei, 2013; Goedhart et al., 2023; Mason et al., 2014; Ramirez-Nafarrate et al., 2014; Schlosser & Gönsch, 2023; Wang, Demeulemeester, Vansteenkiste, & Rademakers, 2024; White, 1993), and therefore a compact D, this is not as restrictive an assumption as it may seem at first glance.

Corollary 1. *For all $t = 1, ..., N$, if $u \in \bigcup_{\pi\in\Pi}\{u_t^\pi\}$, there is an efficient return function $v \in e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$ such that $v \succeq u$.*

Example 3. *We illustrate part (2) of Theorem 1 in tandem with Corollary 1. In a certain model with state space $S = \{1, 2\}$ and $m = 2$, the return functions generated by all policies from time $t = 1$ onward were found to be given by*

$$\bigcup_{\pi\in\Pi}\{u_1^\pi\} = \{u^1, u^2, u^3, u^4, u^5, u^6, u^7, u^8\} \subset F(S, \mathbb{R}^2),$$

where, for example, $u^1(1) = (1.64, 12.6)$; $u^1(2) = (2.44, 8.56)$; $u^4(1) = (3.44, -2.44)$; $u^4(2) = (1.62, -7.62)$; $u^8(1) = (-4.62, -6.37)$; $u^8(2) = (-3.75, -10.25)$. The exact values are immaterial to the purposes of this example; what is important here are the relations among the points in the above set. We can see that $u^1 \succeq u^8$, $u^4 \succeq u^8$, and that no comparison is possible between u^1 and u^4. The full network of relations is summarized in the diagram of Figure 1. For example, the diagram indicates that $u^1 \succeq u^3$, but also that $u^1 \succeq u^7$, as $u^3 \succeq u^7$ and $\succeq$ is transitive. Such drawings are known in set theory as Hasse diagrams, and are particularly useful for determining chains and antichains (subsets of which no distinct points are comparable) in a partially ordered set.

We shall verify the assertion in Theorem 1(2) that every totally ordered subset of $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$ is upper bounded in $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$ relative to $\succeq$. According to Figure 1, the totally ordered subsets in this example comprise:

(1) eight singletons $\{u^i\}$, $i = 1, ..., 8$, which are bounded above by virtue of $\succeq$ being reflexive;
(2) fifteen two-point sets including $\{u^1, u^5\}$, $\{u^4, u^8\}$, $\{u^1, u^8\}$, and $\{u^2, u^8\}$, all of which are bounded above by the element from which the arrow (or arrows) originates (originate);
(3) and six three-point sets $\{u^1, u^3, u^8\}$, $\{u^1, u^3, u^7\}$, $\{u^2, u^4, u^8\}$, $\{u^2, u^3, u^8\}$, $\{u^2, u^4, u^7\}$, and $\{u^2, u^3, u^7\}$, all of which are bounded above either by u^1 or by u^2.

On the same diagram we can observe that $e(\bigcup_{\pi\in\Pi}\{u_1^\pi\}) = \{u^1, u^2\}$, since u^1 and u^2 are the only points towards which no arrows are directed. If the claim in Corollary 1 is correct, then each u^i, $i = 1, ..., 8$, should satisfy $u^1 \succeq u^i$, or $u^2 \succeq u^i$, or both. A simple inspection of Figure 1 reveals this to be indeed the case.

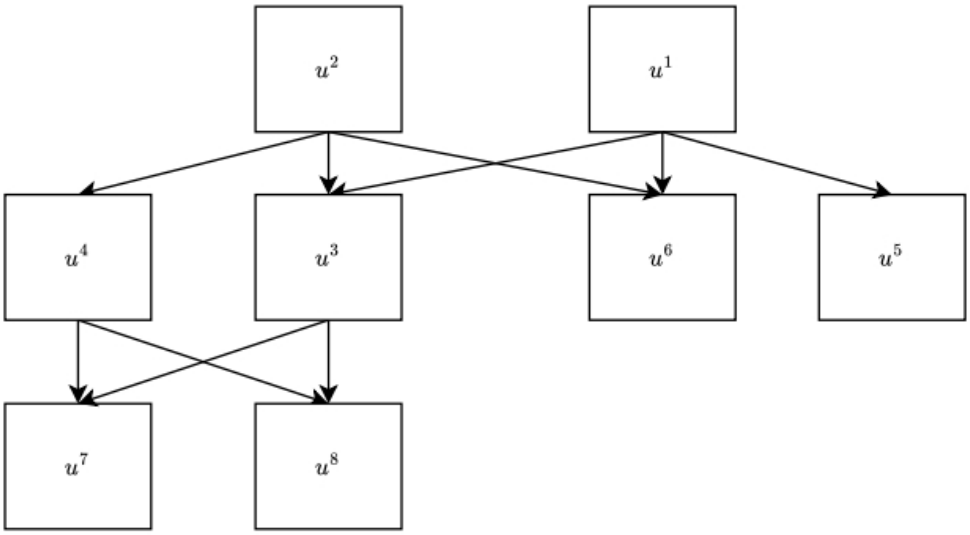

FIGURE 1. A Hasse diagram of the set $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$ in Example 3, ordered by $\succeq$. An outward-pointing arrow from u^i to u^j indicates that $u^i \succeq u^j$.

A byproduct of Corollary 1 is that the sets $e(\bigcup_\pi\{u_1^\pi\}), ..., e(\bigcup_\pi\{u_N^\pi\})$ are nonempty. In particular, there is at least one F-optimal policy.

Theorem 2. *Let Π_F^* be the set of all F-optimal policies. Then $\Pi_F^* \neq \emptyset$.*

This existence result can also be obtained in a different way. Puterman (2014) has shown that a scalar-valued Markov decision process satisfying the assumptions of this work – namely, a finite S, a compact A, and rewards and transition probabilities which are continuous on A – has at least one optimal policy, that is, a π^* such that $u_1^{\pi^*}(s) = \max_{\pi\in\Pi} u_1^\pi(s)$ for all states s. Take now any m positive scalars $\lambda_1, ..., \lambda_m$, and write $\lambda = (\lambda_1, ..., \lambda_m)$. Recalling that $R_t(s,a) = (r_t(s,a)_1, ..., r_t(s,a)_m)$, Puterman's result implies that the Markov decision process with rewards $\langle \lambda,\ R_t(s,a)\rangle \in \mathbb{R}$ has a policy π^* such that $\langle \lambda,\ u_1^{\pi^*}(s)\rangle \geq \langle \lambda,\ u_1^\pi(s)\rangle$ for all $\pi \in \Pi$ and $s \in S$. It is a straightforward exercise to prove that such a π^* must be V-optimal, and therefore F-optimal, for the vector-valued process.

We now propose dynamic programming equations that yield an algorithm for enumerating the set of F-optimal policies. Our proof of the equations' validity will employ the following observation.

Lemma 3. *For all $t = 1, ..., N-1$,*

$$\bigcup_{d_t\in D}\left\{S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s))v(j) : v \in \bigcup_\pi \{u_{t+1}^\pi\}\right\} = \bigcup_\pi \{u_t^\pi\}$$

Proof. Let $t = 1, ..., N-1$. Let $w \in F(S, \mathbb{R}^m)$ such that

$$\forall s \in S,\ w(s) = R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s))v(j)$$

for some $d_t \in D$ and $v \in \bigcup_\pi \{u^\pi_{t+1}\}$. We may write $v = u^\pi_{t+1}$ for some $\pi = (d_1, ..., d_{N-1}) \in \Pi$. Let $\pi' \in \Pi$ be any policy such that $\bar{\pi}'(t) = (d_t, d_{t+1}, ..., d_{N-1})$. Then for all $s \in S$, $w(s) = u^{\pi'}_t(s)$. Thus, $w = u^{\pi'}_t$, whence

$$\bigcup_{d_t \in D} \left\{ S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s))v(j) : v \in \bigcup_\pi \{u^\pi_{t+1}\} \right\} \subseteq \bigcup_\pi \{u^\pi_t\}.$$

The converse inclusion can readily be obtained from (3); as a result, the lemma is established. □

Recall that D, the set of all decision rules, is the set of all mappings from S into A. We may now state the relation between $e(\bigcup_{\pi \in \Pi} \{u^\pi_t\})$ and $e(\bigcup_{\pi \in \Pi} \{u^\pi_{t+1}\})$ for all $t = 1, ..., N-1$. From it we will deduce a dynamic programming algorithm that finds all F-optimal policies by leveraging the structure of the equations.

Theorem 3. *For all $t = 1, ..., N$, $e(\bigcup_{\pi \in \Pi} \{u^\pi_t\})$ is the unique solution U_t to either of the following equations:*

$$U_t = e\left(\bigcup_{d_t \in D} \left\{ S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s))v(j) : v \in U_{t+1} \right\} \right);\ t < N \tag{9}$$

$$U_t = \{R_N\};\ t = N \tag{10}$$

Proof. For greater legibility we set, for all $t = 1, ..., N$,

$$G_t = \bigcup_{d_t \in D} \left\{ S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s))v(j) : v \in U_{t+1} \right\}.$$

We proceed by induction on t. For any policy $\pi \in \Pi$, $u^\pi_N = R_N$. Thus, $\bigcup_\pi \{u^\pi_N\}$ is the singleton $\{R_N\}$, and $e(\bigcup_\pi \{u^\pi_t\}) = \{R_N\} = U_N$. The property then holds for $t = N$. Assume it is true for $t+1$, for some $t < N$. Let $u \in e(\bigcup_\pi \{u^\pi_t\})$. There exists a $\pi = (d_1, ..., d_{N-1}) \in \Pi$ such that $u = u^\pi_t$. By Corollary 1, there exists a $v \in e(\bigcup_\pi \{u^\pi_{t+1}\})$ such that $v \succeq u^\pi_{t+1}$. Let $w \in F(S, \mathbb{R}^m)$ be the function such that $w(s) = R_t(s, d_t(s)) + \sum_{j \in S} p_t(j|s, d_t(s))v(j)$ for all $s \in S$. Then $w \in \bigcup_\pi \{u^\pi_t\}$ by Lemma 3, and $w \succeq u$ by Lemma 1. But u is efficient in $\bigcup_\pi \{u^\pi_t\}$; therefore, $u = w$. Appealing to the induction hypothesis for v then to Lemma 3 proves $u \in U_t$. This establishes $e(\bigcup_\pi \{u^\pi_t\}) \subseteq U_t$.

To show the converse inclusion, let $v \in U_t$. We have $v \in \bigcup_\pi \{u^\pi_t\}$. Consider now some $u \in \bigcup_\pi \{u^\pi_t\}$ such that $u \succeq v$. We shall show that we necessarily have $v = u$. Applying Corollary 1 then Lemma 1, there is a $u' \in e(\bigcup_\pi \{u^\pi_{t+1}\})$ such that $w := s \mapsto R_t(s, d_t(s)) +$

$\sum_{j\in S} p_t(j|s,d_t(s))u'(j)) \succeq u$. Then $w \succeq v$. By our induction hypothesis, $u' \in U_{t+1}$, and thus $w \in G_t$. But v being efficient in G_t, we must have $v = w$ and therefore $v \succeq u$. Consequently, $v = u$. The requisite inclusion then follows, and the property holds for all $t = 1, ..., N$. □

Although Equations (9) and (10) bear a striking resemblance to the White equations (see Section 1), the two sets of equations differ in crucial respects. In the first place, the unknowns in (9) and (10) are subsets of $F(S, \mathbb{R}^m)$, whereas in White's case they are subsets of $\mathbb{R}^m$. White's equations involve a total of $N \cdot |S|$ unknowns; ours involve N unknowns. In the second place, the solution of (9) or (10) at an epoch t must be a subset of $\bigcup_{\pi\in\Pi}\{u_t^\pi\}$ by virtue of, *inter alia*, Lemma 3, the key argument in the proof above. In contrast, we cannot guarantee in general that White's solution sets are contained in the $\bigcup_{\pi\in\Pi}\{u_t^\pi(s)\}$'s, $s \in S$, except in conditions like those expatiated in (Mifrani, 2025). Incidentally, two of the conditions – namely, that the dynamics of the model be deterministic, and that the decision-making horizon be short of three epochs – are special cases of this paper's hypotheses, and thus ensure the validity of both White's and our equations.

It is clear that by construction each member of U_t, $t = 1, ..., N-1$, is characterized by some sequence of decision rules $(d_t, ..., d_{N-1})$. Thus, if $\mathcal{L}_t$ is the mapping from D^{N-t} into U_t defined by $\mathcal{L}_t(d_t, ..., d_{N-1}) = u_t^\pi$, where π is any policy with $\bar{\pi}(t) = (d_t, ..., d_{N-1})$, then $\mathcal{L}_t$ is onto. This mapping need not be one-to-one, as distinct policies may well have the same return function.

Theorem 3 and the remarks of the previous paragraph naturally give rise to the following algorithm for calculating F-optimal policies, an algorithm that also solves Equations (9) subject to (10).

Algorithm 1. *Solution of Equations (9) subject to (10) and calculation of* Π_F^*.

(1) Set $t = N - 1$ *and*

$$U_{N-1} = e\left(\bigcup_{d_t\in D}\left\{S \ni s \mapsto R_{N-1}(s, d_t(s)) + \sum_{j\in S} p_{N-1}(j|s, d_t(s))R_N(j)\right\}\right) \tag{11}$$

$$P_{N-1}^* = \{d_t \in D : S \ni s \mapsto R_{N-1}(s, d_t(s)) + \sum_{j\in S} p_{N-1}(j|s, d_t(s))R_N(j) \in U_{N-1}\} \tag{12}$$

(2) Substitute $t - 1$ *for* t *and set*

$$U_t = e\left(\bigcup_{d_t\in D}\left\{S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s))v(j) : v \in U_{t+1}\right\}\right) \tag{13}$$

$$P_t^* = \{(d_t, d_{t+1}, ..., d_{N-1}) \in D^{N-t} : (d_{t+1}, ..., d_{N-1}) \in P_{t+1}^* \\ \text{and } S \ni s \mapsto R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s))u_{t+1}^\pi(j) \in U_t\} \tag{14}$$

where $\pi \in \Pi$ is any policy such that $\bar{\pi}(t+1) = (d_{t+1}, ..., d_{N-1})$.

(3) If $t = 1$, stop. Otherwise, go to (2).

Proposition 2. *The sets U_t returned by Algorithm 1 are the solutions to Equations (9) and (10), and therefore satisfy $U_t = e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$ for each $t = 1, ..., N$.*

At termination, P_1^* contains every policy satisfying $u_t^\pi \in U_t$ for all $t = 1, ..., N$. By Theorem 3, such policies are F-optimal, but one might ask whether these include all or only a subset of F-optimal policies. It turns out from Lemma 1 that if π is F-optimal, then the subpolicy $\bar{\pi}(t')$, $1 < t' \leq N$, is also F-optimal with respect to the portion of the decision-making horizon that begins at t', i.e., $u_{t'}^\pi \in U_t'$. Phrased loosely, an F-optimal policy π is "F-optimal" at every stage of decision-making: not only does it achieve an efficient return function u_1^π over the N epochs, but it also achieves an efficient return function u_t^π from any epoch t onward. Specifically, if for each epoch t we let E_t denote the efficient set $e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$, we have the sequence of implications

$$u_1^\pi \in E_1 \implies u_2^\pi \in E_2 \implies \cdots \implies u_{N-1}^\pi \in E_{N-1},$$

or, viewed from a logically equivalent angle,

$$u_{N-1}^\pi \notin E_{N-1} \implies u_{N-2}^\pi \notin E_{N-2} \implies \cdots \implies u_1^\pi \notin E_1.$$

A formal statement and proof of this structural property of the model follow.

Proposition 3. *For any $\pi \in \Pi$, $\pi \in \Pi_F^*$ implies $u_t^\pi \in e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$ for all $t = 1, ..., N$.*

Proof. Let $\pi = (d_1, ..., d_{N-1}) \in \Pi_F^*$. We proceed by induction on t. By definition of Π_F^*, $u_1^\pi \in e(\bigcup_{\pi\in\Pi}\{u_1^\pi\})$. Let $t = 2, ..., N-1$ such that $u_t^\pi \in e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$. We will show that $u_{t+1}^\pi \in e(\bigcup_{\pi\in\Pi}\{u_{t+1}^\pi\})$. Assume to the contrary that $u_{t+1}^\pi \notin e(\bigcup_{\pi\in\Pi}\{u_{t+1}^\pi\})$. There then exists a $\pi' \in \Pi$ such that $u_{t+1}^{\pi'} \succ u_{t+1}^\pi$. Therefore, $s \mapsto R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s)) u_{t+1}^{\pi'}(j) \succ u_t^\pi$ by Lemma 1. However, Lemma 3 tells us that $s \mapsto R_t(s, d_t(s)) + \sum_{j\in S} p_t(j|s, d_t(s)) u_{t+1}^{\pi'}(j) \in \bigcup_{\pi\in\Pi}\{u_t^\pi\}$, which contradicts the fact that $u_t^\pi \in e(\bigcup_{\pi\in\Pi}\{u_t^\pi\})$ and completes the proof. □

The argument of this proof is very straightforward. Suppose that a policy $\pi = (d_1, \ldots, d_{N-1})$ does *not* achieve an efficient return function over the period $t+1$, ..., N. This means we can find a policy π' whose return function for the same period is better, i.e., $u_{t+1}^{\pi'} \succ u_{t+1}^\pi$. Obviously, because decisions taken prior to epoch $t+1$ cannot influence how well a policy does between $t+1$ and N, we may write, with a slight abuse of notation, $u_{t+1}^\pi = u_{t+1}^{\bar{\pi}(t+1)}$ and $u_{t+1}^{\pi'} = u_{t+1}^{\bar{\pi}'(t+1)}$, so that $u_{t+1}^{\bar{\pi}'(t+1)} \succ u_{t+1}^{\bar{\pi}(t+1)}$. Now suppose we extended both $\bar{\pi}(t+1)$ and $\bar{\pi}'(t+1)$ by the decision rule d_t, giving rise to two (partial) policies $(d_t, \bar{\pi}(t+1))$ and $(d_t, \bar{\pi}'(t+1))$. Since both policies employ the same decision rule at time t, and since $u_{t+1}^{\bar{\pi}'(t+1)} \succ u_{t+1}^{\bar{\pi}(t+1)}$, we deduce from Equation (3) that $(d_t, \bar{\pi}'(t+1))$ has the superior return function, which is to say, $u_t^{(d_t, \bar{\pi}'(t+1))} \succ u_t^{(d_t, \bar{\pi}(t+1))}$.

But d_t being the t-th decision rule in π, we have $(d_t, \bar{\pi}(t+1)) = \bar{\pi}(t)$ by construction, so that $u_t^{(d_t, \bar{\pi}(t+1))} = u_t^{\pi}$. Thus, π does not achieve an efficient return function over the period t, ..., N. What we have just shown, in summary, is that if π is not "F-optimal" from some epoch $t+1$ onward, it is not "F-optimal" from epoch t onward either, no matter what actions are prescribed at epoch t (for a practical illustration, see "Accompanying example for Proposition 3" of the supplemental material).

As a result of Proposition 3, we are assured that Algorithm 1 will discover all F-optimal policies, since the policies that satisfy $u_1^{\pi} \in U_1 = e(\bigcup_{\pi \in \Pi} \{u_1^{\pi}\})$ are precisely those that satisfy $u_t^{\pi} \in U_t = e(\bigcup_{\pi \in \Pi} \{u_t^{\pi}\})$ for all $t = 1, ..., N$.

Corollary 2. *Algorithm 1 is guaranteed to locate all F-optimal policies at termination, i.e., $P_1^* = \Pi_F^*$, where P_t^* is defined by Equation (14) for each $t = 1, ..., N-1$.*

Recall that part of the motivation for introducing F-optimality was that it is a necessary condition for V-optimality. Hence, if Π_V^* denotes the set of all V-optimal policies, then $\Pi_V^* \subseteq \Pi_F^*$. As a preliminary to an algorithm for the determination of Π_V^*, we show that given an initial state $s \in S$, each efficient policy return vector accrued over the decision-making horizon is attained by at least an F-optimal policy. For any state s, the efficient elements in $\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}$ are therefore a subset of $\bigcup_{\pi \in \Pi_F^*} \{u_1^{\pi}(s)\}$. This has the practical effect of reducing the task of "maximizing" return vectors over the whole of Π to the less onerous task of "maximizing" return vectors over Π_F^*.

Theorem 4. *Let $s \in S$. Then* $e\left(\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}\right) = e\left(\bigcup_{\pi \in \Pi_F^*} \{u_1^{\pi}(s)\}\right)$.

Proof. Decompose $\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}$ as follows:

$$\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\} = F_1 \cup F_2,$$

where

$$F_1 = \bigcup_{\pi \in \Pi \setminus \Pi_F^*} \{u_1^{\pi}(s)\}$$

and

$$F_2 = \bigcup_{\pi \in \Pi_F^*} \{u_1^{\pi}(s)\}.$$

We shall first prove that $e\Big(\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}\Big) \subseteq e(F_2)$. Pick a policy $\pi^* \in \Pi$ such that $u_1^{\pi^*}(s) \in e\Big(\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}\Big)$. It shall be established that $u_1^{\pi^*}(s) \in F_2$, which, given that F_2 is a subset of $\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}$, implies that $u_1^{\pi^*}(s)$ is efficient in F_2. Suppose, for the sake of contradiction, that $u_1^{\pi^*}(s) \notin F_2$. Then $\pi^* \notin \Pi_F^*$, and there is therefore, applying Theorem 1, a $\pi' \in \Pi_F^*$ with $u_1^{\pi'} \succ u_1^{\pi^*}$. Thus $u_1^{\pi'}(s) \geq u_1^{\pi^*}(s)$. Since $u_1^{\pi^*}(s)$ is efficient in $\bigcup_{\pi \in \Pi} \{u_1^{\pi}(s)\}$, it follows that

$u_1^{\pi^*}(s) = u_1^{\pi'}(s) \in F_2$: a contradiction. Therefore, $u_1^{\pi^*}(s) \in F_2$, hence $u_1^{\pi^*}(s) \in e(F_2)$. This concludes the demonstration of the fact that $e\Big(\bigcup_{\pi\in\Pi}\{u_1^{\pi}(s)\}\Big) \subseteq e(F_2)$.

Consider now an F-optimal policy $\pi^* \in \Pi_F^*$ satisfying $u_1^{\pi^*}(s) \in e(F_2)$. To show that $u_1^{\pi^*}(s)$ is also efficient in $\bigcup_{\pi\in\Pi}\{u_1^{\pi}(s)\}$, let $\pi \in \Pi$ be some policy such that $u_1^{\pi}(s) \geq u_1^{\pi^*}(s)$. Either $\pi \in \Pi_F^*$ or $\pi \notin \Pi_F^*$. If $\pi \in \Pi_F^*$, then $u_1^{\pi}(s) \in F_2$ and consequently $u_1^{\pi}(s) = u_1^{\pi^*}(s)$. Otherwise, invoking Theorem 1 again, there exists a $\pi' \in \Pi_F^*$ such that $u_1^{\pi'}(s) \geq u_1^{\pi}(s)$. Due to the transitivity of $\geq$ and given that $u_1^{\pi'}(s) \in F_2$, this implies $u_1^{\pi'}(s) = u_1^{\pi^*}(s)$. It follows that $u_1^{\pi^*}(s) \geq u_1^{\pi}(s)$, which when combined with the fact that $u_1^{\pi}(s) \geq u_1^{\pi^*}(s)$ yields $u_1^{\pi}(s) = u_1^{\pi^*}(s)$. In both cases, we have that $u_1^{\pi}(s) = u_1^{\pi^*}(s)$. This proves the efficiency of $u_1^{\pi^*}(s)$ in $\bigcup_{\pi\in\Pi}\{u_1^{\pi}(s)\}$ and concludes the proof of the theorem. □

The appendix carries an example illustrating this theorem (see "Accompanying example for Theorem 4" of the supplemental material).

A characterization of V-optimal policies follows at once from Theorem 4, namely that a policy $\pi^* \in \Pi$ is V-optimal if and only if $u_1^{\pi^*}(s) \in e\left(\bigcup_{\pi\in\Pi_F^*}\{u_1^{\pi}(s)\}\right)$ for each state $s \in S$. We therefore have an alternate – and, as will now be shown, useful – representation of Π_V^*:

$$\Pi_V^* = \left\{\pi \in \Pi_F^* : \forall s \in S, u_1^{\pi}(s) \in e\left(\bigcup_{\pi\in\Pi_F^*}\{u_1^{\pi}(s)\}\right)\right\}.$$

Write $S = \{s_1, ..., s_{|S|}\}$. Supposing Algorithm 1 was used to generate $\Pi_F^* = \{\pi_1, ..., \pi_n\}$, which we assume here to be finite, Theorem 4 suggests and justifies the following procedure for the calculation of V-optimal policies.

Algorithm 2. *Calculation of* Π_V^*.

Set $T = \Pi_F^*$.
For $i \in \{1, ..., |S|\}$ ***do:***
  For $j \in \{1, ..., n\}$ ***do:***
    For $k \in \{1, ..., n\}\backslash\{j\}$ ***do:***
      If $u_1^{\pi_k}(s_i) > u_1^{\pi_j}(s_i)$, ***drop*** π_j *from* T.

Proposition 4. *Algorithm 2 terminates with* $T = \Pi_V^*$.

Proof. At termination, a policy π is in T if and only if $\forall s \in S,\ \forall \pi' \in \Pi_F^*,\ u_1^{\pi'}(s) \not\succ u_1^{\pi}(s)$. Thus, $\pi \in T$ if and only if $\forall s \in S,\ u_1^{\pi}(s) \in e(\bigcup_{\pi\in\Pi_F^*}\{u_1^{\pi}(s)\})$. Consequently, $T = \Pi_V^*$ by the latter set's alternate representation. □

4. Numerical Experiments

The algorithms that have been developed were tested on randomly-generated instances of a three-state, two-action model with six epochs and varying numbers of criteria (m). In each

instance, the rewards and transition probabilities associated with an epoch t were sampled from exponential distributions then, in the case of the probabilities, scaled to $[0, 1]$.

Algorithm 1 was used to locate the efficient return functions in $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$, in addition to the corresponding F-optimal policies, through solving Equations (9) and (10) for $U_N = U_6$, U_5, U_4, U_3, U_2 then U_1. Algorithm 2 was used afterwards to determine which of the F-optimal policies were V-optimal.

To implement steps (1) and (2) of Algorithm 1, full search was used given the finite number of decision rules available. The experiments were programmed in C and run on a quadcore Intel Core i5-1145G7 laptop with 16GB of RAM.

The results are collected in Table 1, where the experiments are grouped by m into ten groups of a hundred experiments each. Bearing in mind that $U_1 = e(\bigcup_{\pi\in\Pi}\{u_1^\pi\})$ (Proposition 2), the column $|U_1|$ reports the minimum and maximum value observed in a single group of experiments of the number of efficient points in $\bigcup_{\pi\in\Pi}\{u_1^\pi\}$. Similarly, the column $|\Pi_F^*|$ contains the range of values taken by the number of F-optimal policies for a single group. Immediately to the right is a column indicating the range of CPU time expended on Algorithm 1. The last two columns provide analogous statistics for Algorithm 2, with $|\Pi_V^*|$ indicating the minimum and maximum number of V-optimal policies identified in a group.

m	$\lvert U_1\rvert$	$\lvert\Pi_F^*\rvert$	Algorithm 1 (seconds)	$\lvert\Pi_V^*\rvert$	Algorithm 2 (seconds)
1	1-2	1-2	0	1	0
2	11-253	11-253	0-0.0280	1-85	0
3	217-3903	217-3903	0.0120-2.4238	70-874	0-0.0040
4	853-7580	853-7580	0.2199-5.8955	249-6496	0
5	1259-7159	1259-7159	0.1440-6.7715	790-3345	0-0.0040
6	3817-21701	3817-21701	1.5478-39.5032	1726-18190	0-0.0040
7	2230-18169	2230-18169	0.4879-27.0269	1174-15611	0-0.0040
8	5668-21874	5668-21874	3.5077-56.9811	5123-14579	0-0.0040
9	2972-25937	2972-25937	1.8531-60.5031	2395-25937	0-0.0080
10	8005-27636	8005-27636	10.0394-93.4665	6558-27636	0-0.0079

TABLE 1. Results for randomly-generated problems grouped by m. The data is presented in minimum-maximum format.

As one would have predicted, the sizes of U_1, Π_F^* and $\Pi_V^* \subseteq \Pi_F^*$ tended to grow as m increased. The number of F-optimal policies ranged from one policy for a scalar-valued problem to 27 636 policies for a problem with ten objectives. The fraction of F-optimal policies that were V-optimal rose together with m, although V-optimal policies accounted in the majority of experiments for less than half of Π_F^*. The increase in the number of V-optimal policies is not

surprising in light of the definition of a V-optimal policy. When a policy π is *not* V-optimal, there are, by definition, a state s and a policy π' such that the inequality $u_1^{\pi'}(s)_i \geq u_1^{\pi}(s)_i$ holds for each objective $i = 1, ..., m$. As m increases, more inequalities have to be satisfied, making it less likely for a policy not to be V-optimal.

The computational demands for solving Equations (9) and (10) then finding every policy in Π_F^* and Π_V^* also tended to grow with m. For $m \leqq 4$, CPU time ranged from zero to six seconds for the two algorithms combined. Given that a typical multi-objective decision-making problem seldom exceeds four objectives (Stewart, Palmer, & DuPont, 2021), these results suggest that the algorithms possess the potential for being effective solution methods in a wide array of real world applications.

It should be noted that, relative to Algorithm 1, Algorithm 2 required negligible amounts of time due to the alternate representation of Π_V^* arrived at in Section 3. Recall that a policy π is V-optimal by definition if for all states s the vector $u_1^{\pi}(s)$ is efficient in $\bigcup_{\pi}\{u_1^{\pi}(s)\}$. This is equivalent, as we have seen, to efficiency of each $u_1^{\pi}(s)$ merely in the subset $\bigcup_{\pi \in \Pi_F^*}\{u_1^{\pi}(s)\}$. To fully appreciate the practicality of this alternate representation, we measured, during each of the previous experiments, the time needed to locate all V-optimal policies through an exhaustive search of the sets $\bigcup_{\pi}\{u_1^{\pi}(s)\}$, $s \in S$. In Table 2 we juxtapose the results against those already reported for Algorithm 2.

m	Exhaustive search (seconds)	Algorithm 2 (seconds)
1	10.4838-11.0478	0
2	15.5674-30.9460	0
3	26.7423-39.8178	0
4	31.0000-53.0195	0
5	54.9910-84.9934	0-0.0040
6	50.0692-98.9275	0-0.0040
7	79.7329-121.6538	0-0.0040
8	91.9299-146.0090	0-0.0040
9	94.4560-126.2161	0-0.0080
10	148.1788-186.5719	0-0.0079

TABLE 2. Comparison of the execution times of two methods of enumerating the set of V-optimal policies: exhaustive search of Π, and Algorithm 2. The problems considered here are those summarized in Table 1.

It is clear from the table that Algorithm 2, which relies on the alternate representation of Π_V^*, was consistently and substantially faster than a strategy based on direct optimization over

Π. The gap between the two methods became more pronounced as m increased. More importantly, whereas the time required by exhaustive search multiplied more than tenfold between $m = 1$ and $m = 10$, Algorithm 2 ran at a comparatively stable speed.

5. Application to Inventory Control

As a further illustration of our theoretical results, we consider the stochastic inventory control problem described in (Puterman, 2014, p. 38). The problem will be restated here for completeness. We take the position of a warehouse manager overseeing the inventory on hand of a particular product. Based on the inventory level at the beginning of each month, the manager may elect to order additional stock so long as it does not exceed the warehouse's capacity, M. The manager must keep sufficient inventory to meet external demand, but must also avoid overordering stock so as to minimize storage and ordering costs.

During each month, the events unfold as follows: (1) the decision is made to purchase (or not) additional stock; (2) the order is instantly fulfilled; (3) customer demand for the product arrives; then (4) if inventory is sufficient, all the demand is met on the last day of the month. External demand in month t, D_t, follows a time-homogeneous distribution $p_j = P(D_t = j)$, for all non-negative integers j. The cost of ordering u units in any month is $O(u) = K + c(u)$ if $u > 0$ and 0 if $u = 0$, where $K > 0$ is a fixed cost and c a nondecreasing function of u. The cost of holding u units between delivery of additional stock and sale of inventory at the end of a month is $h(u)$, h being a nondecreasing function of u. If the demand is for j units and sufficient inventory is available, then the revenue from selling those j units is $f(j)$, where f is nondecreasing in j.

Puterman proposes the following Markov decision process formulation. States represent the inventory level on the first day of a month, and actions represent the amount of stock the manager can order each month. In our notation,

$$S = \{0, ..., M\} \tag{15}$$

and

$$A_s = \{0, ..., M - s\} \tag{16}$$

for all $s \in S$. Thus, for example, if there are 3 units in the inventory for a warehouse capacity of 5, the manager can order up to two units of stock.

Let s_t be the state of the inventory in month t, and a_t the amount of stock ordered in that month. Clearly, s_{t+1} depends only on s_t, a_t and the random demand D_t. Then for any month

t and state $j \in S$:

$$p_t(j|s_t, a_t) = \begin{cases} 0 & \text{if } M \geq j > s_t + a_t \\ p_{s_t+a_t-j} & \text{if } M \geq s_t + a_t \geq j > 0 \\ q_{s_t+a_t} & \text{if } M \geq s_t + a_t \text{ and } j = 0 \end{cases} \tag{17}$$

where $q_{s_t+a_t} = 1 - \sum_{k=0}^{s_t+a_t-1} p_k$. Details of the derivation of (17) are supplied in (Puterman, 2014, p. 40).

The manager's objective in the original formulation is to maximize the difference between the expected revenue made over N months and the expected holding and ordering costs incurred in that period. Consequently, the reward received in month t is taken to be $F(s_t + a_t) - O(a_t) - h(s_t + a_t)$, where $F(u)$ is the expected monthly revenue when the stock level prior to demand is u. Details of the derivation of F from f and the p_j's are also provided in (Puterman, 2014, p. 39). We assume that the value of the inventory at the start of month N is nil.

Our only difference with the formulation above is that we treat revenue and costs as competing optimization objectives. Thus, we define the reward accrued in month t as:

$$R_t(s_t, a_t) = (F(s_t + a_t), -O(a_t) - h(s_t + a_t)) \tag{18}$$

with $R_N(s_N) = (0, 0)$. States, actions and transition probabilities are unaffected.

For comparative purposes, we take the same parameter values as (Puterman, 2014, p. 38): $N = 4$ months, $M = 3$, $K = 4$, $c(u) = 2u$, $h(u) = u$, and $f(u) = 8u$. Demand is distributed according to $p_0 = \frac{1}{4}$, $p_1 = \frac{1}{2}$, $p_3 = \frac{1}{4}$ and $\forall j > 3, p_j = 0$. Revenue is 8 per unit sold, inventory holding cost is 1 per unit, placing an order costs 4, and additional stock costs 2 per unit. The warehouse's capacity is 3 units at a time, and the planning horizon is of 4 months.

There is a total of $(M + 1)! = 24$ decision rules in this model. Over three decision epochs, this gives rise to 13824 policies. Algorithm 1 returned 1506 F-optimal policies, one of which is given in Table 3. Column one represents the inventory level at the beginning of a month, s. Columns two through four represent, respectively, $d_1(s)$, $d_2(s)$ and $d_3(s)$. Column five reports the value of $u_1^\pi(s)$. For example, if we start the first month with one item in stock, this policy recommends that two items be ordered at the end of the first and second months, then no item be ordered at the end of the third month. This would guarantee, in expectation, a revenue of 16 against a total cost of 12.7.

A subsequent comprehensive search of the policy space confirmed equality between the set of F-optimal policies, Π_F^*, and the set generated by Algorithm 1. This fact agrees with the conclusion that was drawn in Corollary 2 as to the ability of Algorithm 1 to identify all F-optimal policies.

As a test of the validity of Algorithm 2, we compared its output, $T = S_{\Pi_V^*}$, with the set of all V-optimal policies, Π_V^*, obtained after a full search. Of the 1506 F-optimal policies

TABLE 3. Example of an F-optimal policy $\pi = (d_1, d_2, d_3)$ for the stochastic inventory problem.

Start of month inventory	Order 1	Order 2	Order 3	(Exp. Revenue, - Exp. Costs)
0	1	2	0	(16.0, -14.7)
1	2	2	0	(16.0, -12.7)
2	0	1	0	(16.0, -6.7)
3	0	0	0	(22.0, -11.9)

mentioned earlier, 61 were V-optimal. The 61 matched the policies in T, thus corroborating the characterization of Π_V^* provided at the end of Section 3. Unsurprisingly, the "never order" strategy, which incurs minimal costs over the 4 months, was V-optimal.

It is instructive to compare the V-optimal policies constructed by solving the problem in its original scalar-valued formulation (Puterman, 2014, p. 96) with those obtained here. In fact, Puterman solves his problem under the same set of parameters (N, M, K, f, g, h, c) as ours, and concludes that the unique V-optimal policy is a nonstationary (Σ, σ) strategy, namely a policy of the form: "if units in the inventory are below σ at the start of the month, order enough stock to reach Σ units; otherwise, do not order" (Puterman, 2014, p. 38). We make three comments in this connection. First, whereas the optimal policy is unique in the scalar setting, there are 61 such policies in the vector setting. Second, Π_V^* contains both (Σ, σ) and non-(Σ, σ) policies, only one of which is stationary (the "never order" policy). Third, Puterman's optimal policy, which was selected to maximize the difference between revenue and cost, is also V-optimal. To see this, recall our comment in the discussion following Theorem 2 that a policy optimal with respect to a positive linear combination of the reward components, i.e., a π^* satisfying $\sum_{i=1}^m \lambda_i u_1^{\pi^*}(s)_i \geq \sum_{i=1}^m \lambda_i u_1^{\pi}(s)_i$ for some positive weights $\lambda_1, ..., \lambda_m > 0$, is V-optimal for the vector-valued model. In Puterman's case, $\lambda_1 = \lambda_2 = 1 > 0$, and the conclusion follows (note that the cost component of our rewards is given by the negative of the physical cost, so that the sum of the two components equals, in real terms, the difference between revenue and cost).

For each stock level $s \in S$, Figure 2 portrays the returns achieved by the V-optimal policies over the $N = 4$ months if the initial inventory level is s units; that is, each plot depicts $\bigcup_{\pi \in \Pi_V^*} \{u_1^{\pi}(s)\}$ for some $s \in S$. A point corresponds to one or several V-optimal policies. It should be stressed here that $\bigcup_{\pi \in \Pi_V^*} \{u_1^{\pi}(s)\}$, $s \in S$, is an efficient set by construction. This means that given two V-optimal policies π_1 and π_2 yielding different returns from an inventory level s, either π_1 generates a higher revenue while incurring greater costs than π_2, or vice versa. Therefore, each policy represented in $\bigcup_{\pi \in \Pi_V^*} \{u_1^{\pi}(s)\}$ expresses a particular tradeoff between costs and revenue. This is clearly reflected in all four plots.

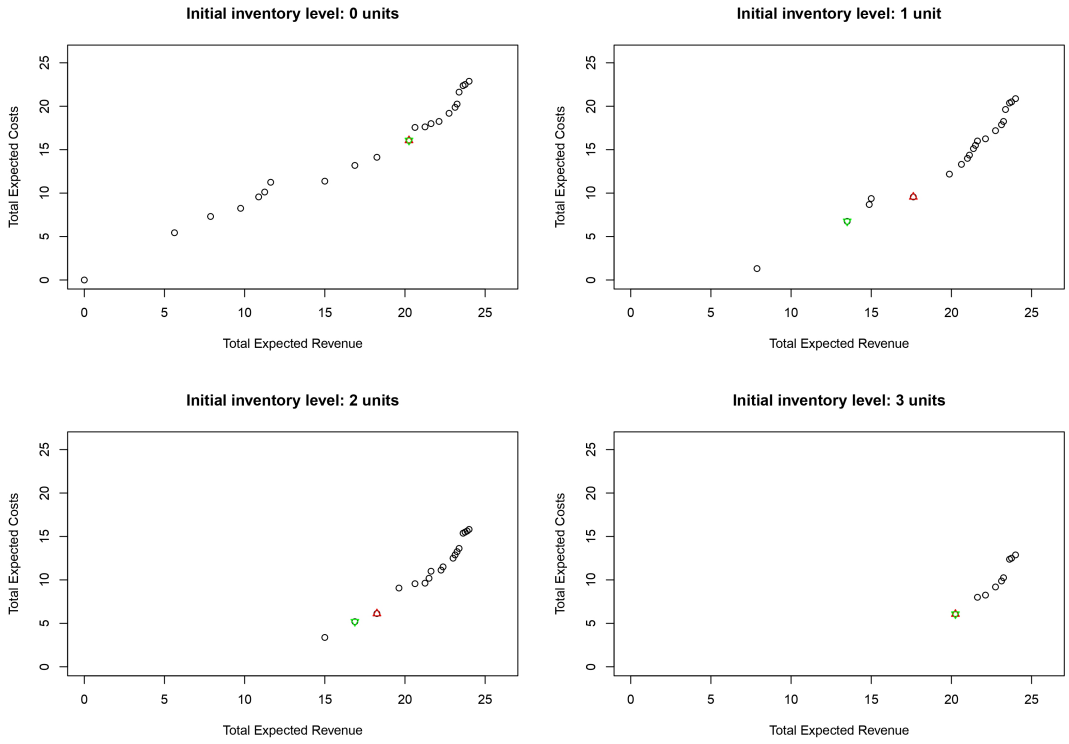

FIGURE 2. Graphical depiction of $\bigcup_{\pi\in\Pi_V^*}\{u_1^\pi(s)\}$, $s=0,1,2,3$. A point represents one or several V-optimal policies. Sixty-one policies are represented in each plot. Table 1's policy is highlighted in green, while Puterman's optimal (Σ,σ) policy is indicated in red.

Some final remarks on such plots as those of Figure 2 seem in order. When a decision is to be made as to what policy should be enacted among those supplied by such plots, the fact that $\bigcup_{\pi\in\Pi_V^*}\{u_1^\pi(s)\}$ is an efficient set means the decision maker has some latitude. For example, if among the m objectives there is a high priority objective, the decision maker will prefer policies that realize most gains in that objective. If, for example, the system has to operate under constraints, the decision maker will, when feasible, discard policies that violate these constraints. Moreover, access to such plots allows the decision maker to locate policies where small concessions in one objective produce considerable improvements in others. Thus, for instance, a car manufacturer may learn that a slight increase in costs allows for substantial reduction in tailpipe emissions, or a call center may realize that throughput may be greatly enhanced by hiring one additional employee. Examples such as these suggest that better informed decisions can be made when plots of $\bigcup_{\pi\in\Pi_V^*}\{u_1^\pi(s)\}$ are available.

6. Conclusions and Discussion

To summarize, this paper endeavored to solve a class of vector-valued Markov decision processes within two frameworks: (1) a policy is V-optimal if it delivers a maximal return from any initial state; and (2) a policy is F-optimal if its return function over the total decision-making horizon is maximal among all return functions. An exact dynamic programming algorithm was proposed for the second framework, which helped provide the basis for a procedure for calculating all V-optimal policies. Fundamental to the procedure were, first, the insight that framework (1) is subsumed under (2), and second, that a computationally useful representation of the set of V-optimal policies can be derived from this connection. Investigation of the set of F-optimal policies revealed that it satisfies a certain property which ensures the discovery of all such policies by the exact algorithm. The algorithms were illustrated with numerical experiments and a bi-objective variant of a stochastic inventory management problem.

In an effort to simplify the exposition, we have restricted ourselves to models with additive rewards, but our results extend to the multiplicative case provided that the rewards meet further assumptions. Specifically, let $x \circ y = (x_i y_i)_{1 \leq i \leq m}$ denote the componentwise product of x and y for any $x, y \in \mathbb{R}^m$, and let $u_t^\pi(s) = \mathbb{E}_\pi^s[R_t(X_t, d_t(X_t)) \circ ... \circ R_{N-1}(X_{N-1}, d_{N-1}(X_{N-1})) \circ R_N(X_N)]$ be the expected total reward for using $\pi = (d_1, ..., d_{N-1}) \in \Pi$ from t onward assuming the state at this epoch is s. Supposing then that $R_t(s, a)$ (resp., $R_N(s)$) has only nonnegative components for all $s \in S$, $a \in A$ and $t = 1, ..., N-1$ (resp., for all $s \in S$), every proposition in Section 3, except Lemma 3 and Theorem 3, follows without changes. A formal proof does not seem befitting at this stage of the paper, but the key points are these: (1) it is straightforward to check that $u_t^\pi(s) = R_t(s, d_t(s)) \circ \sum_{j \in S} p_t(j|s, d_t(s)) u_{t+1}^\pi(j)$ for all $s \in S$, $\pi = (d_1, ..., d_{N-1}) \in \Pi$, $t = 1, ..., N-1$; (2) expanding the sum in the previous expression yields the analogue of Equation (5) where "$\circ$" is substituted for "+"; and (3) with nonnegative reward components, $\circ$ preserves inequalities with respect to $\geq$. Points (1) and (3) suffice to prove Lemma 1. The resulting expression in point (2) suffices to show, as in the original proof of Theorem 1, that f_t is upper semicontinuous for all $t \leq N$, and therefore that Theorem 1 is correct. Proposition 1, Lemma 2, Birkhoff's theorem, Theorem 4 and Proposition 4 are unrelated to whether rewards are multiplicative or additive, and thus follow independently. The alternate representation of Π_V^* follows from Proposition 1 and Theorem 4. Corollary 1 is a consequence of Theorem 1. Theorem 2 follows from Corollary 1. The analogue of Lemma 3 where "$\circ$" replaces "+" on the left-hand side of the equality can be proven without difficulty, using the same arguments as the original proof. From this follows Proposition 3. Theorem 3 follows from the analogue of Lemma 3 of Theorem 1, with Equation (9) modified to reflect the change in Lemma 3. As Algorithm 1 and Proposition 2 are based entirely on Theorem 3, they remain valid *mutatis mutandis*. From the validity of Algorithm 1 follows that of Algorithm 2, and Proposition 4

holds. Finally, that Algorithm 1 is capable of finding all F-optimal policies, a result stated in Corollary 2, is an immediate consequence of Proposition 3.

A difficulty to be encountered when implementing Algorithm 1 is in the computation of efficient sets. In the numerical experiments as well as in the inventory problem, we used enumeration because the number of actions, and therefore the number of decision rules, was finite. If there is an infinity of actions, then analytic methods for determining the efficient return functions in Algorithm 1 may be required. As regards Algorithm 2, the requirement that Π_F^* be finite may be fulfilled even under a finite action space. However, an infinite Π_F^* would mean that an analytic alternative to Algorithm 2 would be in order. These questions will be the object of a future paper.

Finally, it remains to consider potential applications of these results in areas beyond inventory management. The operations research literature is replete with decision-making problems that can be treated as instantiations of the model studied here. For example, Chanson, Puterman, and Wong (1989) look at the problem of controlling the number of jobs processed by a computer at any given moment. Allowing too many jobs in memory can cause excessive competition for resources and hence considerable deterioration in performance. Just how many and *which* jobs are admitted to memory is determined by what the authors call a "load control" policy. The policy must minimize a weighted sum of the number of batch and interactive jobs in the system. The problem is formulated as a Markov decision process whose state is characterized by the number of jobs of either type and the fraction of jobs occupying memory. The actions permissible in any state are either to admit batch or interactive jobs to memory. Transition probabilities are derived from a queueing model, and the instantaneous cost of an action is defined as a weighted sum of the current number of jobs in each class. An alternative to this weighted sum approach would be to treat the classes as separate components of a vector-valued cost. In our notation, this would translate to a reward function $R_t(s, a) = (-N_1(t), -N_2(t))$, where $N_i(t)$ equals the number of class $i = 1, 2$ jobs in the system at time t.

Similar problems arise in other fields. In medicine, for instance, Denton, Kurt, Shah, Bryant, and Smith (2009) study when to begin administering statins to treat lipid abnormalities in diabetes patients. A tradeoff must be found between the expected future quality-adjusted life years (QALYs) and the annual treatment costs. Given the sequential and probabilistic aspects of the problem, a Markov decision process is used. The state is defined by the patient's risk factors at the moment of decision. At periodic intervals, the decision maker must elect either to initiate statins or delay treatment by one period. The goal of the study is to identify policies that maximize the expected long-term QALYs minus therapy costs over the patient's future. However, one might also treat QALYs and costs as two components of a vector-valued reward function.

In multi-objective problems such as these, the practitioner will enact the policy that represents their most preferred tradeoff between the criteria of interest. For the decision to be an informed one, the policy options at the practitioner's disposal must be diverse enough to shed light on the relationships between the criteria. At the same time, the options must be efficient, meaning there should not be alternatives that surpass them in all criteria. The algorithms described in this article generate policies that possess both of these characteristics. In that respect, we believe that they would make useful assets for a decision maker.

Data availability statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Disclosure of interest

The authors have no competing interests to disclose.

References

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, *60*(6), 503–515.

Berge, C. (1985). *Graphs and Hypergraphs*. Elsevier Science Ltd.

Birkhoff, G. (1940). *Lattice Theory* (Vol. 25). American Mathematical Soc.

Borrero, J., & Akhavan-Tabatabaei, R. (2013). Time and inventory dependent optimal maintenance policies for single machine workstations: An MDP approach. *European Journal of Operational Research*, *228*(3), 545-555.

Brown, T. A., & Strauch, R. E. (1965). Dynamic programming in multiplicative lattices. *Journal of Mathematical Analysis and Applications*, *12*(2), 364–370.

Burns, L. D., Hall, R. W., Blumenfeld, D. E., & Daganzo, C. F. (1985). Distribution strategies that minimize transportation and inventory costs. *Operations Research*, *33*(3), 469–490.

Chanson, S. T., Puterman, M. L., & Wong, W. C. (1989). A Markov decision process model for computer system load control. *INFOR: Information Systems and Operational Research*, *27*(3), 387–402.

Chen, D. Z., Trevizan, F., & Thiébaux, S. (2023). Heuristic search for multi-objective probabilistic planning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, pp. 11945–11954).

Coldman, A. J., & Murray, J. (2000). Optimal control for a stochastic model of cancer chemotherapy. *Mathematical Biosciences*, *168*(2), 187–200.

Denton, B. T., Kurt, M., Shah, N. D., Bryant, S. C., & Smith, S. A. (2009). Optimizing the start time of statin therapy for patients with diabetes. *Medical Decision Making*, *29*(3),

351–367.

Furukawa, N. (1980). Characterization of optimal policies in vector-valued Markovian decision processes. *Mathematics of Operations Research*, *5*(2), 271–279.

Geoffrion, A. M. (1968). Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, *22*(3), 618–630.

Goedhart, J., Haijema, R., Akkerman, R., & de Leeuw, S. (2023). Replenishment and fulfilment decisions for stores in an omni-channel retail network. *European Journal of Operational Research*.

Golabi, K., Kulkarni, R. B., & Way, G. B. (1982). A statewide pavement management system. *Interfaces*, *12*(6), 5-21.

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., ... others (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, *36*(1), 26.

Henig, M. I. (1983). Vector-valued dynamic programming. *SIAM Journal on Control and Optimization*, *21*(3), 490–499.

Henig, M. I. (1985). The Principle of Optimality in dynamic programming with returns in partially ordered sets. *Mathematics of Operations Research*, *10*(3), 462–470.

Mandow, L., Pérez-de-la Cruz, J.-L., & Pozas, N. (2022). Multi-objective dynamic programming with limited precision. *Journal of Global Optimization*, *82*(3), 595–614.

Mason, J., Denton, B., Shah, N., & Smith, S. (2014). Optimizing the simultaneous management of blood pressure and cholesterol for type 2 diabetes patients. *European Journal of Operational Research*, *233*(3), 727–738.

Mifrani, A. (2025). A counterexample and a corrective to the vector extension of the Bellman equations of a Markov decision process. *Annals of Operations Research*, *345*(1), 351–369.

Morin, T. L. (1982). Monotonicity and the Principle of Optimality. *Journal of Mathematical Analysis and Applications*, *88*(2), 665–674.

Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Ramirez-Nafarrate, A., Hafizoglu, A. B., Gel, E. S., & Fowler, J. W. (2014). Optimal control policies for ambulance diversion. *European Journal of Operational Research*, *236*(1), 298–312.

Roijers, D., Röpke, W., Nowe, A., & Radulescu, R. (2021, July 14). On following Pareto-optimal policies in multi-objective planning and reinforcement learning.. Retrieved from `http://modem2021.cs.nuigalway.ie/` (Multi-Objective Decision Making Workshop 2021, MODeM 2021 ; Conference date: 14-07-2021 Through 16-07-2021)

Ruiz-Montiel, M., Mandow, L., & Pérez-de-la Cruz, J.-L. (2017). A temporal difference method for multi-objective reinforcement learning. *Neurocomputing*, *263*, 15–25.

Schlosser, R., & Gönsch, J. (2023). Risk-averse dynamic pricing using mean-semivariance

optimization. *European Journal of Operational Research*, *310*(3), 1151-1163.

Stewart, R. H., Palmer, T. S., & DuPont, B. (2021). A survey of multi-objective optimization methods and their applications for nuclear scientists and engineers. *Progress in Nuclear Energy*, *138*, 103830.

Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research*, *15*(1), 3483–3512.

Wang, L., Demeulemeester, E., Vansteenkiste, N., & Rademakers, F. E. (2024). Capacity and surgery partitioning: An approach for improving surgery scheduling in the inpatient surgical department. *European Journal of Operational Research*, *313*(1), 112-128.

White, D. J. (1982). Multi-objective infinite-horizon discounted Markov decision processes. *Journal of Mathematical Analysis and Applications*, *89*(2), 639–647.

White, D. J. (1993). A survey of applications of Markov decision processes. *Journal of the Operational Research Society*, *44*(11), 1073–1096.

Wiering, M. A., & De Jong, E. D. (2007). Computing optimal stationary policies for multi-objective Markov decision processes. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (pp. 158–165).

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, *8*(1), 59–60.

Zorn, M. (1935). A remark on method in transfinite algebra. *Bulletin of the American Mathematical Society*, *41*(10), 667–670.