Quickest Change Detection Approach to Optimal Control in Markov Decision Processes with Model Changes

Taposh Banerjee[1], Miao Liu[2] and Jonathan P. How[3]

***Abstract*— Optimal control in non-stationary Markov decision processes (MDP) is a challenging problem. The aim in such a control problem is to maximize the long-term discounted reward when the transition dynamics or the reward function can change over time. When a prior knowledge of change statistics is available, the standard Bayesian approach to this problem is to reformulate it as a partially observable MDP (POMDP) and solve it using approximate POMDP solvers, which are typically computationally demanding. In this paper, the problem is analyzed through the viewpoint of quickest change detection (QCD), a set of tools for detecting a change in the distribution of a sequence of random variables. Current methods applying QCD to such problems only passively detect changes by following prescribed policies, without optimizing the choice of actions for long term performance. We demonstrate that ignoring the reward-detection trade-off can cause a significant loss in long term rewards, and propose a two threshold switching strategy to solve the issue. A non-Bayesian problem formulation is also proposed for scenarios where a Bayesian formulation cannot be defined. The performance of the proposed two threshold strategy is examined through numerical analysis on a non-stationary MDP task, and the strategy outperforms the state-of-the-art QCD methods in both Bayesian and non-Bayesian settings.**

I. Introduction and Motivation

In their life time, autonomous agents may be required to deal with non-stationary stochastic environments, where parameters of the world (dynamics, cost and goals) change over time (for the same states spaces). For example, in robotic navigation, a path planner might need to accommodate dynamically changing friction coefficients of road surfaces or wind conditions to ensure safety. In inventory control, an order management agent might need to account for the time varying distribution of demands for goods and services when determining the order at each decision period. A service robot may have to adapt to different personality or behavior of the person it is assisting. In a multi-agent system, an agent may have to adapt to changing strategies of the opponent or other players [1]. In order to maximize long term rewards, it is paramount that the agents are able to quickly detect the changes in the environment and adapt their policies accordingly.

When the environment is stationary, many of these problems for autonomous agents, path planning or robot control can be formulated as an MDP, for which there is a rich literature [2].When the environment is non-stationary, there is a limited understanding on the best strategy to employ. In this paper, we are interested in the case where the non-stationary process consists of several stationary processes. Each stationary process corresponds to an MDP, and the non-stationary nature is captured by changes in the transition and/or reward structure of the MDPs. That is, we consider the whole operation in a non-stationary environment as a global task, which can be decomposed into several stationary subtasks. This is a general case often encountered in real world applications, such as robotics, inventory controls, etc. For example, in inventory control, it is reasonable to assume that the demand rate would be steady for some time, before changing to another value. For path planning, one can assume that the road surfaces or wind conditions are steady for a while before changing, etc.

Solving dynamic decision making in non-stationary environments optimally is an extremely hard problem. Although it is extensively studied in the literature, due to the analytical intractability of the problem, however, existing works aim towards developing approximate solutions. For example, in some papers, the problem is reformulated as a Partially Observable MDP (POMDP), and approximate POMDP solvers are used to obtain a solution; e.g., see [3] and [4]. In others, estimates are maintained of the current MDP parameters, and next state or reward predictors are used to assess whether the parameters of the active MDP has changed; see e.g. [5] and [6]. In [7] and [8], new models such as hidden mode MDPs (hmMDPs) and mixed observability MDPs (MOMDPs) are proposed to capture the transition between different MDPs, and approximate solutions are obtained. Although these approximate solutions are promising tools for solving non-stationary MDPs, they are restricted to problems where the model sizes are known a priori, and they might still suffer from high computational overhead. The approach of multitask or transfer learning has also been used to study these problems. However, in these approaches, tasks are usually assumed to be well separated and given before learning and planning take place. There is no explicit concept of automatic change detection in these problems. Problems of this nature are also explored in the adaptive control literature. For example, in [9] and [10], the authors study optimal control of an MDP with unknown parameters in the transition kernel. Again, there is no concept of a change in this problem.

In this paper, we approach the problem of optimal control in MDPs with model changes through the viewpoint of classical quickest change detection (QCD) [11]. The changes in properties of MDPs (transition or reward) cause a change in the law of the state-action sequence. Tools and ideas

[1] Taposh Banerjee is with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. tbanerjee@seas.harvard.edu

[2] Miao Liu is with IBM T. J. Watson Research Center, Yorktown Heights, New York. miao.liu1@ibm.com

[3] Jonathan P. How is with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA. jhow@mit.edu

Work completed while the first two authors were at LIDS, MIT.

from the QCD literature can be used to detect these changes in law. QCD algorithms are scalable, and are optimized for quickest detection. In spite of these properties, these tools are either never used, or are not fully exploited in science and engineering literature, especially in the problem of interest in this paper. For example, sequential detection approaches are applied in [12] and [13], where the optimal policy for each MDP is executed, and a change detection algorithm is employed to detect model changes. We will show in this paper that such a naive approach may lead to significant loss in performance. The key idea is that what is optimal for optimizing rewards may not necessarily be optimal for quickest detection of model changes. Thus, there is a fundamental reward-detection trade-off to be exploited. The purpose of this paper is to articulate this trade-off in a mathematically precise manner, and to propose solutions that we claim best exploit this trade-off. Specifically, we propose a computationally efficient, simple two-threshold strategy to quickly detect model changes, without significant loss in rewards. We show that such an approach is superior to the existing methods in the literature. Specifically, we show that the two-threshold strategy leads to better performance as compared to the single threshold classical QCD tests, as employed in [13]. Note that it is shown in [13] that a QCD approach is better (with respect to the Bayesian problem we use in this paper; see (2) below) than other approaches in the literature, e.g., [6]. For benchmarking purposes, we also compare with the MOMDP solution [8] and a random action strategy.

II. Problem Formulation

Assuming we have a family of MDPs $\{\mathbf{M}_\theta\}$, where θ takes value in some index set Θ, which could be finite or infinite. For each θ, an MDP is defined by a four tuple: $\mathbf{M}_\theta = (\mathscr{S}, \mathscr{A}, \mathscr{T}_\theta, \mathscr{R}_\theta)$, where $\mathscr{S}$ and $\mathscr{A}$ are respectively the state space and the action space common to all MDPs, $\mathscr{T}_\theta$ is the transition kernel, and $\mathscr{R}_\theta$ is the reward function. A decision maker observes a sequence of states $\{s_k\}_{k\geq 0}$, $s_k \in \mathscr{S}, \forall k$, and for each observed state s_k, it chooses an action $a_k \in \mathscr{A}$. For each pair (s_k, a_k), the next state s_{k+1} takes values according to the law dictated by the kernel: $\forall k$

$$\mathscr{T}_\theta(s,a,s') := \mathbb{P}(s_{k+1} = s' \mid a_k = a, s_k = s),\ s, s' \in \mathscr{S}, a \in \mathscr{A}.$$

The reward obtained by choosing action a_k after observing s_k is given by $\mathscr{R}_\theta(s_k, a_k)$.

We operate in a non-stationary environment, which means that the transition structure or the reward structure can change over time. Specifically, at time say γ_1, the parameter θ of the MDP changes from $\theta = \theta_0$ to say $\theta = \theta_1$. At a later time, say γ_2, the parameter changes from $\theta = \theta_1$ to $\theta = \theta_2$, and so on. To be precise, denote the non-stationary dynamics for $k \geq 0$ as

$$\begin{aligned} &\mathbb{P}(s_{k+1} = s' \mid a_k = a, s_k = s) \\ &\qquad = \begin{cases} \mathscr{T}_{\theta_0}(s,a,s'), & \text{for } k < \gamma_1 \\ \mathscr{T}_{\theta_1}(s,a,s'), & \text{for } \gamma_1 \leq k < \gamma_2 \end{cases} \\ &\text{Reward for } (s_k,a_k) := R_k(s_k,a_k) \\ &\qquad = \begin{cases} \mathscr{R}_{\theta_0}(s_k,a_k), & \text{for } k < \gamma_1 \\ \mathscr{R}_{\theta_1}(s_k,a_k), & \text{for } \gamma_1 \leq k < \gamma_2. \end{cases} \end{aligned} \tag{1}$$

For simplicity of exposition, for now we restrict our attention to the case where there is only one change point γ, and there are only two models, a pre-change model $\mathbf{M}_0 = (\mathscr{S}, \mathscr{A}, \mathscr{T}_0, \mathscr{R}_0)$ and a post-change model $\mathbf{M}_1 = (\mathscr{S}, \mathscr{A}, \mathscr{T}_1, \mathscr{R}_1)$. In Section VI, we discuss extensions of the ideas developed for the two model case, to the case where there are more than two models (possibly infinite), and possibly more than one change point.

We consider two different problem formulations: Bayesian and Non-Bayesian.

A. Bayesian Formulation

Suppose we have a prior on the single change point $\gamma = \gamma_1$, and we have chosen a policy $\Pi = (\pi_1, \pi_2, \cdots)$ that maps the states $\{s_k\}$ to actions $\{a_k\}$. The MDPs $\mathbf{M}_0$ and $\mathbf{M}_1$, the policy Π, and the prior on the change point, together induce a joint distribution on the product space of state spaces and actions which we denote by $\mathbb{P}$, and the corresponding expectation by $\mathbb{E}$. The objective is to choose the policy Π so as to maximize a long term reward.

$$\max_{\Pi}\ \mathbb{E}\left[\textstyle\sum_{k=0}^{\infty} \beta^k R_k(s_k, a_k)\right], \tag{2}$$

where $\beta \in [0,1)$ is a discount factor. Note that the change point γ is unobservable. As a result, a Markov policy may not be optimal, and the policy Π above has to use the past history to choose the optimal action. To accommodate model changes, problem (2) can be reformulated as a Partially Observable MDP (POMDP) (see Lemma 1 below), and can be solved using approximate POMDP solvers. However, it is computationally expensive to solve this problem numerically, especially when the number of possible models is large. We thus use other approaches to obtain approximate solutions. The vehicle we have chosen to obtain the approximate solutions is the theory of QCD. We show in Section VII that our proposed solutions performs as well as the solution obtained from standard approximate POMDP solvers in the literature, while at the same time it incurs significantly less computational cost, online and off-line.

Lemma 1: The non-stationary MDP problem (2) is equivalent to a POMDP problem with the model specified as $\mathbf{M}_{POMDP}(\mathscr{X}, \mathscr{A}, \mathscr{O}, \widetilde{\mathscr{T}}, \Omega, \widetilde{\mathscr{R}})$, where $\mathscr{X} = \mathscr{S} \times \Theta$ is the POMDP state space, which concatenates the fully observable MDP state space $\mathscr{S}$ and the unobservable model index space Θ; $\mathscr{O}$ is the set of observations, which is equivalent to the set of MDP states; $\widetilde{\mathscr{T}} : X \times A \times X \to [0,1]$ is the POMDP state transition function; $\Omega : \mathscr{X} \times \mathscr{A} \times \mathscr{O} \to [0,1]$ is the observation function; $\widetilde{\mathscr{R}} : \mathscr{X} \times \mathscr{A} \to \mathbb{R}$ is the reward function with

$$\begin{aligned} \widetilde{\mathscr{T}}\big(x' = (s',\theta')|x = (s,\theta), a\big) &= \mathscr{T}_{\theta'}(s,a,s')\mathscr{F}(\theta'|\theta)\mathbb{I}(\theta' \neq \theta) \\ &\quad + \mathscr{T}_{\theta}(s,a,s')\mathscr{F}(\theta'|\theta)\mathbb{I}(\theta' = \theta) \\ \omega(o|x',a) &= \mathbb{I}(o = s') \\ \tilde{\mathscr{R}}(x,a) &= \mathscr{R}_\theta(s,a) \end{aligned} \tag{3}$$

where $\mathbb{I}(\cdot)$ is an indicator function, and $\mathscr{F}$ is a model transition matrix. Given the prior over a change point to be geometric distribution, i.e.,$\gamma \sim Geom(\lambda)$, it can be shown that

$$\mathscr{F}(\theta'|\theta) = \begin{cases} \lambda \mathbb{I}(\theta' \neq \theta) + (1-\lambda)\mathbb{I}(\theta' = \theta) & \text{if } \theta = 0 \\ \mathbb{I}(\theta' \neq \theta) & \text{if } \theta = 1 \end{cases} . \tag{4}$$

By converting the original non-stationary MDP problem into a POMDP problem, one can apply any existing POMDP solvers to obtain an approximate POMDP policy, which maps the belief state $b(x)$ to an action a. Considering the fact that here the state space can be factorized into an observable part and an unobservable part, this POMDP model can be treated as a mixed observable MDP (MOMDP). In an MOMDP, the belief state becomes a union of $|S|$ disjoint $|\Theta|$-dimension subspaces, and all the operations are performed in the lower dimension, hence is more time efficient to solve a MOMDP than to solve a POMDP.

B. Non-Bayesian Formulation

In practice, a prior on the change point is not often known. Thus, a Bayesian formulation cannot be defined. If the change point is treated as an unknown constant, there are infinite possible laws on the state-action joint spaces, one for each possible change point. In statistics, there are two ways to study non-Bayesian problems: minimax or maxmin and Neyman-Pearson type criteria.

For minimax or maxmin criterion we may consider the following problem:

$$\max_{\Pi} \min_{\gamma} \mathbb{E}_{\gamma} \left[\sum_{k=0}^{\infty} \beta^k R_k(s_k, a_k) \right],$$

where $\mathbb{E}_{\gamma}$ is the expectation with respect to the probability measure when change occurs at time γ. However, there are two major issues with such an approach in practice. First, a minimax approach may lead to a conservative or pessimistic approach to policy design. Secondly, it is not always straightforward to compute such costs for comparing various competing algorithms.

Since the problem under consideration is about identifying the correct regime, we also consider a Neyman-Pearson type of criterion: trying to maximize reward in one regime subject to a constraint on the performance under alternative regime. We take this approach (in addition to the Bayesian formulation) in our paper and consider the following problem.

$$\begin{aligned} \max_{\Pi} \quad & \mathbb{E}_1 \left[\sum_{k=0}^{\infty} \beta^k R_k(s_k, a_k) \right], \\ \text{subj. to} \quad & \mathbb{E}_{\infty} \left[\sum_{k=0}^{\infty} \beta^k R_k(s_k, a_k) \right] \geq \alpha. \end{aligned} \tag{5}$$

Thus, the objective is to optimize rewards when the change occurs at time 1, subject to a constraint on the performance when the change never occurs. In sequential analysis literature, one often uses a minimax objective rather than the average under $\mathbb{E}_1$. As mentioned earlier, this leads to a pessimistic viewpoint. Also, for many sequential detection algorithms in the literature, the worst case is often achieved for $\gamma = 1$. Thus, $\mathbb{E}_1 \left[\sum_{k=0}^{\infty} \beta^k R_k(s_k, a_k) \right]$ can also be seen as a proxy for the minimax or maxmin reward.

Another advantage of the Neyman-Pearson type criterion is that it maintains a balance in reward structure between the pre- and post-change regime. A discounted cost criterion with a discount factor $\beta < 1$, as chosen in Bayesian and maxmin problems, may not always be appropriate for the application in hand. This is because such a criterion penalizes under-performing under $\mathbf{M}_0$ more than it penalizes under-performing under $\mathbf{M}_1$. In reality, we may be interested in maintaining good performance under every regime.

C. Oracle Policy

Let $\mathbf{V}_0(x)$ and $\mathbf{V}_1(x)$ be the cost/reward to go at state x for the MDP models $\mathbf{M}_0$ and $\mathbf{M}_1$, respectively. Suppose the following conditions are satisfied:

1) the change point is exactly known,
2) the change point $\gamma >> 1$,
3) for some small $\delta > 0$, the function $|\mathbf{V}_1(x) - \mathbf{V}_1(y)| \leq \delta$, $\forall x, y$, that is the reward to go for the model $\mathbf{M}_1$ is not sensitive to the initial state x.

The implication of the first two conditions is that it is approximately optimal to use the optimal policy for $\mathbf{M}_0$ before the change point. The implication of the last condition is that, no matter which policy is used for model $\mathbf{M}_0$, it is still approximately optimal to use the optimal policy for model $\mathbf{M}_1$ after the change point. As a result, if all three of these conditions are satisfied, then one can just implement the optimal policy for the model $\mathbf{M}_0$ before change and use the optimal policy for the model $\mathbf{M}_1$ after change, and achieve close to maximum average rewards. In the following, such a policy is referred to as the *Oracle*.

Since the change point is unknown, it is clear that if the change can be reliably detected, then one can hope to achieve rewards close to what the Oracle can achieve. There are powerful algorithms in the literature that one can use to detect this change as quickly as possible. We briefly discuss relevant ideas and algorithms in the next section, which serves as a background to readers not familiar with the literature on quickest change detection.

III. Background on Quickest Change Detection

In the quickest change detection literature, algorithms are developed that allows one to quickly detect a change in the distribution of a stochastic process from one law to another. The optimality properties of the algorithms are studied under various modeling assumptions and problem formulations. The theoretical foundations were laid by the work of Wald [14] and Shiryaev [15]; see also [11] for a survey of the area. We discuss three such algorithms in the context of problems (2) and (5).

Suppose we employ a certain policy Π and observe a sequence of states and actions (driven by the policy) $\{(s_k, a_k)\}$. At the change point Γ, modeled as a random variable with prior probability mass function ϕ, the model changes from $\mathbf{M}_0$ to $\mathbf{M}_1$. Thus, the law of the stochastic sequence $\{(s_k, a_k)\}$ changes. Then the following algorithm, called the Shiryaev algorithm [16], has strong optimality properties. The Shiryaev algorithm dictates that we compute the following statistic at each time n:

$$S_n = \sum_{1 \leq k \leq n} \phi(k) \prod_{i=k}^{n} \frac{\mathscr{T}_1(s_{i-1}, a_{i-1}, s_i)}{\mathscr{T}_0(s_{i-1}, a_{i-1}, s_i)}, \tag{6}$$

and declare that a change in the model has occurred at the stopping time

$$\tau_s = \min\{n \geq 1 : S_n > A\}. \tag{7}$$

Here, A is a threshold chosen to control false alarms. If the prior on the change point is geometric with parameter ρ, then the statistic S_n can be computed recursively:

$$S_n = \frac{1 + S_{n-1}}{1-\rho} \frac{\mathscr{T}_1(s_{n-1}, a_{n-1}, s_n)}{\mathscr{T}_0(s_{n-1}, a_{n-1}, s_n)}; \quad S_0 = 0. \tag{8}$$

It is shown in [16] that if $A = \frac{1-\zeta}{\zeta}$ then

$$\mathbb{E}[(\tau_s - \Gamma)^+] \sim \min_{\tau : \mathbb{P}(\tau \leq \Gamma) \leq \zeta} \mathbb{E}[(\tau - \Gamma)^+] \sim \frac{|\log \zeta|}{I_\pi + d} \text{ as } \zeta \to 0. \tag{9}$$

Thus, the Shiryaev algorithm achieves the best average detection delay over all stopping times that satisfied a given probability of false alarm constraint of ζ, as $\zeta \to 0$. Here, d is a constant that is a function of the prior ϕ and equals $|\log(1-\rho)|$ for $\Gamma \sim \text{Geom}(\rho)$. The quantity I_π is the Kullback-Leibler information number defined by

$$I_\pi = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \log\left(\frac{\mathscr{T}_1(s_{k-1}, a_{k-1}, s_k)}{\mathscr{T}_0(s_{k-1}, a_{k-1}, s_k)}\right) \text{ under } \mathbb{P}_1. \tag{10}$$

Thus, larger this number I_π is, the smaller is the detection delay. We use the subscript π to emphasize that the information number is a function of the policy Π employed.

If the prior on the change point is not known, or if a prior cannot be defined, one can replace the Shiryaev statistic by the CUmulative SUM statistic (CUSUM) [17]

$$W_n = \max_{n-m \leq k \leq n} \sum_{i=k}^{n} \log \frac{\mathscr{T}_1(s_{i-1}, a_{i-1}, s_i)}{\mathscr{T}_0(s_{i-1}, a_{i-1}, s_i)}, \tag{11}$$

where m is a window size that depends on the false alarm constraint. The CUSUM algorithm has strong optimality properties, similar to (9), but with average delay and probability of false alarm expressions replaced by suitable minimax delay and mean time to false alarm expressions [17]. Specifically, if

$$\tau_c = \min\{n \geq 1 : W_n > A\}, \tag{12}$$

then, with $A = \log \eta$,

$$\begin{aligned} \sup_\gamma \mathbb{E}_\gamma[\tau_c - \gamma | \tau_c > \gamma] &\sim \min_{\tau : \mathbb{E}_\infty[\tau] \geq \eta} \sup_\gamma \mathbb{E}_\gamma[\tau - \gamma | \tau > \gamma] \\ &\sim \frac{|\log \eta|}{I_\pi} \text{ as } \eta \to \infty. \end{aligned} \tag{13}$$

Here, $\mathbb{E}_\gamma$ is the expectation when the change point is at γ, and $\mathbb{E}_\infty$ is the expectation when the change never occurs. Thus, the CUSUM algorithm minimizes the maximum of detection delay–maximum over all possible change points–subject to a constraint η on the mean time to false alarm, as $\eta \to \infty$.

Again, the optimal performance, for a given false alarm, depends on the information number I_π. Thus, the number I_π is the fundamental quantity of interest in quickest change detection problems. It will also play a fundamental role in the rest of the paper.

An alternative to CUSUM in the non-Bayesian setting is to put $\rho = 0$ in the Shiryaev algorithm and obtain what is called the Shiryaev-Roberts (SR) statistic

$$SR_n = (1 + SR_{n-1}) \frac{\mathscr{T}_1(s_{n-1}, a_{n-1}, s_n)}{\mathscr{T}_0(s_{n-1}, a_{n-1}, s_n)}; \quad SR_0 = 0, \tag{14}$$

and use the stopping time

$$\tau_{sr} = \min\{n \geq 1 : SR_n > A\}. \tag{15}$$

In the numerical results reported in Section VII, we use the SR test instead of CUSUM because of the ease of implementation.

The algorithms discussed above can be easily modified to also account for a change of distribution of rewards. We do not discuss that here for simplicity of exposition. Also, the state and action sequences are more informative. Thus, unless only the reward undergoes a change, and the transition functions do not change, only then it will be useful to consider rewards for detection.

Trade-off Between Detection Performance and Reward

Note that the problems in (2) and (5) are not classical change detection problems. The objective is not to optimize delay subject to constraint on false alarms, but to optimize long term rewards. Thus, while quick detection is key, what is optimal for change detection may not be optimal with respect to optimizing rewards, and vice versa. Thus, there is a fundamental trade-off between detection performance and maximizing rewards. See Section IV below for a more rigorous statement.

The fundamental contribution of this paper is a computationally efficient two-threshold algorithm that optimizes this detection-reward trade-off to achieve near optimal performance. As discussed in the introduction, most of the works in the existing literature either do not exploit tools from the change detection literature, or very few who exploit them, ignore this trade-off in their design.

IV. Quickest Detection with Locally Optimal Solution

One approach to solving either (2) or (5) is to use a locally optimal approach, i.e., to start with using the optimal policy for model $\mathbf{M}_0$, compute the change detection statistics (Shiryaev, SR or CUSUM) over time, and switch to the optimal policy for the model $\mathbf{M}_1$ at the time when the change detection algorithm crosses its threshold. Mathematically, let $\Pi_0 = (\pi_0, \pi_0, \cdots)$ be the (stationary) Markov optimal policy for model $\mathbf{M}_0$, and let $\Pi_1 = (\pi_1, \pi_1, \cdots)$ be the (stationary) Markov optimal policy for model $\mathbf{M}_1$. Then use the following policy

$$\Pi_{\text{loc}} = (\underbrace{\pi_0, \pi_0, \cdots, \pi_0}_{\tau-1 \text{ times}}, \underbrace{\pi_1, \pi_1, \cdots}_{\tau \text{ onward}}), \tag{16}$$

where τ is the stopping time for the Shiryaev, SR or the CUSUM algorithm.

a) Choosing the threshold: In the quickest change detection problem, the threshold A used in the Shiryaev, SR or CUSUM tests is designed to satisfy the constraint on the rate of false alarms. However, for problems in (2) and (5), there is no notion of a false alarm. For the non-Bayesian criterion, one can choose the threshold to satisfy the constraint of α (see (5)). However, in the Bayesian setting, the threshold is a free parameter, and can be optimized for optimal performance.

b) Issues with policy Π_{loc}: As we will show in Section VII, the policy Π_{loc} may lead to poor performance. This is because, as discussed in Section III, the detection performance of the stopping rule τ used depends on the information number I_{π_0}; see (10). Let

$$I_{\max} := \max_{\pi} I_{\pi}.$$

Then, it is possible that

$$I_{\pi_0} \ll I_{\max},$$

and that I_{π_0} is itself quite small. This may lead to poor detection performance, and hence may cause significant loss of revenue or rewards. If it so happens that $I_{\pi_0} \approx I_{\max}$, and I_{π_0} is significant, then one can expect that this simple strategy Π_{loc} itself may lead to near optimal performance. But, this is more a matter of chance, and more sophisticated approaches are needed that considers every possibility. We note that in some cases one may be able to use MOMDP or POMDP based approximations to compute the optimal policy for our problem, but such solutions are computationally demanding. Using a quickest change detection approach leads to computationally efficient algorithmic techniques.

V. A Two-Threshold Switching Strategy to Exploit Reward-Detection Trade-Off

One way to improve the performance of the policy Π_{loc} is to replace the policy Π_0 by another policy that better exploits the reward-detection trade-off. One option is to replace Π_0 by a policy that maximizes the information number I_π. Define for each state $s \in \mathscr{S}$, the information maximizing policy $\pi_{\text{KL}}(s)$ as

$$\pi_{\text{KL}}(s) = \text{ arg max}_a \sum_{s'} \mathscr{T}_1(s,a,s') \, \log \frac{\mathscr{T}_1(s,a,s')}{\mathscr{T}_0(s,a,s')}. \tag{17}$$

Specifically, use

$$\Pi_{\text{KL}} = (\underbrace{\pi_{\text{KL}}, \pi_{\text{KL}}, \cdots, \pi_{\text{KL}}}_{\tau-1 \text{ times}}, \underbrace{\pi_1, \pi_1, \cdots}_{\tau \text{ onward}}). \tag{18}$$

Using this policy will lead to quickest and most efficient detection of model changes. However, since the policy π_{KL} may not be optimal for the MDP model $\mathbf{M}_0$, it may cause significant loss of rewards before the change point.

We propose to use a simple two-threshold strategy that switches between π_0 and π_{KL}, using π_0 to optimize rewards, and switching to π_{KL} for information extraction. The proposed switching strategy is closely related to the notion of sequential design of experiments [18], or to the notion of exploitation and exploration in multi-arm bandit problems [19].

The two-threshold policy Π_{TT} is described in Algorithm 5.1. In words, fix two thresholds A and B, with $B < A$. Also, assume that we are in the Bayesian setting so that we can compute the Shiryaev statistic S_n (8). Then the policy Π_{TT} is defined as follows. Start with the locally optimal policy $\Pi_0 = (\pi_0, \pi_0, \cdots)$ and compute the statistic S_n. As long as $S_n \leq B$, keep using the Markov policy π_0. If $S_n > B$, suggesting that we may be in the wrong model, extract more information by using π_{KL}. If S_n goes below B again, switch back to π_0, else we keep using π_{KL}. If $S_n > A$, switch to the optimal Markov policy for model $\mathbf{M}_1$, that is π_1.

Algorithm 5.1: Π_{TT}: Two-Threshold Switching Strategy

```
Require: Thresholds $A$ and $B$, $B < A$; optimal policies $\pi_0$, $\pi_1$, $\pi_{\text{KL}}$; transition kernels $\mathscr{T}_1(s,a,s')$ and $\mathscr{T}_0(s,a,s')$
Start with $S_0 = 0$
while $S_n \leq A$ do
    Use locally optimal policy $\Pi_0 = (\pi_0, \pi_0, \cdots)$
    Compute the statistic $S_n$ using (8)
    If $S_n \leq B$ continue using the policy $\pi_0$
    If $S_n > B$ use $\pi_{\text{KL}}$
    n = n + 1
end while
Switch to policy $\Pi_1 = (\pi_1, \pi_1, \cdots)$
return
```

In short, use π_0 when $S_n \in [0, B]$, use π_{KL} when $S_n \in (B, A]$, and use π_1 when $S_n \in (A, \infty)$. Thus, the change detection statistic S_n is used as a belief on the unknown model, and is used for detection between $\mathbf{M}_0$ and $\mathbf{M}_1$, depending on whether S_n is small or large, respectively. For moderate values of S_n, the policy π_{KL} is used to improve detection performance.

$$\begin{aligned} \Pi_{\text{TT}} = (&\underbrace{\pi_0, \cdots, \pi_0}_{S_n \leq B}, \underbrace{\pi_{\text{KL}}, \cdots, \pi_{\text{KL}}}_{B < S_n \leq A}, \underbrace{\pi_0, \cdots, \pi_0}_{S_n \leq B}, \cdots, \\ &\cdots, \underbrace{\pi_{\text{KL}}, \cdots, \pi_{\text{KL}}}_{B < S_n \leq A \text{ until } \tau-1}, \underbrace{\pi_1, \pi_1, \cdots}_{\tau \text{ onward}}). \end{aligned} \tag{19}$$

In the non-Bayesian setting, we can replace the Shiryaev statistic S_n by the SR statistic SR_n (14) or the CUSUM statistic W_n (11).

a) On the choice of thresholds A and B: Note first that setting $A = B$ reduces the two-threshold policy Π_{TT} to the locally optimum policy Π_{loc}. Also, by setting $B = 0$ the policy reduces to Π_{KL}. As discussed earlier, this will lead to good detection performance, but may lead to loss of rewards. Akin to Π_{loc}, the thresholds A and B here are free parameters, even for the non-Bayesian setting. Thus, the thresholds can be optimized for best performance: choosing the optimal thresholds is equivalent to choosing the best detection-reward trade-off.

b) Bayesian vs Non-Bayesian: We emphasize that the definition of Π_{TT} is implicitly different for Bayesian and non-Bayesian cases. In the Bayesian case, we have access to the prior on the change point, and we use that to compute the Shiryaev statistic S_n. However, in the non-Bayesian case, we replace the Shiryaev statistic by the Shiryaev-Roberts statistic SR_n or the CUSUM statistic W_n.

c) When will Π_{TT} outperform Π_{loc}?: The amount of performance gain depends in general on the problem structure:

reward function and the transition kernel. However, a condition which must be satisfied is $I_{\pi_0} \ll I_{\max}$. This condition often leads to better detection performance for Π_{TT} as compared to Π_{loc}, resulting in efficient detection and more average rewards. Another condition under which performance gains are significant is when under-performing under any of the models is penalized equally. For example, in the Bayesian case (2), if the discount factor β is small, then the rewards lost in delaying a transition to the optimal policy of $\mathbf{M}_1$ will not affect the overall cost. As a result, a policy can cause large delays, without significantly loosing rewards, and a good detection performance may not necessarily lead to noticeable gain in rewards. Numerical results are provided in the next section to support these observations.

VI. Multiple Models and Change Points

For a parametrized family of models $\{\mathbf{M}_\theta\}$, if the model changes from $\mathbf{M}_{\theta_0}$ to some other unknown model in the family, then the Shiryaev, SR or CUSUM statistic cannot be used for change detection. This is because one needs the exact knowledge of the post-change distribution to compute those statistics. Popular alternatives are to use generalized likelihood ratio (GLR) based tests or mixture based tests. We only discuss the former. A GLR statistic for change detection is defined as

$$G_n = \max_{n-m\le k\le n} \sup_{\|\theta-\theta_0\|\ge\varepsilon} \sum_{i=k}^{n} \log \frac{\mathscr{T}_\theta(s_{i-1},a_{i-1},s_i)}{\mathscr{T}_{\theta_0}(s_{i-1},a_{i-1},s_i)},$$

where ε is the minimum magnitude of change that can occur in the problem.

A replacement for Π_{loc} in this setting would be to use locally optimal policy and switch at the time of stopping, where now the stopping rule is

$$\tau_g = \min\{n \ge 1 : G_n > A\}.$$

Since there is more than one possibility for the post-change model, one can use the maximum likelihood estimate (the θ that achieved the maximum in the expression for G_n at the time of stopping) to choose the model. When the number of models is finite, one can also use the theory of fault isolation to choose the right model [20].

A replacement for Π_{TT} in this setting is even less obvious because of the difficulty in defining an appropriate version of π_{KL}. We propose the following

$$\pi_{\mathrm{KL}}(s) = \arg\max_a \min_{\|\theta-\theta_0\|\ge\varepsilon} \sum_{s'} \mathscr{T}_\theta(s,a,s') \log \frac{\mathscr{T}_\theta(s,a,s')}{\mathscr{T}_{\theta_0}(s,a,s')}. \tag{20}$$

Thus, maximize the worst case KL divergence. Analytical and numerical performance of these schemes will be reported elsewhere.

If there are multiple change points, then as long as the gap between the change points is large enough, we can repeat the two model algorithm for consecutive change points. Specifically, if the GLR based algorithm detects a change and gives θ_1 as an estimate for the post-change parameter, then we can reset the algorithm with θ_1 as the pre-change parameter, and reapply the GLR based test. As long as there is a significant gap between model parameters, one would expect this technique to be close to optimal.

VII. Numerical Results

We now compare the two policies Π_{TT} and Π_{loc} for an inventory control problem, and show that using Π_{TT} results in significant gains. We use the inventory control problem to illustrate the ideas because of its simplicity. The understanding developed through this simple problem can be extended to more complex problems.

a) Inventory control problem: The state s_k is the level of the inventory at time k, and the maximum size of the inventory is N. The action a_k is the additional inventory to be ordered based on the state s_k, making the total inventory $s_k + a_k$. A stochastic demand w_k arrives making the residual inventory or the next state $s_{k+1} = \max\{0, s_k + a_k - w_k\}$. Let c to be cost of ordering a unit of inventory, h be the holding cost per unit, and p be the cost of loosing a unit of demand. Thus the cost per unit time is given by

$$\begin{aligned} C(s_k,a_k) = \mathbb{E}_{w_k}\,[\,ca_k + h\max\{0, s_k + a_k - w_k\} \\ + p\max\{0, -(s_k + a_k - w_k)\}\,]. \end{aligned} \tag{21}$$

It is assumed that at the change point, the distribution of the demands changes from Poisson(λ) to Uniform. The demands are assumed to be independent, conditioned on the change point. For clarification, note that the problem formulations (2) and (5) are about maximizing rewards, but the objective in the inventory control problem is to minimize cost.

b) Numerical results for Bayesian setting: Under the Bayesian setting (2), it is assumed that the change occurs with a geometric prior $\rho = 0.01$. The other parameter values used are $c = 1$, $h = 5$, $\beta = 0.99$, $\lambda = 2$, $N = 10, 20$, and p is chosen to be $100, 200$ or 300. This is a classical optimal control problem and the optimal policy under each model, Poisson or Uniform, can be found using value iteration [2]. The performance of various policies discussed are tabulated in Table I. The results are obtained by performing average over 1000 independent runs, each of horizon size 1000. The Oracle policy is the one that knows the change point, and switches from the optimal policy for the Poisson model to the optimal policy for the Uniform model, exactly at the change point. The policy Π_{loc} employs the Shiryaev algorithm (7). Since there is no concept of false alarms, the choice of threshold A is optimized to achieve minimum possible cost. The policy Π_{TT} also employs the Shiryaev algorithm, and the values of the two thresholds are also chosen to achieve minimum possible cost. For comparison, the costs achieved by performing an MOMDP approximation and a random action strategy is also shown. The performance of the policy Π_{TT} is comparable to that of the MOMDP solution, and significantly better than that of Π_{loc}. Note that the gain is more significant when $I_{\max}$ is much larger than I_{π_0}. Also, since the Poisson arrival rate chosen implies less demand as compared to the Uniform arrival rate, the optimal control is more aggressive for the Uniform arrival model. A delay in detecting the change will result in significant loss of demands. The chosen value of the discount factor β and p ensures that delays in detection,

TABLE I: Bayes discounted cost (2) for different policies: $c = 1$, $h = 5$, $\beta = 0.99$, $\rho = 0.01$. The values in the table are obtained by averaging costs over 1000 independent runs, each of horizon size 1000.

N	p	Oracle	Π_{loc}	Π_{TT}	MOMDP	RANDOM	$I_{\max}$	I_{π_0}
20	100	4057	4980	4579	4688	16489	8.45	2.56
20	200	4229	5678	5051	5144	27655	8.45	2.56
20	300	4277	5884	5238	5325	38821	8.45	3.24
10	100	2336	2619	2515	2677	7861	1.72	1.17
10	200	2445	2902	2728	2876	13623	1.72	1.42
10	300	2550	2920	2813	3020	19386	1.72	1.61

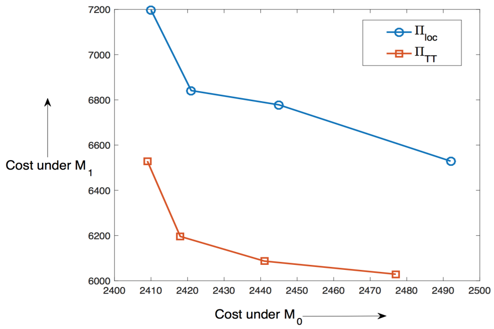

Fig. 1: Performance under $\mathbf{M}_1$ subject to constraints on the performance under $\mathbf{M}_0$. The SR algorithm (14) is used while implementing the policies.

and hence loss in demands, are sufficiently penalized under both regimes, pre-change and post-change.

c) Numerical results for Non-Bayesian setting: An overall discounted criterion like (2) may not be the best or even fair way to compare competing algorithms for a non-stationary environment. One of the primary reasons for using (2) as a criterion was to show that the proposed algorithm can perform comparable to an MOMDP approximation. A more appropriate performance metric is a variational one (5). In this setting, instead of optimizing the threshold used in the policies, the thresholds are selected to satisfy the constraint of α on $\mathbb{E}_\infty \left[\sum_{k=0}^{\infty} \beta^k C_k(s_k, a_k)\right]$, in which the change never occurs. For these choices of the thresholds, the performance $\mathbb{E}_1 \left[\sum_{k=0}^{\infty} \beta^k C_k(s_k, a_k)\right]$ is evaluated for both Π_{TT} and Π_{loc} in which the change occurs at time 1. This way, performance under both the regimes or MDPs are equally weighted, and an algorithm that performs well in every setting can be obtained. The performance comparison is plotted in Fig. 1. The parameter used are $c = 1$, $h = 5$, $\beta = 0.99$, $\lambda = 2$, $N = 20$, and $p = 200$. We use the Shiryaev-Roberts (14) algorithm to detect changes. Clearly, Π_{TT} is superior. Finally, note that since we do not have a prior on the change point, a POMDP or MOMDP solution cannot be obtained.

VIII. Conclusions

This paper presents a novel way to combine techniques used for stationary MDPs and quickest change detection to solve non-stationary MDPs by considering the trade-off between change detection and reward maximization, an important problem that has not been adequate addressed before. Our method uses a two threshold switching strategy to exploit reward-detection trade-off. Our numerical results show that the proposed method can achieve better trade-off, and outperform the state-of-the art QCD method and MOMDP method for solving non-stationary MDP tasks. Future work will include application of ideas developed in this paper to design of reinforcement learning based algorithms for decision making in non-stationary environments. See [21] for a stochastic approximation based approach to this problem.

Acknowledgment

This research was funded in part by Northrop Grumman and Lincoln Laboratory.

References

[1] P. Hernandez-Leal, M. E. Taylor, B. Rosman, L. E. Sucar, and E. M. de Cote, “Identifying and tracking switching, non-stationary opponents: a Bayesian approach,” in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[2] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[3] E. Hadoux, A. Beynier, and P. Weng, “Solving hidden-semi-Markov-mode Markov decision problems,” in *Scalable Uncertainty Management*. Springer, 2014, pp. 176–189.

[4] P. Dayan and T. J. Sejnowski, “Exploration bonuses and dual control,” *Machine Learning*, vol. 25, no. 1, pp. 5–22, 1996.

[5] K. Doya, K. Samejima, K.-i. Katagiri, and M. Kawato, “Multiple model-based reinforcement learning,” *Neural computation*, vol. 14, no. 6, pp. 1347–1369, 2002.

[6] B. C. Da Silva, E. W. Basso, A. L. Bazzan, and P. M. Engel, “Dealing with non-stationary environments using context detection,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 217–224.

[7] S. P. Choi, D.-Y. Yeung, and N. L. Zhang, “Hidden-mode Markov decision processes for nonstationary sequential decision making,” in *Sequence Learning*. Springer, 2000, pp. 264–287.

[8] I. Chades, J. Carwardine, T. Martin, S. Nicol, R. Sabbadin, and O. Buffet, “MOMDPs: a solution for modelling adaptive management problems,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, 2012.

[9] V. Borkar and P. Varaiya, “Adaptive control of Markov chains, i: Finite parameter set,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 953–957, 1979.

[10] ——, “Identification and adaptive control of Markov chains,” *SIAM Journal on Control and Optimization*, vol. 20, no. 4, pp. 470–489, 1982.

[11] V. V. Veeravalli and T. Banerjee, “Quickest change detection,” *Academic press library in signal processing: Array and statistical signal processing*, vol. 3, pp. 209–256, 2013.

[12] R. Allamaraju, H. Kingravi, A. Axelrod, G. Chowdhary, R. Grande, J. P. How, C. Crick, and W. Sheng, “Human aware UAS path planning in urban environments using nonstationary MDPs,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1161–1167.

[13] E. Hadoux, A. Beynier, and P. Weng, “Sequential decision-making under non-stationary environments via sequential change-point detection,” in *Learning over Multiple Contexts (LMCE)*, 2014.

[14] A. Wald, *Sequential analysis*. Courier Corporation, 1973.

[15] A. N. Shiryaev, “On optimum methods in quickest detection problems,” *Theory of Probability & Its Applications*, vol. 8, no. 1, pp. 22–46, 1963.

[16] A. G. Tartakovsky and V. V. Veeravalli, “General asymptotic Bayesian theory of quickest change detection,” *Theory of Probability & Its Applications*, vol. 49, no. 3, pp. 458–497, 2005.

[17] T. L. Lai, “Information bounds and quick detection of parameter changes in stochastic systems,” *Information Theory, IEEE Transactions on*, vol. 44, no. 7, pp. 2917–2929, 1998.

[18] H. Chernoff, “Sequential design of experiments,” *The Annals of Mathematical Statistics*, vol. 30, no. 3, pp. 755–770, 1959.

[19] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[20] A. G. Tartakovsky, “Multidecision quickest change-point detection: Previous achievements and open problems,” *Sequential Analysis*, vol. 27, no. 2, pp. 201–231, 2008.

[21] R. Singh and T. Banerjee, “Kiefer wolfowitz algorithm is asymptotically optimal for a class of non-stationary bandit problems,” *arXiv preprint arXiv:1702.08000*, 2017.