Published in final edited form as:
J Mach Learn Res. 2016 ; 17: .

Multi-Objective Markov Decision Processes for Data-Driven Decision Support

Daniel J. Lizotte and
Department of Computer Science, Department of Epidemiology & Biostatistics, The University of Western Ontario, 1151 Richmond Street, London, ON N6A 3K7, Canada

Eric B. Laber
Department of Statistics, North Carolina State University, Raliegh, NC 27695, USA

Abstract

We present new methodology based on Multi-Objective Markov Decision Processes for developing sequential decision support systems from data. Our approach uses sequential decision-making data to provide support that is useful to many different decision-makers, each with different, potentially time-varying preference. To accomplish this, we develop an extension of fitted-Q iteration for multiple objectives that computes policies for all scalarization functions, i.e. preference functions, simultaneously from continuous-state, finite-horizon data. We identify and address several conceptual and computational challenges along the way, and we introduce a new solution concept that is appropriate when different actions have similar expected outcomes. Finally, we demonstrate an application of our method using data from the Clinical Antipsychotic Trials of Intervention Effectiveness and show that our approach offers decision-makers increased choice by a larger class of optimal policies.

Keywords

multi-objective optimization; reinforcement learning; Markov decision processes; clinical decision support; evidence-based medicine

1. Introduction

Markov Decision Processes (MDPs) (Bertsekas and Tsitsiklis, 1996) provide a framework for reasoning about the actions of an autonomous decision-making agent in an environment as it strives to achieve long-term success. Operating within this framework, reinforcement learning (RL) methods for finding optimal actions in MDPs hold great promise for using vast amounts of accumulating longitudinal data to help humans make better-informed decisions. Batch reinforcement learning methods, including fitted Q-learning (Ernst et al., 2005), A-learning (Blatt et al., 2004), and regret regression (Henderson et al., 2010), are already being used to aid decision-making in diverse areas including medicine (Alagoz et al., 2010; Shortreed et al., 2011; Burnside et al., 2012), ecology (Păduraru et al., 2012), intelligent tutoring systems (Brunskill and Russell, 2011), and water reservoir control

(Castelletti et al., 2010). Although headway has been made in these application areas, progress is hampered by the fact that many sequential decision *support* problems are not modelled well by MDPs.

One reason for this is that in most cases, human action selection is driven by multiple competing objectives; for example, a medical decision will be based not only on the effectiveness of a treatment, but also on its potential side-effects, cost, and other considerations. Because the relative importance of these objectives varies from user to user, the quality of a policy is not well captured by a universal single scalar "reward" or "value." Multi-Objective Markov Decision Processes (MOMDPs) accommodate this by allowing vector-valued rewards (Roijers et al., 2013) and proposing an application-dependent *solution concept*. A solution concept is essentially a partial order on policies; the set of policies that are maximal according to the partial order are considered "optimal" and are indistinguishable under that solution concept. Depending on the application, a single policy may be selected from among these, or a set of policies may be presented in some way. Computing and presenting a set of policies is termed the *decision support* setting by Roijers et al. and is the setting we consider here.

2. Existing Methods and Our Contributions

Roijers et al. (2013) note that, "…there are currently no methods for learning multiple policies with non-linear [preferences] using a value-function approach." We present a method that fills this gap, and that additionally uses value function approximation to accommodate continuous state features, thus allowing us to use the MOMDP framework to analyze continuous-valued sequential data. Previous work (Lizotte et al., 2012) on this problem computes a set of policies based on the assumptions that i) end-users have a "true reward function" that is linear in the objectives and ii) all future actions will be chosen optimally with respect to the same "true reward function" over time. Our new method relaxes both of these assumptions as it allows the decision-maker to revisit action selection at each decision point in light of new information, both about state and about their own preferences and priorities over different outcomes of interest. Therefore, the proposed method can accommodate changes in preference over time while still making optimal decisions according to our new solution concepts by introducing the *non-deterministic multi-objective fitted-Q* algorithm, which computes policies for all scalarization functions, i.e., preference functions, simultaneously from continuous-state, finite-horizon data. This allows us to present a greater variety of action choices by acknowledging that preference functions may be non-linear. We then present the vector-valued expected returns associated with the different policies in order to provide decision support without having to refer to any particular scalarization function. Showing the expected returns in the original reward space allows us to more easily understand the qualitative differences between action choices. Although decision support is important in many application areas, we are motivated by clinical decision-making; therefore we demonstrate the use of our algorithm using data from the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE).

Simplified versions of some of our ideas were presented in a shorter paper by Laber et al. (2014a), but we treat the problem in its full generality here. In particular, our work goes beyond "Set-Valued Dynamic Treatment Regimes" in four significant ways:

- We introduce a complete *non-deterministic fitted-Q* algorithm that is applicable to arbitrary numbers of actions and arbitrary time horizons. (Previous work was limited to binary actions and maximum two decision points.) This allows us to perform fitted-Q backups in general settings using multiple reward functions over continuous-valued state features.
- We prove that our algorithm finds all policies that are optimal for some scalarization function by considering a collection of policies at the next time step that is only polynomial in the data set size.
- We formalize a solution concept, *practical domination*, that is more flexible than Pareto domination for identifying whether an action is not desirable. A similar concept was introduced in previous work (Laber et al., 2014a), but we show that using practical domination, while useful, is problematic for more than two decision points because it is does not induce a partial order on actions. However, we show that a modification of practical domination leads to a partial ordering for any number of actions or time points.
- We demonstrate the use of our algorithm on the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) and we compare our approach quantitatively and qualitatively with a competing approach derived from previous work of Lizotte et al. (2010, 2012).

3. Motivation

Our work is motivated by a clear opportunity for reinforcement learning methods to provide novel ways of analyzing data to produce high-quality, evidence-based decision support. We briefly review some specific applications here where we believe our approach could be particularly relevant.

3.1 Intelligent Tutorial Systems

Brunskill and Russell (2011), and Rafferty et al. (2011) study the automatic construction of adaptive pedagogical strategies for intelligent tutoring systems. They employ POMDP models to capture the partially observable and sequential aspect of this problem, using hidden state to represent a student's knowledge. Their approach uses time taken to learn all skills as a cost, i.e., negative reward, that drives teaching action selection. Chi et al. (2011) use an MDP formulation and use "Normalized Learning Gain," a quantification of skill acquisition, as a reward; however, they do not explicitly consider time spent. The ability to consider both of these rewards simultaneously would empower the learner or the teacher to emphasize one or the other over the course of their interaction with the system. The method we present could offer a selection of teaching actions that are all optimal for different preferences over these rewards, and possibly others as well.

3.2 Computational Sustainability

Păduraru et al. (2012) identify an application within the domain of sustainable wildlife management where the MDP framework is particularly appropriate. They investigate the efficacy of several off-policy methods for developing control policies for mallard duck populations. Their output, rather than providing autonomous control, is intended to provide decision support for public environmental policy-makers. They use "number of birds harvested per year" as the reward. However, in practical management plans, several outcomes may be of interest including minimum population size, program cost, and so on. Because formulating a (e.g. linear) trade-off among these rewards would be difficult, our method is relevant to this problem.

3.3 Treating Chronic Disease

Reinforcement learning has also been used as a means of analyzing sequential medical data to inform clinical decision-makers of the comparative effectiveness of different treatments (Shortreed et al., 2011). RL methods are suited to decision support for treating chronic illness where a good *policy* for choosing treatments over time is crucial for success. Indeed, optimal policies—known as "Dynamic Treatment Regimes" in statistics and the behavioral sciences—have been learned for the management of chronic conditions including attention deficit hyperactivity disorder (Laber et al., 2014b), HIV infection (Moodie et al., 2007), and smoking addiction (Strecher et al., 2006). They have also been applied to sequences of diagnostics as well, for example in breast cancer (Burnside et al., 2012). We present a case study in this domain in Section 7.

4. Background

We introduce a new approach for solving Multi-Objective Markov Decision Processes with the goal of providing data-driven decision support. Our approach uses non-deterministic policies to encode the set of all non-dominated policies. In this section, we review the most relevant existing literature on MOMDPs and NDPs.

4.1 Multi-objective Optimization and MOMDPs

The most basic definition of a Markov Decision Process is as a 4-tuple $\langle \mathscr{S}, \mathscr{A}, P, R \rangle$ where $\mathscr{S}$ is a set of states, $\mathscr{A}$ is a set of actions, $P(s, a, s') = \Pr(s'|s, a)$ gives the probability of a state transition given action and current state, and $R(s, a)$ is the immediate scalar reward obtained in state s when taking action a. One common goal of "solving" an MDP, if we assume a finite time horizon of T steps, is to find a policy $\pi : \mathscr{S} \to \mathscr{A}$ that maximizess

$$V^{\pi}(s)=\mathbb{E}^{\pi}[\sum_{t=1}^{T} R(s_t, a_t)|s_1=s]$$

pointwise for all states. In the preceding, $\mathbb{E}^{\pi}$ indicates that the expectation is taken assuming the state-action trajectories are obtained by following policy π. Because in the finite-horizon setting that the optimal π is in general non-stationary (Bertsekas, 2007), we define π to be a sequence of functions π_t for $t \in \{1, \ldots, T\}$, where $\pi_t: \mathscr{S}_t \to \mathscr{A}_t$.

Like previous work by Lizotte et al. (2010, 2012) and by many others (Roijers et al., 2013), we focus on the setting where the definition of an MDP is augmented by assuming a D-dimensional reward vector $\mathbf{R}(s_t, a_t)$ is observed at each time step. We define a finite-horizon MOMDP with finite time horizon T as a tuple of state spaces $\mathscr{S}_t$, action spaces $\mathscr{A}_t$, state transition functions P_t: $\mathscr{S}_t \times \mathscr{A}_t \to \mathbb{P}(\mathscr{S}_{t+1})$ where $\mathbb{P}(\mathscr{S}_{t+1})$ is the space of probability measures on $\mathscr{S}_{t+1}$, and reward functions $\mathbf{R}_t$: $\mathscr{S}_t \times \mathscr{A}_t \to \mathbb{R}^D$ for $t \in \{1, \ldots, T\}$. In keeping with the Markov assumption, both $\mathbf{R}_t$ and P_t depend only on the current state and action. In this work we assume finite action sets, but we do *not* assume that state spaces are finite. The *value* of a policy $\boldsymbol{\pi}$ is then given by

$$\mathbf{V}^{\pi}(s) = \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} \mathbf{R}^{t}(s_t, a_t) | s_1 = s\right] \quad (1)$$

which is the expected sum of (vector-valued) rewards we achieve by following policy $\boldsymbol{\pi}$.

Just as “solving” an MDP is an optimization problem (i.e. we want the optimal value function or policy), “solving” a MOMDP is a *multi-objective optimization* (MOO) problem. Whereas in typical scalar optimization problems having a unique solution is viewed as typical or at least desirable, in the MOO setting, the most common goal is to produce a *set* of solutions that are *non-dominated.*

Definition 1 (Non-dominated a.k.a. Pareto optimal solutions)—*Let $\mathscr{X}$ be the set of all feasible inputs to a multi-objective optimization problem with objective* $\mathbf{f}(x)$. *Let $\mathscr{Y}$ be the range of* $\mathbf{f}$ *on $\mathscr{X}$. (In the RL context, one can think of $\mathscr{X}$ as the set of all possible policies starting from a given state, and $\mathscr{Y}$ as their corresponding values, which are vectors in this case.) A solution vector* $\mathbf{y} \in \mathscr{Y}$ *is* non-dominated *if* $\nexists \mathbf{y}' \in \mathscr{Y}$ *s.t.* $\forall_i y_i' \geqslant y_i$ *and* $\exists_i y_i' > y_i$. *A preimage* $\mathbf{x} \in \mathscr{X}$ *of such a* $\mathbf{y}$ *is sometimes called an* efficient solution, *but we will also refer to such inputs as* non-dominated.

A common goal in MOO is to find *all* of the non-dominated solutions (Miettinen, 1999; Ehrgott, 2005). Some work on MOMDPs has this same goal (Perny and Weng, 2010). One approach to finding non-dominated solutions of a MOO problem is to solve a set of optimization problems that are *scalarized* versions of the MOO. A *scalarization function* ρ is chosen which maps vector-valued outcomes scalars and then one solves

$$\max_{x \in \mathscr{X}} \rho(\mathbf{f}(x)),$$

the scalar optimization defined by composing the vector-valued outcome function with the scalarization function. If a specific “correct” scalarization function is fixed and known, we can simply apply it to all outcome vectors and reduce our MOO problem to a scalar optimization problem (and arguably we never had a MOO problem to begin with.) Otherwise, we may seek solutions to scalarized problems for all ρ belonging to some function class. We assume that any ρ of interest is non-decreasing along each dimension, that is, it is always preferable to increase a reward dimension, all else being equal. It is well-known (Miettinen, 1999) that the set of Pareto-optimal solutions corresponds to the set of all solutions that are optimal for some scalarization function; we will use these two views of optimality as we construct our algorithm.

Previous work by Lizotte et al. (2012) uses dynamic programming to compute policies for all possible scalar rewards that are a convex combination of the basis rewards, using Q-functions learned by linear regression. Thus the output produces the optimal policies for all ρ such that $\rho(\mathbf{r}) = \mathbf{r}^\top \mathbf{w}$, $w_d > 0$, $\sum_d w_d = 1$. Each convex combination is interpreted as a preference describing the relative importance of the the different basis rewards, and the method is used to show how preference relates to optimal action choice. This gives a new and potentially useful way of visualizing the connection between preference and action choice, but there are drawbacks to the approach. First, one must assume that the convex combination is fixed for all time points—that is, preferences do not change over time. This assumption enables dynamic programming to work, but is not reasonable for some applications, particularly in clinical decision-making where a patient’s first-hand experience with a treatment may influence subsequent preferences for symptom versus side-effect reduction. Second, the method is overly eager to eliminate actions. Consider two actions a_1 and a_2 that are extreme, e.g., a_1 has excellent efficacy but terrible side-effects and a_2 has no side-effects but poor efficacy. These could eliminate a third action a_3 that is moderately good according to both rewards. An example of this situation is illustrated in Figure 1(a), which shows that the actions chosen by this method are restricted to the convex hull of the Pareto frontier, rather than the entire frontier. In this circumstance where a_3 is qualitatively very different from both a_1 and a_2, we argue that a decision support system should suggest *all three* treatments and thereby allow the decision-maker to make the final choice based on her expertise. The third drawback of using this approach is that it is limited to ordinary least squares regression, which may not work well for data with non-Gaussian errors, e.g., a binary terminal reward.

Rather than use the method of Lizotte et al., we will instead base our method on assessing actions and policies using a *partial order* on their vector of Q-values. Perhaps the most common partial order on vectors comes from the notion of *Pareto-optimality* (Vamplew et al., 2011). For example, an action a is *Pareto-optimal* at a state s_T if $\forall(d, a') \; Q_{T[d]}(s_T, a) \geqslant Q_{T[d]}(s_T, a')$. We will show in Section 5 that for $t < T$, the problem of deciding which actions are optimal is more complex, but we will still leverage the idea of a partial order. The problem of identifying Pareto-optimal policies is of significant interest in RL (Perny and Weng, 2010; Vamplew et al., 2009) and is closely related to what we wish to accomplish. Basing our work on the Pareto-optimal approach rather than on the previous work of Lizotte et al. avoids assuming that preferences are fixed over time, and it avoids the problem of “extreme” actions eliminating “moderate” ones. Furthermore, our approach works with a larger class of regression models, including ordinary least squares, the lasso, support vector regression, and logistic regression. While our Pareto-based approach makes these three improvements, using Pareto-optimality can *still* result actions being eliminated unnecessarily; this is illustrated in Figure 1(b). We address this problem by introducing an alternative notion of domination in Section 6. Each of these contributions leads to increased action choice for the decision-maker by considering a larger class of preferences over reward vectors.

4.2 Non-Deterministic Policies

Milani Fard and Pineau (2011) describe non-deterministic policies for Markov Decision Processes (MDPs) with a finite state space and a single reward function. The term *non-deterministic* is used as in the study of *non-deterministic finite automata* and indicates that there are choices made as the system evolves about which we assume we have no information.[1] Given an MDP with *state space* $\mathscr{S}$ and an *action set* $\mathscr{A}$, an NDP Π is a map from the state space to the set $2^{\mathscr{A}}\backslash\{\varnothing\}$. Milani Fard and Pineau assume that a *user* operating the MDP will, at each timestep, choose an action from the set Π(s). They are motivated by the same considerations that we are in the sense that they wish to provide choice to the user while still achieving good performance; thus, they only eliminate actions that are clearly sub-optimal. Because they consider only a single reward function, they can measure performance using the expected discounted infinite sum of future (scalar) rewards in the usual way, and they can produce an NDP Π that has near-optimal performance even if the user chooses the "worst" actions from Π(s) in each state.

One can view the NDP as a compact way of expressing a set of policies that might be executed. Suppose that $\#A = |\Pi(s)|$, the number of actions provided by the NDP Π, is the same at all states. Then the number of policies that are *consistent* with Π, that is, the policies for which $\boldsymbol{\pi}(s) \in \Pi(s)$, is $\#A^{|\mathcal{S}|}$. So the NDP Π is a compact encoding of an exponential number of policies. We will make use of this property to encode our policies. The two most important differences between our work and that of Milani Fard and Pineau are that our motivation for learning non-deterministic policies is driven explicitly by having more than one basis reward of interest, and that we use more general value function models rather than a tabular representation. Having multiple basis rewards combined with value function approximation leads us to a different, novel algorithm for learning NDPs.

5. Fitted-*Q* for MOMDPs

Our non-deterministic fitted-*Q* algorithm for multiple objectives uses finite-horizon, batch data. We present a version that uses linear value function approximation because this model is commonly used by statisticians working in clinical decision support (Strecher et al., 2006; Lizotte et al., 2010, 2012; Laber et al., 2014b), and because available data often contain continuous-valued features, e.g., symptom and side-effect levels, laboratory values, etc., and outcomes, e.g., symptom scores, body mass index. It is a flexible model because we will not restrict the state features one might use. For learning, we assume a batch of n data trajectories of the form

$$s_1^i, a_1^i, r_{1[1]}^i, \ldots, r_{1[D]}^i, s_2^i, a_2^i, r_{2[1]}^i, \ldots, r_{2[D]}^i, \ldots, s_T^i, a_T^i, r_{T[1]}^i, \ldots, r_{T[D]}^i \text{ for } i=1, \ldots, n.$$

[1] Note that *non-deterministic* does not mean "stochastic"; i.e., we do not suppose a known stationary random policy will be followed.

In the following exposition, we begin by specifying how the algorithm works for the last time point $t = T$. This would be the only step needed in a “non-sequential” decision problem. We then describe the steps analogous to the fitted-Q “backup” operation for earlier timepoints $t < T$, which are more complex.

5.1 Final time point, *t* = *T*

At time T, we define the approximate Q-function for reward dimension d as the linear least squares fit

$$\hat{Q}_{T[d]}(s_T, a_T) = \phi_T(s_T, a_T)^\top \hat{\mathbf{w}}_{T[d]}, \hat{\mathbf{w}}_{T[d]} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \left(\phi_T(s_T^i, a_T^i)^\top \mathbf{w} - r_{T[d]}^i \right)^2 \quad (2)$$

giving the estimated vector-valued expected reward function

$$\hat{\mathbf{Q}}_T(s_T, a_T) = (\hat{Q}_{T[1]}(s_T, a_T), \ldots, \hat{Q}_{T[D]}(s_T, a_T))^\top. \quad (3)$$

Here, $\phi_T(s_T, a_T)$ is a feature vector of state and action. As discussed by Lizotte et al. (2012), $\phi_T(s_T, a_T)$ would typically include: a constant component for the intercept, features describing s_T, dummy variables encoding the discrete action a_T, and the product of the dummy variables with the features describing s_T (Cook and Weisberg, 1999). One could also include other non-linear functions of s_T and a_T as features if desired. We present our method assuming that $\hat{\mathbf{w}}_{T[d]}$ are found by least squares regression, but one could for example add an L_1 penalty, or use support vector regression (Hastie et al., 2001). Furthermore, unlike previous work by Lizotte et al. (2012), any Generalized Linear Model (GLM) with a monotonic increasing link function (e.g. logistic regression, Poisson regression, and so on) can also be used (Cook and Weisberg, 1999). Note that we can recover a “tabular” representation if the states are discrete and we assign mutually orthogonal feature vectors to each one.

Having obtained the $\hat{\mathbf{Q}}_T$ from (2), we construct an NDP Π_T that will give, for each state, the actions one might take at the last time point. For each state s_T at the last time point, each action a_T is associated with a *unique* vector-valued estimated expected reward given by $\hat{\mathbf{Q}}_T(s_T, a_T)$. Thus, we decide which among these vectors is a desirable outcome, and include their associated actions in $\Pi_T(s_T)$. Our main focus will be to construct $\Pi_T(s_T)$ for each state based on the multi-objective criterion of Pareto optimality; however, an important advantage of our algorithm is that it can use other definitions of Π_T as well; we discuss an extension in Section 6. For example, different definitions of Π_T allow us to recover other varieties of Q-learning:

Scalar Fitted-Q: Defining

$$\Pi_T(s_T) = \{\arg \max_a \hat{Q}_{T[0]}(s_T, a)\}$$

gives standard fitted-Q applied to reward dimension 0.

Convex Pareto-optimal Fitted-Q: Defining

$$\Pi_T(s)=\{a:\exists\boldsymbol{\theta}\,(\boldsymbol{\theta}\geqslant\mathbf{0}\wedge\boldsymbol{\theta}^\top\mathbf{1}=1\wedge\forall a'\hat{\mathbf{Q}}_T(s_T,a)^\top\boldsymbol{\theta}\geqslant\hat{\mathbf{Q}}_T(s_T,a')^\top\boldsymbol{\theta})\}$$

includes those actions whose expected reward is on the convex hull of the Pareto frontier; these are the actions that would be included by the previous method of Lizotte et al. (2012).

Pareto-optimal Fitted-Q: Defining

$$\Pi_T(s)=\{a:\nexists a'(\forall d\,\hat{Q}_{T[d]}(s_T,a)<\hat{Q}_{T[d]}(s_T,a'))\}$$

includes precisely those actions whose expected reward is on the (weak) Pareto frontier; this is a superset of those included by the method of Lizotte et al. (2012). It includes the actions that are optimal for some scalarization function, which is our action set of interest.

5.2 Earlier time points, $t < T$

For $t < T$, it is only possible to define the expected return of taking an action in a given state by also deciding which particular policy will be followed to choose future actions. In standard fitted-Q, for example, one assumes that the future policy is given by $\pi_j(s) = \arg\max_a \hat{Q}_j(s, a)$ for all $j > t$. In the non-deterministic setting, we may know that the future policy belongs to some set of possible policies derived from Π_j for $j > t$, but in general we do not know which among that set will be chosen; therefore, we explicitly include the dependence of $\hat{\mathbf{Q}}_t$ on the choice of future policies π_j, $t < j \leqslant T$:

$$\hat{\mathbf{Q}}_t(s_t,a_t;\pi_{t+1},\ldots,\pi_T)=\left\{\hat{Q}_{t[1]}(s_t,a_t;\pi_{t+1},\ldots,\pi_T),\ldots,\hat{Q}_{t[D]}(s_t,a_t;\pi_{t+1},\ldots,\pi_T)\right\}^\top$$

where

$$\text{for }d=1,\ldots,D,\hat{Q}_{t[d]}(s_t,a_t;\pi_{t+1},\ldots,\pi_T)=\phi_t(s_t,a_t)^\top\hat{\mathbf{w}}_{t[d]\pi_{t+1},\ldots,\pi_T},$$

and

$$\hat{\mathbf{w}}_{t[d]\pi_{t+1},\ldots,\pi_T}=\underset{\mathbf{w}}{\operatorname{argmin}}\sum_{i=1}^{n}\left[\phi_t(s_t^i,a_t^i)^\top\mathbf{w}-\left\{r_{t[d]}^i+\hat{Q}_{t+1[d]}(s_{t+1}^i,\pi_{t+1}(s_{t+1}^i);\pi_{t+2},\ldots,\pi_T)\right\}\right]^2. \quad (4)$$

We say an expected return is *achievable* if it can be obtained by taking some immediate action in the current state and following it with a fixed sequence of policies until we reach the last time point.

We use $\mathscr{Q}_t$ to denote a set of partially-evaluated Q-functions; each member of $\mathscr{Q}_t$ is a function of s_t and a_t only and assumes a particular fixed sequence $\boldsymbol{\pi}_{t+1}, \ldots, \boldsymbol{\pi}_T$ of future policies. Precisely which future policies should be considered is the subject of the next section. For the last time point, we define $\mathscr{Q}_T = \{\hat{\mathbf{Q}}_T\}$, the set containing the single (multivariate) Q-function for the last time point. Figure 2 is a visualization of an example $\mathscr{Q}_{T-1}$ where each function in the set is evaluated at the same given state and for each of the five available actions, {▼, ■, ▲, ▶, ◀}. Thus, each element of the example $\mathscr{Q}_{T-1}$ corresponds to a collection of five markers on the plot, one for the expected return for each action, assuming we follow a particular $\boldsymbol{\pi}_T$. The question of what collection of $\boldsymbol{\pi}_T$ we should consider is the subject of the next section.

5.3 Constructing Π_t from Π_{t+1}

We now describe the "backup" step that constructs Π_t and $\mathscr{Q}_t$ from Π_{t+1} and $\mathscr{Q}_{t+1}$. A member of $\mathscr{Q}_t$ is constructed from data using equation (4) by choosing two components: An element of $\mathscr{Q}_{t+1}$ (with its implicit choice of $\boldsymbol{\pi}_{t+2}$ through $\boldsymbol{\pi}_T$) and a policy $\boldsymbol{\pi}_{t+1}$. When considering different possible $\boldsymbol{\pi}_{t+1}$, we restrict our attention to policies that i) are *consistent* with Π_{t+1}, and ii) are *representable* using the approximation space chosen for $\hat{\mathbf{Q}}_{t+1}$. In the following, we define these notions of consistency and representability, argue that this subset of policies contains all those we need to consider, and show how the set of consistent and representable policies can be efficiently enumerated using mixed integer linear programming.

To construct $\mathscr{Q}_t$, we will only consider future policies that are consistent with the NDPs we have already learned for later time points. As described above, each $\Pi(s)$ contains each action for which some scalarization function (i.e. preference) prefers that action.

Definition 2 (Policy consistency)—*A policy* $\boldsymbol{\pi}$ *is* consistent *with an NDP* Π, *denoted* $\boldsymbol{\pi} \sqsubset \Pi$, *if and only if* $\boldsymbol{\pi}(s) \in \Pi(s)\ \forall s \in \mathscr{S}$. *We denote the set of all policies consistent with* Π *by* $\mathscr{C}(\Pi)$.

This restriction is analogous to fitted-Q in the scalar reward setting, where we estimate the current Q function assuming we will follow the greedy policy of the estimated optimal Q function at later time points. In our setting, there are likely to be multiple different policies whose values, pointwise at each state, are considered "optimal," e.g. that are Pareto non-dominated. Although we cannot pare down the possible future policies to a single unique choice as in scalar fitted-Q, we can still make significant computational savings. Note that in the batch RL setting, two policies are distinguishable only if they differ in action choice on states observed in our data set. In the following, when we talk about the properties of policies, we mean in particular over the observed states in our data set. Where clarification is needed, we write $\mathscr{S}_t^n$ to mean the n states observed in our data set at time t. Note that $|\mathscr{C}(\Pi_t)| = \times_{s_t \in \mathscr{S}_t^n} |\Pi_t(s_t)|$, the product of the cardinalities of the sets produced by Π_t over the observed data. Because $|\Pi_t(s_t)| \leqslant |\mathscr{A}|$, we have $|\mathscr{C}(\Pi_t)| \leqslant |\mathscr{A}|^n$. If Π_t screens out enough actions from enough observed states, restriction to consistent policies can result in a much smaller $\mathcal{Q}_t$. Unfortunately, in the worst case where $\forall s_t\, \Pi_t(s_t) = \mathcal{A}_t$, we have $|\mathcal{C}(\Pi_t)| = |\mathcal{A}|^n$, and if for some fraction η of the n trajectories $(0 < \eta \leqslant 1)$ we have $|\Pi_t(s_t)| \geqslant 2$, then we have $|\mathcal{C}(\Pi_t)| \in \Omega(2^n)$. Therefore in many interesting cases, computing a $\mathcal{Q}_t$ that includes even just the consistent future policies is computationally intractable.

We therefore impose a further restriction on possible future policies, again only eliminating policies we do not wish to consider. In scalar fitted-Q, the learned optimal policy is given by $\text{argmax}_a\, Q(s, a)$. If the learned Q-functions are linear in some feature space, then the learned optimal policy can be represented by a collection of linear separators that divide feature space into regions where different actions are chosen. This is true for *any* scalar reward signal. Therefore, in the scalar reward case for a given feature space, any future policy that cannot be represented in this way will never considered when computing $\hat{Q}$ for earlier timepoints no matter what the observed rewards are.

In NDP settings where $\dim \phi_t(s_t, a_t) \ll n$, most of the policies that are consistent with $\Pi_t(s_t)$ are not representable in the form $\pi(s_t) = \text{argmax}_a\, Q_t(s_t, a)$, and therefore would never be learned by fitted-Q iteration using *any* scalar reward signal. Figure 3 illustrates this. The top panel shows a non-deterministic policy on a one-dimensional continuous state-space with two possible actions. The middle panel shows a policy that is consistent with the NDP. Though it is consistent, this policy is a complex function of the 1D state and is difficult to justify if the state is a continuous patient measurement and the action is a treatment. Furthermore, there is *no Q-function* linear in the given feature space that produces this consistent policy as its greedy policy. In other words, given the feature space, there is *no scalar reward signal* that would cause us to learn this policy with fitted-Q and linear regression. We therefore will “prune away” these consistent but un-representable policies in order to reduce the size of $\mathcal{Q}_t$ by introducing the notion of *policy* ϕ*-consistency*.

Definition 3 (Policy ϕ-consistency)—*Given a feature map* $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^p$, we say a policy π_t *is* ϕ-consistent *with a non-deterministic policy* Π_t *over a data set with* n *trajectories, if and only if* $\exists \mathbf{w} (\forall i \in 1, \ldots, n \pi_t(s_t^i) \in \Pi_t(s_t^i) \wedge \pi_t(s_t^i) = \text{argmax}_a\, \phi(s_t^i, a)^\top \mathbf{w})$. *We write* $\pi_t \sqsubset_\phi \Pi_t$, *and we denote the set of all policies that are* ϕ*-consistent with* Π_t *by* $\mathcal{C}_\phi(\Pi_t)$.

A ϕ-consistent policy is an element of $\mathcal{C}(\Pi_t)$ that is the argmax policy for some (scalar) Q-function over the feature map ϕ. The form of such a policy is much like that of the function learned by a structured-output SVM (Tsochantaridis et al., 2005).

We now show that the number of ϕ-optimal policies for any given time point is polynomial in the data set size n.

Theorem 1—*Given a data set of size n, a feature map* ϕ, *and an action set* $\mathcal{A}$, *there are at most* $O(n^{\dim(\phi)} \cdot |\mathcal{A}|^{2 \dim(\phi)})$ *feature-consistent policies.*

<u>Proof:</u> The space of ϕ-consistent policies is exactly analogous to the space of linear multiclass predictors with ϕ as their feature map. We therefore port two results from learning theory to analyze the number of ϕ-consistent policies in terms of the dimension of ϕ, the size of the data set *n*, and the size of the action set. The *Natarajan dimension* (Natarajan, 1989; Shalev-Shwartz and Ben-David, 2014) is an extension of VC-dimension to the multiclass setting. For a supervised learning data set of size *n, k* classes, and a hypothesis class $\mathcal{H}$ with Natarajan dimension Ndim ($\mathcal{H}$), the number $|\mathcal{H}_n|$ of hypotheses restricted to the *n* datapoints is subject to the following upper bound due to Natarajan (1989):

$$|\mathscr{H}_n| \leqslant n^{\mathrm{Ndim}(\mathscr{H})} \cdot k^{2\mathrm{Ndim}(\mathscr{H})}. \quad (5)$$

Furthermore, the hypothesis class given by

$$\mathscr{H}_\phi = \{x \mapsto \underset{i}{\operatorname{argmax}}\ \phi(x,i)^\top \mathbf{w} : \mathbf{w} \in \mathbb{R}^{\dim \phi}\} \quad (6)$$

has Natarajan dimension Ndim($\mathcal{H}_\phi$) = dim(ϕ) (Shalev-Shwartz and Ben-David, 2014). Combining Equations (5) and (6) and completes our proof.

Theorem 1 shows that for fixed $|\mathscr{A}|$ and dim(ϕ) there are only polynomially many ϕ-consistent future policies, rather than a potentially exponential number of consistent policies as a function of *n*. Therefore, by considering only ϕ-consistent future policies, we can ensure that the size of $\mathscr{Q}_{T-1}$ is polynomial in *n*. The restriction to ϕ-consistent policies applies to *Q*-functions based on Generalized Linear Models with monotonic increasing link functions (such as logistic regression) as well. Such models have output of the form $g(\phi(s_t, a)^\top \mathbf{w})$ for monotonic increasing *g*. For these models, $\operatorname{argmax}_a g(\phi(s^i, a)^\top \mathbf{w}) = \operatorname{argmax}_a \phi(s_t, a)^\top \mathbf{w}$, so all of our results and algorithms for ϕ-consistency immediately apply.

We note that even if we prune using ϕ-consistency, the number of policies is exponential in dim ϕ, the feature space. Hence, this approach is tractable only for relatively simple *Q*-models. In this work we demonstrate that it is practical in a proof-of-concept setting (the CATIE study) but we acknowledge this limitation and defer it to future work.

We now express $\mathscr{C}_\phi(\Pi)$ in a way that allows us to enumerate it using a Mixed Integer Program (MIP). To formulate the constraints describing $\mathscr{C}_\phi(\Pi)$, we take advantage of *indicator constraints*, a mathematical programming formalism offered by modern solvers; e.g. the CPLEX optimization software package as of version 10.0, which was released in 2006 (CPLEX). Each indicator constraint is associated with a binary variable, and is only enforced when that variable takes the value 1. To construct the MIP, we introduce $n \times |\mathscr{A}|$ indicator variables $\alpha_{i,j}$ that indicate whether $\pi(s^i) = j$ or not. We then impose the following constraints:

$$\forall i \in 1,\ldots,n, j \in 1,\ldots,|\mathscr{A}|, \alpha_{i,j} \in \{0,1\} \quad (7)$$

$$\forall i \in 1,\dots,n, \sum_j \alpha_{i,j}=1 \quad (8)$$

$$\forall i \in 1,\dots,n, \forall j \in 1,\dots,|\mathscr{A}|, \alpha_{i,j}=1 \Rightarrow \forall k \neq j, (\phi(s^i,j)-\phi(s^i,k))^\top \mathbf{w} \geqslant 1. \quad (9)$$

Constraints (7) ensure that the indicator variables for the actions are binary. Constraints (8) ensure that, for each example in our data set, exactly one action indicator variable is on. The indicator constraints in (9) ensure that if the indicator for action j is on for the ith example, then weights must satisfy $j = \mathrm{argmax}_a\, \phi(s^i, a)^\top \mathbf{w}$. Note that the margin condition (i.e., having the constraint be $\geqslant 1$ rather than $\geqslant 0$) avoids a degenerate solution with $w = \mathbf{0}$.

The above constraints ensure that any feasible $\alpha_{i,j}$ define a policy that can be represented as an argmax of linear functions over the given feature space. Imposing the additional constraint that the policy defined is consistent with a given NDP Π is now trivial:

$$\forall i \in 1,\dots,n, \sum_{j\in\Pi(s^i)} \alpha_{i,j}=1. \quad (10)$$

Constraints (10) ensure that the indicator that turns on for the ith example in the data must be one that indicates an action that belongs to the set $\Pi(s^i)$.

Note that we have not specified an objective for this MIP: for the problem of generating ϕ-consistent policies, we are only interested in generating feasible solutions and interpreting the label variables as a potential future policy. Software such as CPLEX can enumerate all possible discrete feasible solutions to the constraints we have formulated. To do so, we give the constraints to the solver and ask for solutions given an objective that is identically zero. Note that if we instead minimized the quadratic objective $\|w\|^2$ subject to these constraints, we would recover the consistent policy with the largest margin between action choices in the feature space. The output would be equivalent to exact transductive learning of a hard-margin multiclass SVM using the actions as class labels (Tsochantaridis et al., 2005).

Given $\mathscr{Q}_t$, our final task is to define $\Pi_t(s_t)$ for all s_t. While $\mathscr{Q}_T$ is a singleton, for $t < T$ this is not the case in general, and we must take this into account when defining $\Pi_t(s_t)$. We present two definitions for $\Pi_t(s_t)$ based on a strict partial order $\prec$. (For example $\prec$ may be the Pareto partial order.)

$$\Pi_\prec^\forall(s_t)=\{a{:}\forall \hat{\mathbf{Q}} \in \mathscr{Q}_t(\nexists(a' \neq a, \hat{\mathbf{Q}}' \in \mathscr{Q}_t)(\hat{\mathbf{Q}}(s_t,a) \prec \hat{\mathbf{Q}}'(s_t,a')))\}$$

$$\Pi_\prec^\exists(s_t)=\{a{:}\exists \hat{\mathbf{Q}} \in \mathscr{Q}_t(\nexists(a' \neq a, \hat{\mathbf{Q}}' \in \mathscr{Q}_t)(\hat{\mathbf{Q}}(s_t,a) \prec \hat{\mathbf{Q}}'(s_t,a')))\}.$$

Algorithm 1

Non-deterministic fitted-Q

```
Learn $\hat{\mathbf{Q}}_T = (\hat{Q}_{T[1]}, \ldots, \hat{Q}_{T[D]})$, set $\mathscr{Q}_T = \{\hat{\mathbf{Q}}_T\}$
for $t = T - 1, T - 2, \ldots, 1$ do
  for all $s_t^i$ in the data do
    Generate $\Pi_{\prec}^{\exists}(s_t^i)$ using $\mathscr{Q}_{t+1}$
  $\mathscr{Q}_t \leftarrow \varnothing$
  for all $\pi_t \in \mathscr{C}_\phi(\Pi_{\prec}^{\exists})$ do
    for all $\hat{\mathbf{Q}}_{t+1} \in \mathscr{Q}_{t+1}$ do
      Learn $(\hat{Q}_{t[1]}(\cdot, \cdot, \pi_t, \ldots), \ldots, \hat{Q}_{t[D]}(\cdot, \cdot, \pi_t, \ldots))$ using $\hat{\mathbf{Q}}_{t+1}$, add to $\mathscr{Q}_t$
```

Under $\Pi_{\prec}^{\forall}$, action *a* is included if for ***all*** fixed sequences of policies we might follow after choosing *a*, no other choice of current action and future policy is preferable according to $\prec$. $\Pi_{\prec}^{\forall}$ is appealing in cases where we wish to guard against a naïve decision maker choosing poor sequences of future actions. For the $\mathscr{Q}_{T-1}$ shown in Figure 2, we would have $\Pi_{\prec}^{\forall}(s_{T-1}) = \{▼, ◀\}$. The ■ action is obviously eliminated because any ▼ point dominates every single ■ point. The ▲ and ▶ actions eliminate each other: There are ▲ points that are dominated by ▶ points, and ▶ points that are dominated by ▲ points. Note that this illustrates how $\Pi_{\prec}^{\forall}(s_t)$ could be empty: if our example only contained the ▶ and ▲ actions, we would have $\Pi_{\prec}^{\forall}(s_{T-1}) = \varnothing$. In practice we find that $\Pi_{\prec}^{\forall}$ can be very restrictive; we therefore present $\Pi_{\prec}^{\exists}$ as an alternative. Under $\Pi_{\prec}^{\exists}$, action *a* is included if there is *at least one* fixed future policy for which *a* is not dominated by a value achievable by another $(a', \hat{\mathbf{Q}}')$ pair. Note that $\Pi_{\prec}^{\exists} \supseteq \Pi_{\prec}^{\forall}$, and that because the relation $\hat{\mathbf{Q}} \prec \hat{\mathbf{Q}}'$ is a partial order on a finite set, there must exist at least one maximal element; therefore $\Pi_{\prec}^{\exists}(s_t) \neq \varnothing$. In the Figure 2 example, we have $\Pi_{\prec}^{\exists}(s_{T-1}) = \{▼, ◀, ▶, ▲\}$; note that ■ is not included because there is always another action that can dominate it if we choose an appropriate future policy. In order to provide increased choice and to ensure we do not generate NDPs with empty action sets, we will use $\Pi_{\prec}^{\exists}$ in our complete non-deterministic multiple-reward fitted-*Q* algorithm, but in our examples we will investigate the effect of choosing $\Pi_{\prec}^{\forall}$ instead.

5.4 Time Complexity

Pseudocode is given in Algorithm 1. The time cost of Algorithm 1 is dominated by the construction of $\mathscr{Q}_t$, whose size may increase by a factor of $O(n^{|A| \dim \phi})$ at each timestep; therefore in the worst case $|\mathscr{Q}_1|$ is exponential in *T*. This can be mitigated somewhat by pruning $\mathscr{Q}_t$ at each step, essentially removing from consideration future policy sequences that are dominated no matter what current action is chosen. Again, this pruning has no impact on solution quality because we are only eliminating future policy sequences that will never be executed. Despite the exponential dependence on *T*, we will show that our method can be successfully applied to real data in Section 7, and we defer the development of approximations to future work.

6. Practical domination

So far we have presented our algorithm assuming we will use Pareto dominance to define $\prec$. However, there are two ways in which Pareto dominance does not reflect the reasoning of a physician when she determines whether one action is superior to another. First, an action that has a *slightly* lower value along a single dimension, but is otherwise equivalent, will be Pareto-dominated (and eliminated) even if this difference is clinically meaningless. A physician with this knowledge would consider both actions in light of other "tie-breaking" factors not known to the RL policy, e.g., cost, allergies, etc. Second, an action that is slightly better for one reward but *much* worse for another would *not* be dominated, even though it may realistically be a very poor choice, and perhaps even unethical. Chatterjee et al. (2006) introduced ε-dominance which would partially address the first issue, but not the second. We wish to eliminate only actions that are "obviously" inferior while maintaining as much freedom of choice as possible. To accomplish this, we use the idea of *practical significance* (Kirk, 1996) to develop a definition of domination based on the idea that in real-world applications, small enough differences in expected reward simply do not matter. Differences that fall below a threshold of importance are termed "practically insignificant."

We introduce two notions of domination that are modifications of Pareto domination. The first, *Practical Domination*, most accurately describes our intuition about the set of actions that should be recommended. However, we show that it has an undesirable non-transitivity property. We then describe an alternative strategy based on what we call *Strong Practical Domination*.

Definition 4 (Practical Domination)

We say that an action a_2 is practically dominated *by a_1 at state s_T, and we write $a_2 \prec_p a_1$, if both of the following hold*

$$\forall d \in 1,\ldots,D \; Q_{t[d]}(s_T, a_2) \leqslant Q_{t[d]}(s_T, a_1) + \Delta_d, \quad (11)$$

$$\exists d \in 1,\ldots,D \; Q_{t[d]}(s_T, a_2) < Q_{t[d]}(s_T, a_1) - \Delta_d. \quad (12)$$

If either of the above do not hold, we write $a_2 \nprec_p a_1$.

Intuitively, an action a_1 practically dominates a_2 if a_2 is "not practically better" than a_1 for any basis reward (property 11), and if a_2 is "practically worse" than a_1 for *some* basis reward (property 12). "Practically better" and "practically worse" are determined by the elicited differences $\Delta_d \geqslant 0$. Note that we could have Δ_d depend on the current state if that were appropriate for the application at hand; for simplicity we assume a uniform Δ_d. We might consider using the relation $\prec_p$ as the ordering that produces our NDP according to one of the mappings from Section 5. Unfortunately, $\prec_p$ is not transitive. Suppose that the Q-vectors $(Q_{T[1]}(s_T, a), Q_{T[2]}(s_T, a))$ are (4.9, 4.9), (3, 5.2), (1.8, 5.6), (4.6, 4.6) for a_1, a_2, a_3, a_4, respectively, and suppose $\Delta_1 = \Delta_2 = 0.5$. Then $a_2 \prec_p a_1$ and $a_3 \prec_p a_2$ but $a_3 \not\prec_p a_1$. This non-transitivity causes undesirable behavior: if we consider only actions a_1 and a_3, we get $\Pi^{\exists}_{\prec}(s_T)=\{a_1, a_3\}$. However, if we consider a_1, a_2 and a_3, we get $\Pi^{\exists}_{\prec}(s_T)=\{a_1\}$! Thus by considering *more* actions, we get a *smaller* $\Pi^{\exists}_{\prec}(s_T)$. This is unacceptable in our domain, so we introduce an alternative.[2]

Definition 5 (Strong Practical Domination)

We say an action a_2 *is* strongly practically dominated *by* a_1 *at state* s_T*, and we write* $a_2 \prec_{sp}$ a$_1$, *if both of the following hold.*

$$\forall d \in 1, \ldots, D \; Q_{T[d]}(s_T, a_2) \leqslant Q_{T[d]}(s_T, a_1) \quad (13)$$

$$\exists d \in 1, \ldots, D \; Q_{T[d]}(s_T, a_2) < Q_{T[d]}(s_T, a_1) - \Delta_d \quad (14)$$

If either of the above do not hold, we write $a_2 \not\prec_{sp} a_1$.

The relation $\prec_{sp}$ is transitive, and will not cause the unintuitive results of $\prec_p$. However, it does not eliminate actions that are slightly better for one basis reward but much worse for another. (Note that $\exists d \in 1 \ldots D$, $Q_{T[d]}(s_T, a_2) > Q_{T[d]}(s_T, a_1) \Rightarrow a_2 \not\prec_{sp} a_1$.) We propose a compromise: we will use $\prec_{sp}$ as our partial order for producing NDPs as in Section 5. However, if an action a would have been eliminated according to $\prec_p$ but not according to $\prec_{sp}$, we may "warn" that it may be a bad choice. This has no impact on computation of Π and $\hat{Q}$ at earlier time points, but can warn the user that choosing a entails taking a practically significant loss on one basis reward to achieve a practically *in*significant gain on another.

7. Empirical Example: CATIE

We illustrate the output of non-deterministic fitted-Q using data from the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) study. The CATIE study was designed to compare sequences of antipsychotic drug treatments for the care of schizophrenia patients. The full study design is quite complex (Stroup and al, 2003; Swartz et al., 2003); we use a simplified subset of the CATIE data in order to more clearly illustrate the proposed methodology. CATIE was an 18-month study of $n = 1460$ patients that was divided into two main phases of treatment. Upon entry, most patients began "Phase 1," and were randomized to one of five treatments[3] with equal probability: olanzapine◀, risperidone ▶, quetiapine▲, ziprasidone▼, or perphenazine■. As time passed, patients were given the

[2] Note that for binary actions, the non-transitivity is not an issue and that this is common in some medical applications (e.g. treatment vs. watchful waiting, high-intensity vs. low-intensity treatment, etc.)

[3] Throughout the text we will suffix each treatment name with its corresponding plot-marker.

opportunity to discontinue their Phase 1 treatment and begin “Phase 2” on a new treatment. The possible Phase 2 treatments depended on the reason for discontinuing Phase 1 treatment. If the Phase 1 treatment was ineffective at reducing symptoms, then patients entered the “Efficacy” arm of Phase 2, and their Phase 2 treatment was chosen randomly as: {clozapine◆} with probability 1/2, or uniformly randomly from the set {olanzapine◀, risperidone▶, quetiapine▲} with probability 1/2. Because relatively few patients entered this arm, and because of the uneven action probabilities, it is reasonable to combine {olanzapine◀, risperidone▶, quetiapine▲} into one “not-clozapine” action, and we will do so here. If the Phase 1 treatment produced unacceptable side-effects, they entered the “Tolerability” arm of Phase 2, and their Phase 2 treatment was chosen uniformly randomly from {olanzapine◀, risperidone▶, quetiapine▲, ziprasidone▼}.

The goal of analyzing CATIE is to develop a two-time point policy ($T = 2$), choosing the intial treatment at $t = 1$ and possibly a follow-up treatment at $t = 2$. From a methodological perspective, the $t = 1$ policy is most interesting as it requires the computation of $\mathcal{Q}_1$ using Algorithm 1. Previous authors have used batch RL to analyze data from this study using a single basis reward (Shortreed et al., 2011) and examining convex combinations of basis rewards (Lizotte et al., 2012). In the following, we present the treatment recommendations of a non-deterministic fitted-Q analysis that considers both symptom relief and side-effects, and we compare with the output of a method by Lizotte et al. (2010, 2012). We begin by describing our basis rewards and our state spaces for $t = 2$ and $t = 1$, and we then present our results, paying particular attention to how much action choice is available using the different methods.

7.1 Basis Rewards

We will use ordinary least squares to learn Q functions for two basis rewards. For our first basis reward, we use the Positive and Negative Syndrome Scale (PANSS) which is a numerical representation of the severity of psychotic symptoms experienced by a patient (Kay et al., 1987). PANSS has been used in previous work on the CATIE study (Shortreed et al., 2011; Lizotte et al., 2012; Swartz et al., 2003), and is measured for each patient at the beginning of the study and at several times over the course of the study. Larger PANSS scores are worse, so for our first basis reward $r_{[1]}$ we use 100 minus the percentile of a patient’s PANSS at their exit from the study. We use the distribution of PANSS at intake as the reference distribution for the percentile.

For our second basis reward, we use Body Mass Index (BMI), a measure of obesity. Weight gain is an important and problematic side-effect of many antipsychotic drugs (Allison et al., 1999), and has been studied in the multiple-reward context (Lizotte et al., 2012). Because having a larger BMI is worse, for our second basis reward, $r_{[2]}$, we use 100 minus the percentile of a patient’s BMI at the end of the study, using the distribution of BMI at intake as the reference distribution.

7.2 State Space

For our state space, we use the patient's most recently recorded PANSS score, which experts consider for decision making (Shortreed et al., 2011). We also include their most recent BMI, and several baseline characteristics.

Because the patients who entered Phase 2 had different possible action sets based on whether they entered the Tolerability or Efficacy arm, we learn separate Q-functions for these two cases. The feature vectors we use for Stage 2 Efficacy patients are given by

$$\begin{aligned}\phi^{\text{EFF}}(s_2, a_2) = \; & [1,\ 1_{\text{TD}},\ 1_{\text{EX}},\ 1_{\text{ST1}},\ 1_{\text{ST2}},\ 1_{\text{ST3}},\ 1_{\text{ST4}},\ s_{2:\text{P}},\ s_{2:\text{B}},\\ & 1_{a_2=\blacklozenge},\ s_{2:\text{P}} \cdot 1_{a_2=\blacklozenge},\ s_{2:\text{B}} \cdot 1_{a_2=\blacklozenge}]^{\mathsf{T}}\end{aligned}$$

Here, $s_{2:\text{P}}$ and $s_{2:\text{B}}$ are the PANSS and BMI percentiles at entry to Phase 2, respectively. Feature $1_{a_2=\blacklozenge}$ indicates that the action at the second stage was clozapine$\blacklozenge$ and not one of the other treatments. We also have other features that do not influence the optimal action choice but that are chosen by experts to reduce variance in the value estimates.[4] 1_{TD} indicates whether the patient has had tardive dyskinesia (a motor-control side-effect), 1_{EX} indicates whether the patient has been recently hospitalized, and 1_{ST1} through 1_{ST4} indicate the "site type," which is the type of facility at which the patient is being treated (e.g. hospital, specialist clinic, etc.)

For Phase 2 patients in the Tolerability arm, the possible actions are $\mathcal{A}_2^{\text{TOL}} = \{\blacktriangleleft, \blacktriangle, \blacktriangleright, \blacktriangledown\}$, and the feature vectors we use are given by

$$\begin{aligned}\phi^{\text{TOL}}(s_2, a_2) = \; & [1,\ 1_{\text{TD}},\ 1_{\text{EX}},\ 1_{\text{ST1}},\ 1_{\text{ST2}},\ 1_{\text{ST3}},\ 1_{\text{ST4}},\ s_{2:\text{P}},\ s_{2:\text{B}},\\ & 1_{a_2=\blacktriangleleft},\ s_{2:\text{P}} \cdot 1_{a_2=\blacktriangleleft},\ s_{2:\text{B}} \cdot 1_{a_2=\blacktriangleleft},\ 1_{a_2=\blacktriangle},\ s_{2:\text{P}} \cdot 1_{a_2=\blacktriangle},\ s_{2:\text{B}} \cdot 1_{a_2=\blacktriangle},\\ & 1_{a_2=\blacktriangleright},\ s_{2:\text{P}} \cdot 1_{a_2=\blacktriangleright},\ s_{2:\text{B}} \cdot 1_{a_2=\blacktriangleright}]^{\mathsf{T}}\end{aligned}$$

Here we have three indicator features for different treatments at Phase 2, $1_{a_2=\blacktriangleleft}$, $1_{a_2=\blacktriangleright}$, $1_{a_2=\blacktriangle}$, with ziprasidone represented by turning all of these indicators off. Again we include the product of each of these indicators with the PANSS percentile s_2. The remainder of the features are the same as for the Phase 2 Efficacy patients.

For Phase 1 patients, the possible actions are $\mathscr{A}_1 = \{\blacktriangleleft, \blacksquare, \blacktriangle, \blacktriangleright, \blacktriangledown\}$, and the feature vectors we use are given by

$$\begin{aligned}\phi^{\text{EFF}}(s_2, a_2) = \; & [1,\ 1_{\text{TD}},\ 1_{\text{EX}},\ 1_{\text{ST1}},\ 1_{\text{ST2}},\ 1_{\text{ST3}},\ 1_{\text{ST4}},\ s_{1:\text{P}},\ s_{1:\text{B}},\\ & 1_{a_2=\blacktriangleleft},\ s_{1:\text{P}} \cdot 1_{a_2=\blacktriangleleft},\ s_{1:\text{B}} \cdot 1_{a_2=\blacktriangleleft},\ 1_{a_2=\blacksquare},\ s_{1:\text{P}} \cdot 1_{a_2=\blacksquare},\ s_{1:\text{B}} \cdot 1_{a_2=\blacksquare},\\ & 1_{a_2=\blacktriangle},\ s_{1:\text{P}} \cdot 1_{a_2=\blacktriangle},\ s_{1:\text{B}} \cdot 1_{a_2=\blacktriangle},\ 1_{a_2=\blacktriangleright},\ s_{1:\text{P}} \cdot 1_{a_2=\blacktriangleright},\ s_{1:\text{B}} \cdot 1_{a_2=\blacktriangleright}]^{\mathsf{T}}\end{aligned}$$

[4] See Section 4.2 of the paper by Shortreed et al. (2011) for an explanation of these kinds of features.

We have four indicator features for different treatments at Phase 2, $1_{a_1=◀}$, $1_{a_1=■}$, $1_{a_1=▲}$, and $1_{a_1=▶}$, with ziprasidone represented by turning all of these indicators off. We include the product of each of these indicators with the PANSS percentile s_1 at entry to the study, and the remainder of the features are the same as for the Phase 2 feature vectors. (These are collected before the study begins and are therefore available at Phase 1 as well.)

7.3 Results

The purpose of our empirical study is to demonstrate that our non-deterministic fitted-Q algorithm is feasible to use on real clinical trial data, and that it can offer increased choice over other approaches in a real-world setting. We will discuss several plots of different NDPs. Each point on a plot represents one value of s_1 in our data set, and at each point is placed a marker for each action recommended by an NDP[5]. To use the plots to make a decision for Phase 1, one would find the point on the plot corresponding to a current patient's state, and see what actions are recommended for that state. One would then decide among them using expert knowledge, knowing that according to the data and the chosen solution concept, any of those actions would be optimal. Then the process would be repeated should the patient move on to Phase 2, using the corresponding plots for $T = 2$ (not shown.) It is important to note that the axes in Figures 5 through 8 represent *state*, even though the same features (measured after treatment) are also used as reward values.

One can think of all of the learned NDPs that we present in the following experiments as transformations of the raw trajectory data into recommended actions, made under different solution concepts. The choice of solution concept is subjective and tied an application at hand; hence we will not argue that one result is necessarily "better" than another, but rather illustrate some of the differences between them. Indeed, the ability to accommodate different solution concepts is a strength of our approach. That said, we argue that if two solution concepts are both acceptable for a given application, we should prefer the one that offers more action choice to the decision-maker. Therefore, as we discuss the appropriateness of different solution concepts for the CATIE data, and we will examine how the amount of action choice varies for different solution concepts.

Figure 5 serves as our baseline. It shows the NDP at Phase 1 produced using the convex combination technique of Lizotte et al. (2012), which assumes a linear scalarization function (equivalent to the convex Pareto partial order) *and* assumes that preferences are fixed over time. One can see that for a large part of the state space, only ziprasidone▼ and olanzapine◀ are recommended. This occurs because for much of the state space, ziprasidone▼ and olanzapine◀ have Q values similar to those in Figure 2: olanzapine◀ performs better on PANSS than on BMI, and ziprasidone▼ has the opposite effect. These two treatments tend to eliminate the more "moderate" actions by the mechanism we described in Figure 1. In this NDP, the mean number of choices per state is 2.26, and 100% of states have had one or more actions eliminated.

[5] Note that Figure 2 is in fact a plot of the Q-function for Phase 1 at a state where (PANSS, BMI) = (50. 1, 48. 6), limited to a $\mathcal{Q}_t$ of size of 20 for clarity.

In our opinion, the convex Pareto domination solution criterion is overly eager to eliminate actions in this context, and the assumption of a fixed scalarization function is unrealistic. Figure 6 shows the NDP learned for Phase 1 using Algorithm 1 with Pareto domination and $\Pi^{\exists}_{\prec}$, which relaxes these two assumptions. As expected, the recommended action sets are larger. Despite the increased choice available, a user following these recommendations can still achieve a value on the Pareto frontier even if their preferences change in Phase 2. In this NDP, the mean number of choices per state is 4.14, and 68% of states have had at least one eliminated.

We now examine the actions that would be recommended if the decision-maker used the Strong Practical Domination solution concept. Figure 7 shows the NDP learned for Phase 1 using our algorithm with Strong Practical Domination ($\Delta_1 = \Delta_2 = 2.5$) and $\Pi^{\exists}_{\prec}$, and actions that receive a “warning” according to Practical Domination have been removed. In this example, choice is further increased by requiring an action to be practically better than another action in order to dominate it, and although we have removed actions that were warned to have a bad trade-off—those that were slightly better for one reward but practically worse for another—we still provide increased choice over using the Pareto frontier alone. In this NDP, the mean number of choices per state is 4.30, and 55% of states have had one or more actions eliminated.

We now consider using the same solution concept but the more strict $\Pi^{\forall}_{\prec}$ definition for constructing the NDP. Figure 8 shows the NDP learned for Phase 1 using our algorithm with Strong Practical Domination ($\Delta_1 = \Delta_2 = 2.5$) and $\Pi^{\forall}_{\prec}$. Again, an action must be practically better than another action in order to dominate it, which tends to increase action choices. However, recall that for $\Pi^{\forall}_{\prec}$ we only recommend actions that are not dominated by another action for any future policy. Hence, these actions are extremely “safe” in the sense that they achieve an expected value on the $\prec_{sp}$-frontier as long as the user selects from our recommended actions in the future. In this NDP, the mean number of choices per state is 2.56, and 100% of states have had one or more actions eliminated. Hence, we have a trade-off here: Relative to $\Pi^{\exists}_{\prec}$, this approach reduces choice, yet increases safety; whether or not this is preferable will depend on the application at hand. That said, using $\Pi^{\forall}_{\prec}$ in this way provides more choice than recommending actions based on convex Pareto optimality and a fixed future policy, while at the same time providing a guarantee that the recommended actions are safe choices even if preferences change.

Using ϕ-consistency to reduce the size of $\mathcal{Q}_t$ was critical for all of our analyses. In the Phase 2 Tolerability NDP there are over 10^{124} consistent policies but only 1213 ϕ-consistent policies, and in the Phase 2 Efficacy NDP there are 1048576 consistent policies but only 98 ϕ-consistent policies. Finding the ϕ-consistent policies took less than one minute on an Intel Core i7 at 3.4GHz using Python and CPLEX.

8. Discussion

Our overarching goal is to expand the toolbox of data analysts by developing new, useful methods for producing decision support systems in very challenging settings. To have maximum impact, decision support must appropriately take into account the sequential aspects of the problem at hand and at the same time acknowledge the fact that different decision makers have different preferences. Working toward this goal, we have presented a suite of novel ideas for learning non-deterministic policies for MDPs with multiple objectives. We gave a formulation of fitted-Q iteration for multiple basis rewards, we discussed ways of producing an NDP from a set $\mathcal{Q}_t$ of Q-functions that depend on different future policies, we introduced the idea of ϕ-consistent policies to control computational complexity, and we introduced "practical domination" to help users express their preference over actions without explicitly eliciting a preference over basis rewards. Finally, we showed using clinical trial data how our method could be used, and we showed that the NDPs we are able to learn offer more optimal action choice than previous approaches.

One of our next steps will be to augment the definition of practical dominance to incorporate our estimation uncertainty in the Q-values. We will also investigate more aggressive "pruning" of the $\mathcal{Q}_t$ to control computational complexity—one could even consider using a single consistent policy per timestep, for example, by adding a margin-based objective to the MIP as described in Section 3.

Rather than restrict ourselves by trying to identify a single "best approach" for all decision support systems, we have developed an algorithm that is modular: One could substitute another notion of domination for the ones we proposed if another notion is more appropriate for a given problem domain. Regardless of this choice, our algorithm will suggest sets of actions that are optimal in the sense we have described. For some applications, $\Pi^{\exists}_{\prec}$ may be appropriate; for other more conservative applications $\Pi^{\forall}_{\prec}$ may be the only responsible choice. Note that we are not dictating how the output from the NDP is used; one could imagine an interface that accepted patient state information and displayed richer information based on $\Pi^{\exists}_{\prec}, \Pi^{\forall}_{\prec}$, and perhaps plots like Figure 2 to convey to the user what the pros and cons are for the different actions. Our contributions make a wide variety of new decision support systems possible.

Acknowledgments

We acknowledge support from the Natural Sciences and Engineering Research Council of Canada. Data used in the preparation of this article were obtained from the limited access data sets distributed from the NIH-supported "Clinical Antipsychotic Trials of Intervention Effectiveness in Schizophrenia" (CATIE-Sz). The study was supported by NIMH Contract N01MH90001 to the University of North Carolina at Chapel Hill. The ClinicalTrials.gov identifier is NCT00014001. This manuscript reflects the views of the authors and may not reflect the opinions or views of the CATIE-Sz Study Investigators or the NIH.

References

Alagoz O, Hsu H, Schaefer AJ, Roberts MS. Markov decision processes: A tool for sequential decision making under uncertainty. Medical decision making : an international journal of the Society for Medical Decision Making. 2010; 30(4):474–483. ISSN 0272-989X. [PubMed: 20044582]

Allison DB, Mentore JL, Heo M, Chandler LP, Cappelleri JC, Infante MC, Weiden PJ. Antipsychotic-induced weight gain: A comprehensive research synthesis. American Journal of Psychiatry. 1999 Nov.156:1686–1696. [PubMed: 10553730]

Bertsekas, DP. Dynamic Programming and Optimal Control, Vol. II. 3rd. Athena Scientific; 2007. ISBN 1886529302, 9781886529304

Bertsekas, DP.; Tsitsiklis, JN. Neuro-Dynamic Programming. Vol. chapter 2.1. Athena Scientific; 1996. p. 12

Blatt, D.; Murphy, SA.; Zhu, J. Technical Report 04-63. The Methodology Center, Penn. State University; 2004. A-learning for approximate planning.

Brunskill E, Russell SJ. Partially observable sequential decision making for problem selection in an intelligent tutoring system. Educational Data Mining (EDM). 2011:327–328.

Burnside ES, Chhatwal J, Alagoz O. What Is the Optimal Threshold at Which to Recommend Breast Biopsy? PLoS ONE. 2012 Nov.7(11):e48820. ISSN 1932-6203. [PubMed: 23144986]

Castelletti A, Galelli S, Restelli M, Soncini-Sessa R. Tree-based reinforcement learning for optimal water reservoir operation. Water Resources Research. 2010:46.

Chatterjee K, Majumdar R, Henzinger T. Markov decision processes with multiple objectives. STACS. 2006:325.

Chi M, VanLehn K, Litman D, Jordan P. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. Int. J. Artif. Intell. Ed. 2011 Jan; 21(1–2):83–113. ISSN 1560-4292.

Cook, RD.; Weisberg, S. Applied Regression Including Computing and Graphics. Wiley; 1999 Aug.

CPLEX. ILOG CPLEX Optimizer. 2012 http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Ehrgott, M. Multicriteria Optimization. second. Vol. chapter 3. Springer; 2005.

Ernst D, Geurts P, Wehenkel L. Tree-Based Batch Mode Reinforcement Learning. Journal of Machine Learning Research. 2005; 6:503–556.

Hastie, T.; Tibshirani, R.; Friedman, J. Springer Series in Statistics. New York, NY, USA: Springer New York Inc; 2001. The Elements of Statistical Learning.

Henderson R, Ansell P, Alshibani D. Regret-regression for optimal dynamic treatment regimes. Biometrics. 2010; 66:1192–1201. [PubMed: 20002404]

Kay SR, Fiszbein A, Opfer LA. The Positive and Negative Syndrome Scale (PANSS) for schizophrenia. Schizophrenia Bulletin. 1987; 13(2):261–276. [PubMed: 3616518]

Kirk RE. Practical significance: A concept whose time has come. Educational and Psychological Measurement. 1996 Oct; 56(5):746–759.

Laber EB, Lizotte DJ, Ferguson B. Set-valued dynamic treatment regimes for competing outcomes. Biometrics. 2014a; 70(1):53–61. [PubMed: 24400912]

Laber EB, Lizotte DJ, Qian M, Pelham WE, Murphy SA. Dynamic treatment regimes: technical challenges and applications. Electronic Journal of Statistics. 2014b; 8(1):1225–1272. [PubMed: 25356091]

Lizotte, DJ.; Bowling, M.; Murphy, SA. Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis; International Conference on Machine Learning (ICML); 2010.

Lizotte DJ, Bowling M, Murphy SA. Linear fitted-Q iteration with multiple reward functions. Journal of Machine Learning Research. 2012 Nov.13:3253–3295. [PubMed: 23741197]

Miettinen, KM. Nonlinear Multiobjective Optimization. Kluwer; 1999.

Milani Fard M, Pineau J. Non-deterministic policies in Markovian decision processes. Journal of Artificial Intelligence Research. 2011; 40:1–24.

Moodie EEM, Richardson TS, Stephens DA. Demystifying optimal dynamic treatment regimes. Biometrics. 2007; 63(2):447–455. [PubMed: 17688497]

Natarajan BK. On learning sets and functions. Machine Learning. 1989; 4(1):67–97.

Perny, P.; Weng, P. On finding compromise solutions in multiobjective Markov decision processes; European Conference on Artificial Intelligence (ECAI); 2010. p. 969p. 970

Păduraru, C.; Precup, D.; Pineau, J.; Comănici, G. A study of off-policy learning in computational sustainability; European Workshop on Reinforcement Learning (EWRL), volume 24 of JMLR Workshop and Conference Proceedings; 2012. p. 89-102.

Rafferty, AN.; Brunskill, E.; Griffiths, TL.; Shafto, P. Faster teaching by POMDP planning. International Conference on Artificial Intelligence in Education (AIED); Springer-Verlag; Berlin, Heidelberg. 2011. p. 280-287.ISBN 978-3-642-21868-2

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

Shalev-Shwartz, S.; Ben-David, S. Understanding Machine Learning. Cambridge University Press; 2014. Cambridge Books Online

Shortreed S, Laber EB, Lizotte DJ, Stroup TS, Pineau J, Murphy SA. Informing sequential clinical decision-making through reinforcement learning: an empirical study. Machine Learning. 2011; 84(1–2):109–136. [PubMed: 21799585]

Strecher VJ, Shiffman S, West R. Moderators and mediators of a web-based computer-tailored smoking cessation program among nicotine patch users. Nicotine & tobacco research. 2006; 8(S. 1):S95. [PubMed: 17491176]

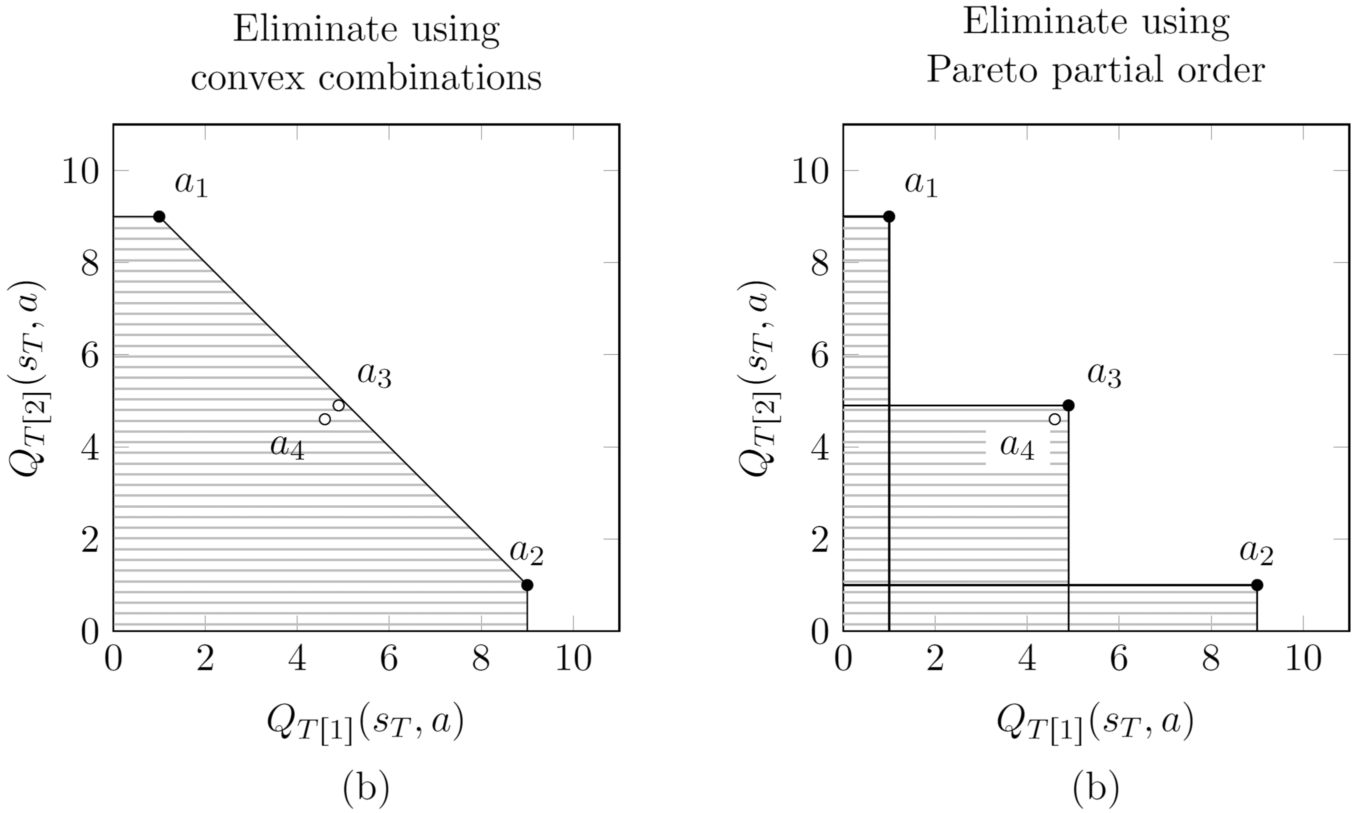
Stroup TS, et al. The national institute of mental health clinical antipsychotic trials of intervention effectiveness (CATIE) project: Schizophrenia trial design and protocol development. Schizophrenia Bulletin. 2003; 29(1)

Swartz MS, Perkins DO, Stroup TS, McEvoy JP, Nieri JM, Haal DD. Assessing clinical and functional outcomes in the clinical antipsychotic of intervention effectiveness (CATIE) schizophrenia trial. Schizophrenia Bulletin. 2003; 29(1)

Tsochantaridis I, Joachims T, Hoffmann T, Altun Y. Large margin methods for structured and independent output variables. Journal of Machine Learning Research. 2005; 6:1453–1484.

Vamplew, P.; Dazeley, R.; Barker, E.; Kelarev, A. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks; The 22nd Australasian Conf. on AI; 2009.

Vamplew P, Dazeley R, Berry A, Issabekov R, Dekker E. Empirical evaluation methods for multiobjective reinforcement learning algorithms. Machine Learning. 2011; 84:51–80.



Eliminate using convex combinations

Eliminate using Pareto partial order

(b)

(b)

Figure 1.

Comparison of existing approaches to eliminating actions at time T. The problems illustrated here have analogs for $t < T$ where the picture is more complicated. In this simple example, we suppose the vector-valued expected rewards ($Q_{T[1]}(s_T, a)$, $Q_{T[2]}(s_T, a)$) are (1, 9), (9, 1), (4.9, 4.9), (4.6, 4.6) for actions a_1, a_2, a_3, a_4, respectively. **Figure 1(a)**: Using the method of Lizotte et al. (2010, 2012) based on convex combinations of rewards, actions a_3 and a_4 would be eliminated, and we would have $\Pi_T(s_t) = \{a_1, a_2\}$. (Any action whose expected rewards fall in the shaded region would be eliminated.) However, we would prefer to at least include a_3 since it offers a more “moderate” outcome that may be important to some decision-makers. **Figure 1(b)**: Using the Pareto partial order, only action a_4 is eliminated, and we have $\Pi_T(s_T) = \{a_1, a_2, a_3\}$. However, we may prefer to include a_4 since its performance is very close to that of a_3, and may be preferable for reasons we cannot infer from our data—e.g. cost, or allergy to a_3.

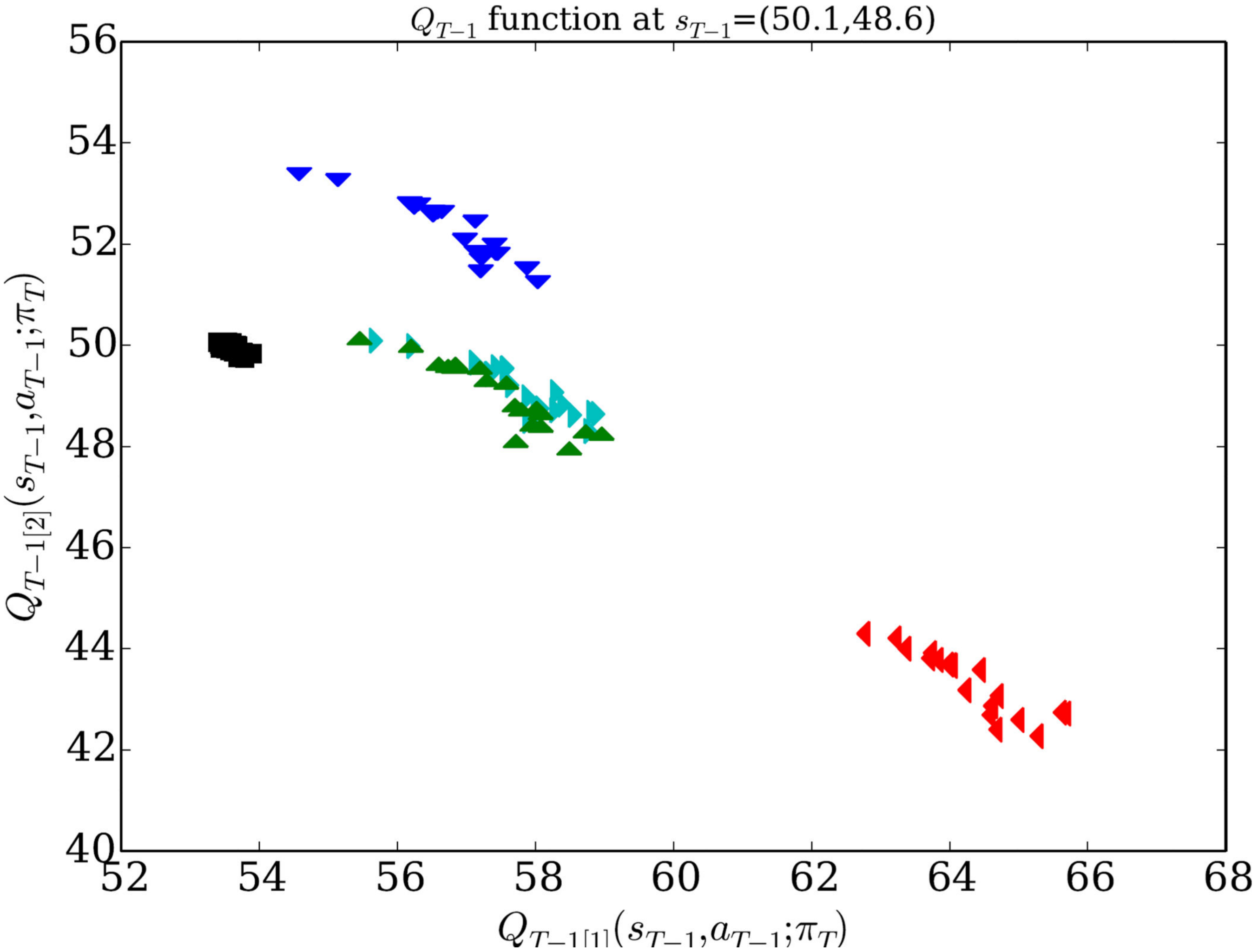


Figure 2.

Partial visualization of the members of an example $\mathscr{Q}_{T-1}$. We fix a state s_{T-1} = (50.1, 48.6) in this example, and we plot $\hat{\mathbf{Q}}_{T-1}(s_T, a_T)$ for each $\hat{\mathbf{Q}}_{T-1} \in \mathscr{Q}_{T-1}$ and for each $a_{T-1} \in$ {▼, ■, ▲, ▶, ◀}. For example, the ▼ markers near the top of the plot correspond to expected returns for each $\hat{\mathbf{Q}} \in \mathscr{Q}_T$ that is achievable by taking the ▼ action at the current time point and then following a particular future policy. This example $\mathscr{Q}_{T-1}$ contains 20 $\hat{\mathbf{Q}}_{T-1}$ functions, each assuming a different $\boldsymbol{\pi}_T$.

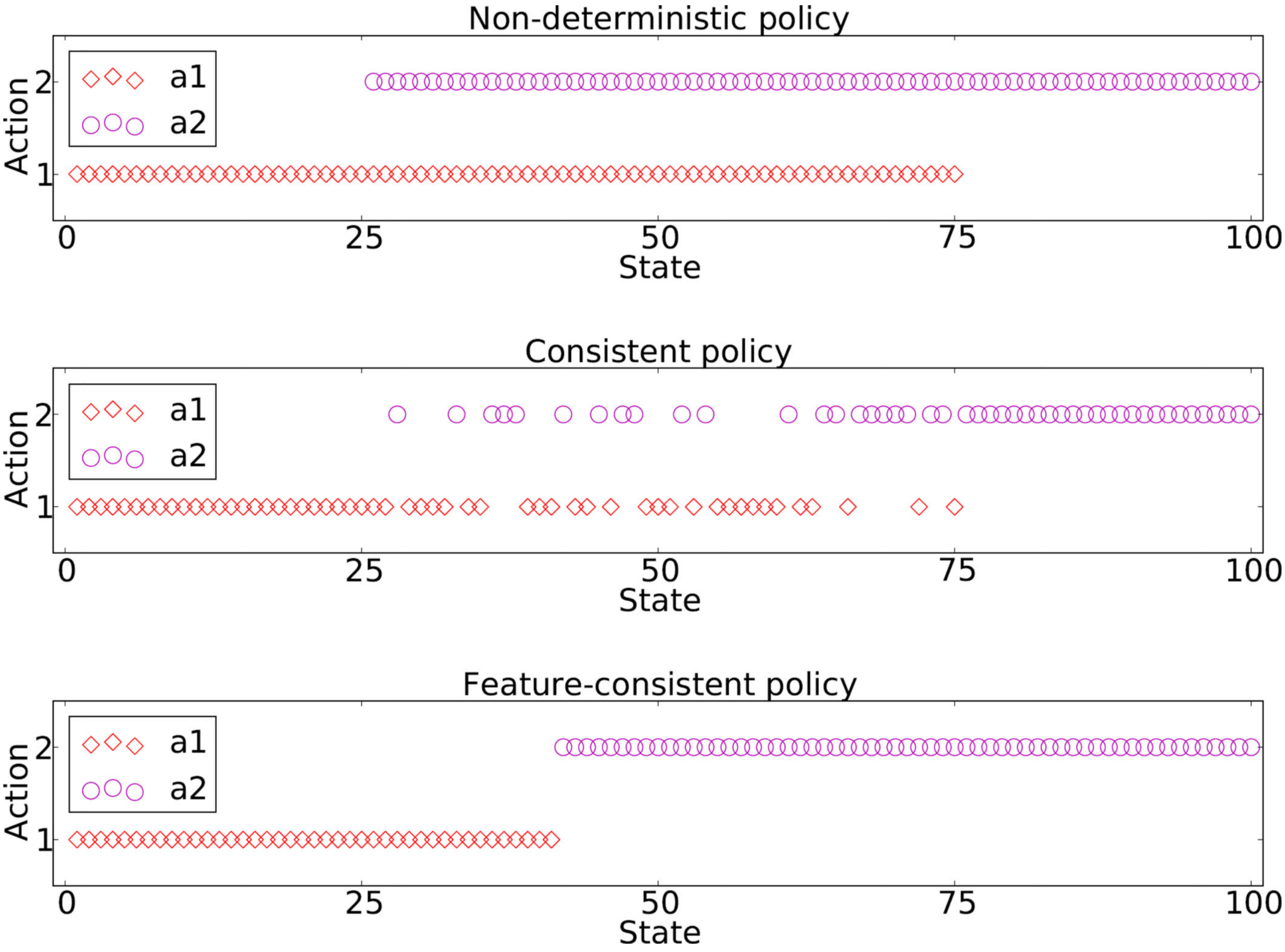


Figure 3.

An NDP on a one-dimensional continuous state-space, a consistent policy, and a ϕ-consistent policy.

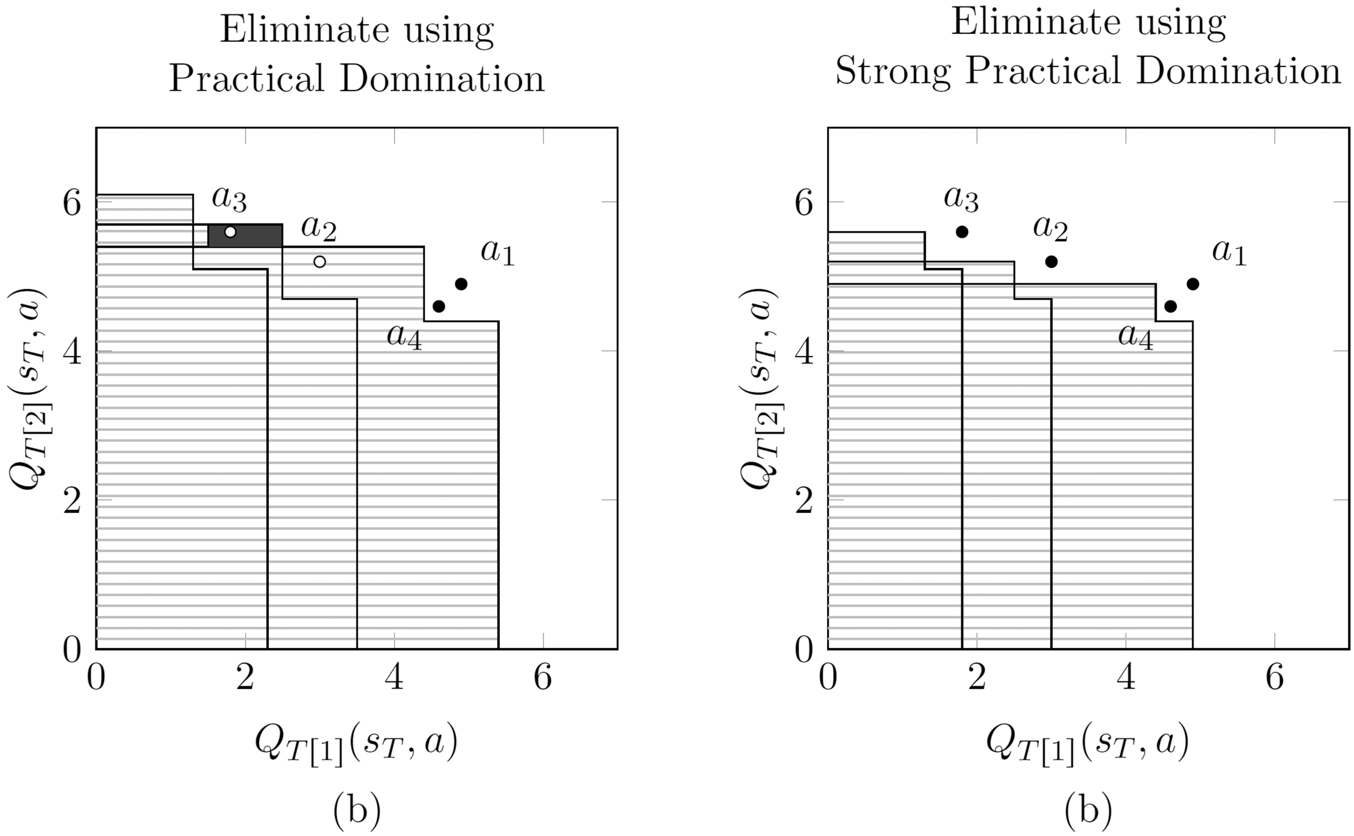


Figure 4.

Comparison of rules for eliminating actions. In this simple example, we suppose the Q-vectors ($Q_{T[1]}$ (s_T, a), $Q_{T[2]}$ (s_T, a)) are (4.9, 4.9), (3, 5.2), (1.8, 5.6), (4.6, 4.6) for a_1, a_2, a_3, a_4, respectively, and suppose $\Delta_1 = \Delta_2 = 0.5$. **Figure 4(a)**: Using the Practical Domination rule, action a_4 is not eliminated by a_3 because it is not much worse according to either basis reward, as judged by Δ_1 and Δ_2. Action a_2 is eliminated because although it is slightly better than a_1 according to basis reward 2, it is much worse according to basis reward 1. Similarly, a_3 is eliminated by a_2. Note the small solid rectangle to the left of a_2: points in this region (including a_3) are dominated by a_2, but not by a_1. This illustrates the non-transitivity of the Practical Domination relation, and in turn shows that it is not a partial order. **Figure 4(b)**: Using Strong Practical Domination, which is a partial order, no actions are eliminated, and there are no regions of non-transitivity.

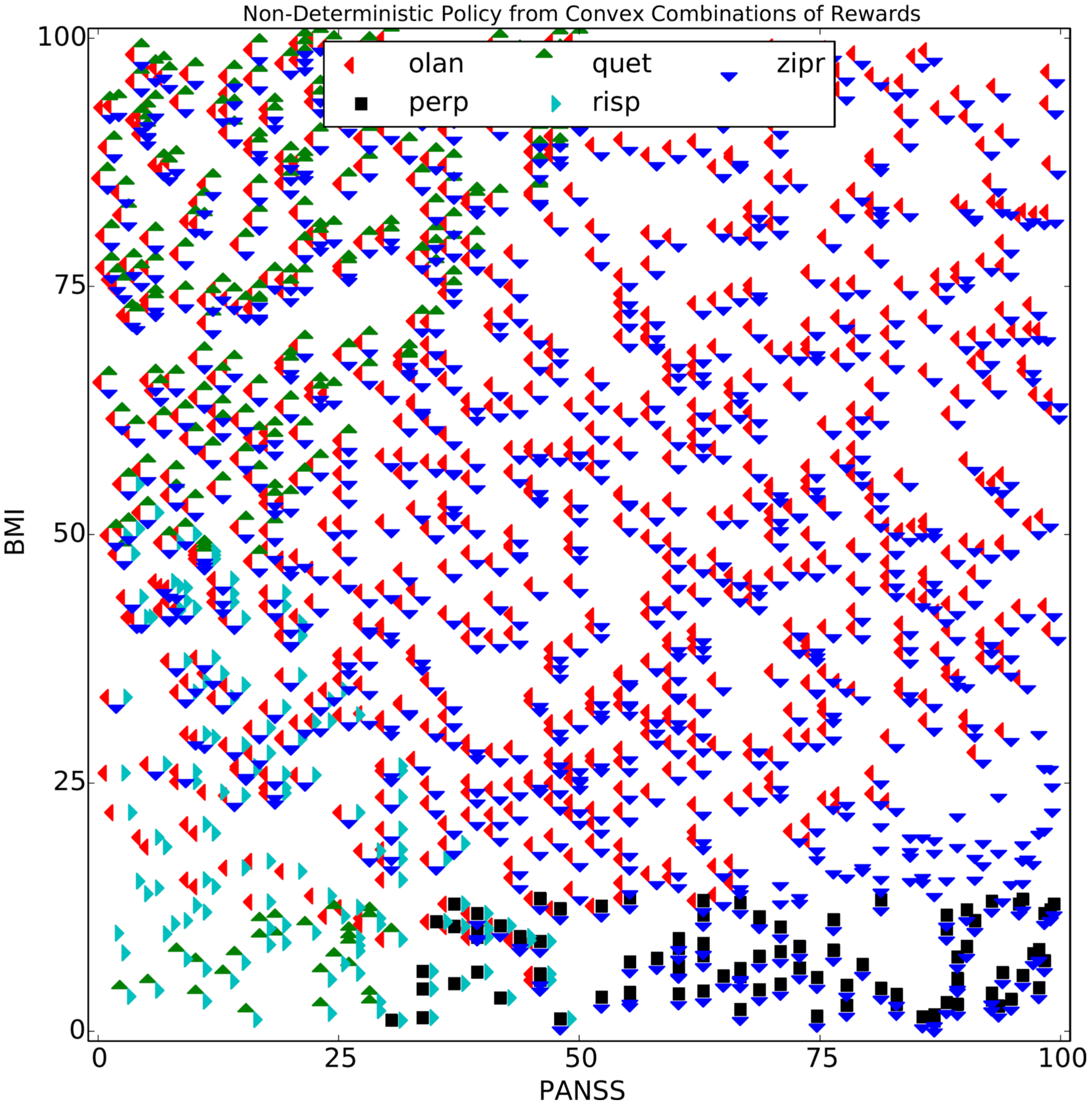


Figure 5.

NDP produced by taking the union over actions recommended by Lizotte et al. (2010, 2012)

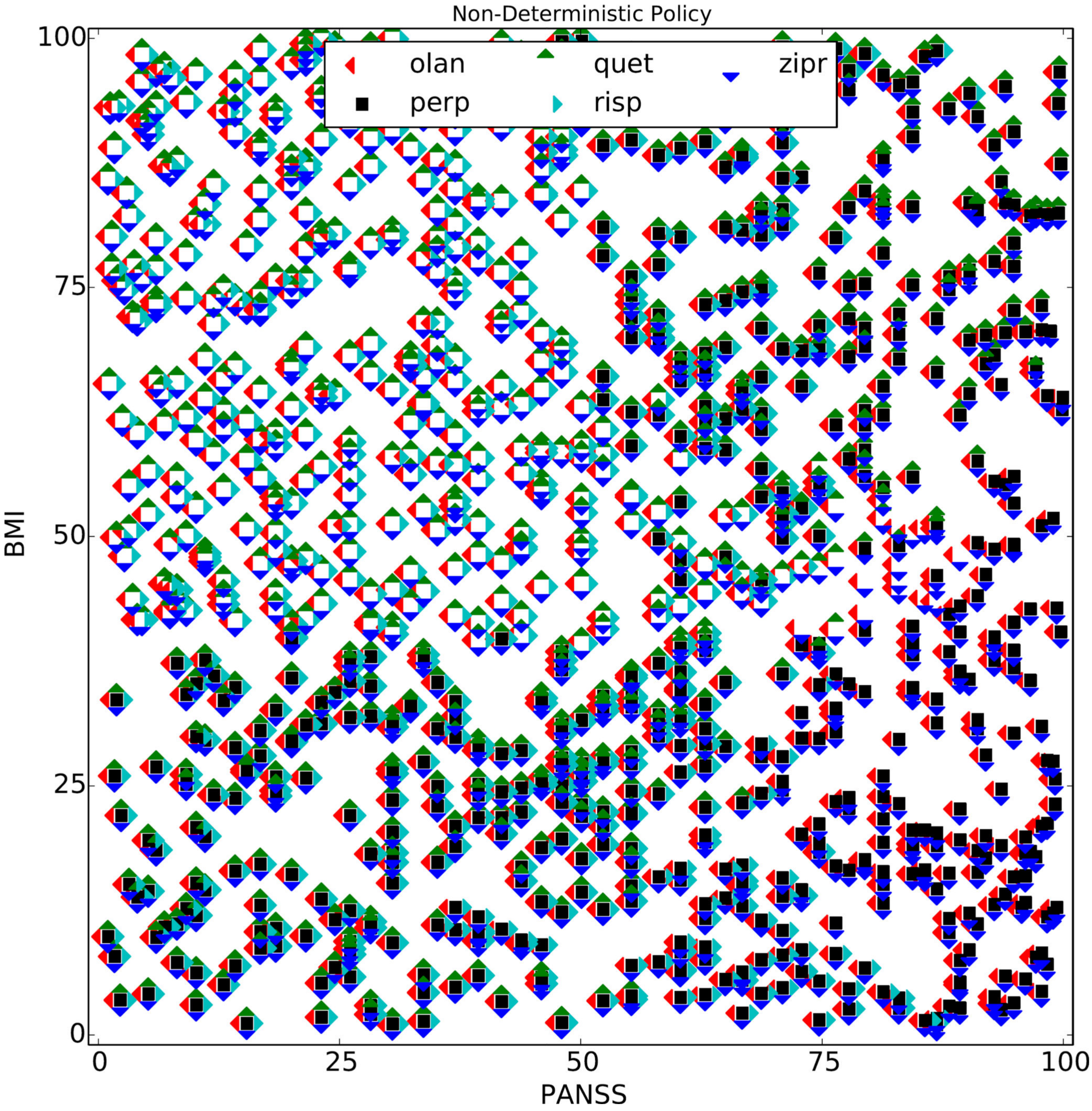


Figure 6.

NDP produced by $\Pi^{\exists}_{\prec}$ with Pareto Domination.

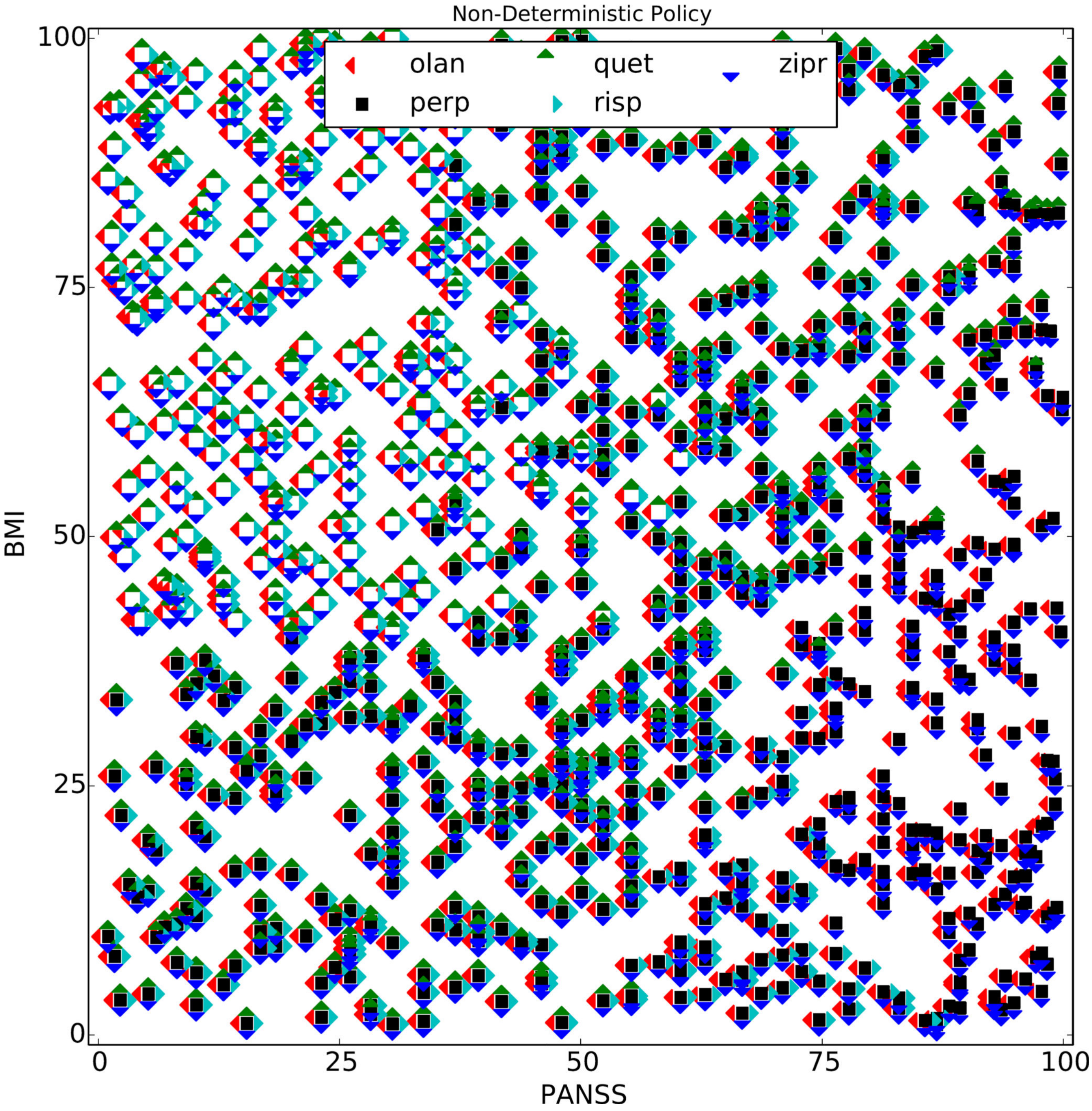


Figure 7.

CATIE NDP for Phase 1 made using $\Pi^{\exists}_{\precsim}$; “warning” actions that would have been eliminated by Practical Domination but not by Strong Practical Domination have been removed.

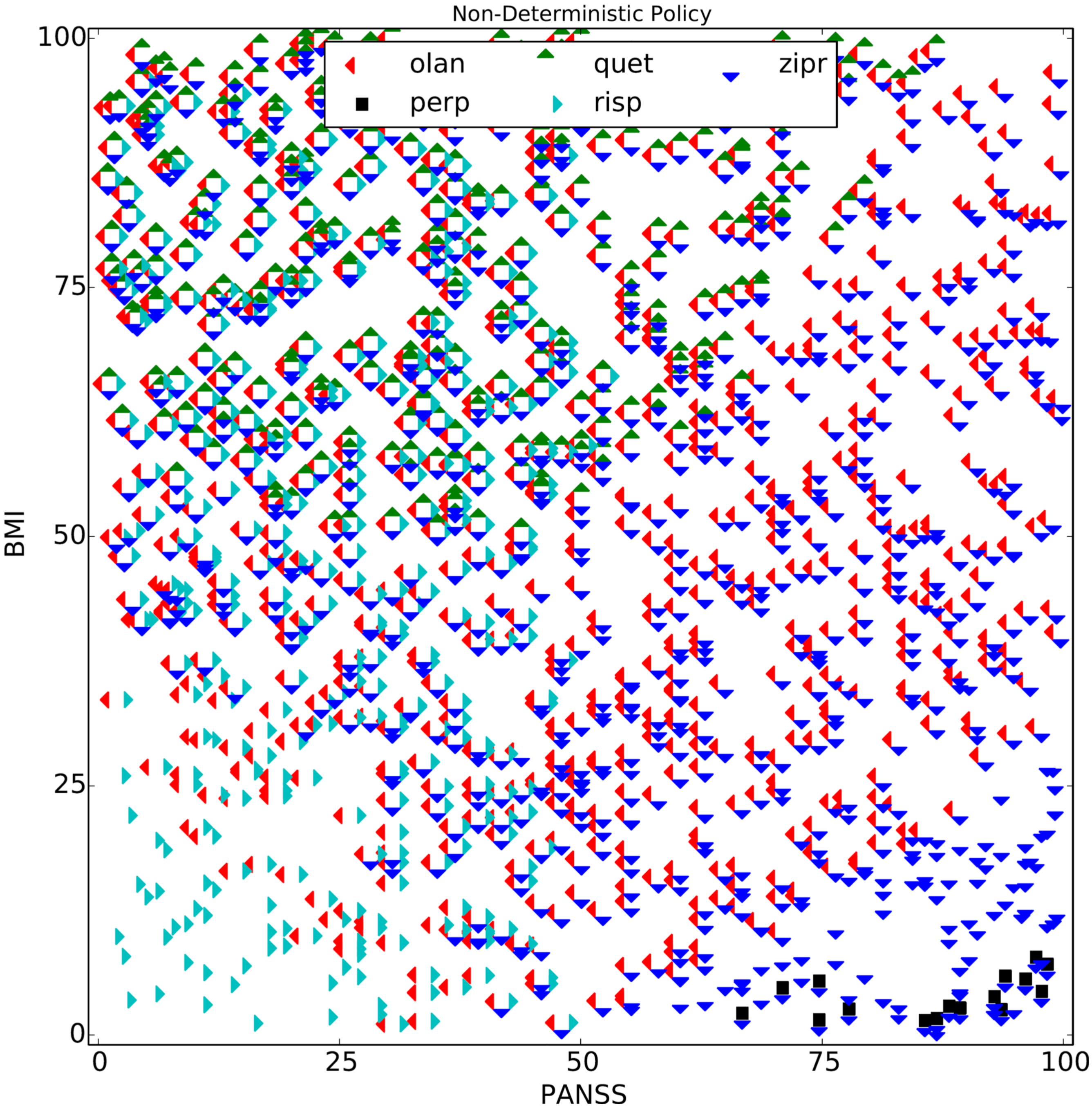


Figure 8.

NDP produced by $\Pi^{\forall}_{\prec}$ with Strong Practical Domination.