Spend Smart, Learn Fast: Budget Curricula for Safe RL

Micheal Rayan
m.a.rayan@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Thomàs De Los Santos Verrijp
t.d.los.santos.verrijp@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Sohail Farkish
s.farkish@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Andrei Popoviciu
a.popoviciu@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Francesco Colasurdo
f.colasurdo@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Figure 1: Young birds attempting their first flight, a natural metaphor for progressive, budgeted learning.

Abstract

...

CCS Concepts

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Keywords

Do, Not, Use, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:
Micheal Rayan, Thomàs De Los Santos Verrijp, Sohail Farkish, Andrei Popoviciu, and Francesco Colasurdo. 2026. Spend Smart, Learn Fast: Budget Curricula for Safe RL. In *Proceedings of 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '26) (KDD '26)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Ensuring safety remains one of the central challenges in reinforcement learning (RL), particularly when deploying agents in real-world environments where unsafe behavior can have severe consequences. Research in safe RL has broadly followed two main directions [3]:

(1) modifying the agent's optimization criterion to explicitly incorporate safety or risk constraints.
(2) shaping the agent's exploration process to avoid unsafe regions during training.

Both approaches aim to balance performance and safety, but they differ in how safety is enforced and guaranteed. In this work, we focus on the first family of methods, especially in constrained reinforcement learning (CRL) which introduces safety directly into the optimization objective. This paradigm offers stronger theoretical guarantees by ensuring that safety constraints are respected throughout training and deployment. A widely adopted formalism within CRL is the Constrained Markov Decision Process (CMDP) [1], where constraints are embedded into the environment's dynamics rather than applied externally. CMDPs therefore require the

KDD '26, Jeju, Republic of Korea

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

agent to jointly optimize performance and constraint satisfaction, providing a principled way to trade off reward and safety. This makes them particularly suitable for real-world domains where constraint violations are unacceptable.

Recent research has explored using CMDPs not only for safety but also for direct behavior specification. For instance, [5] propose describing desired agent behaviors through sets of static constraints instead of relying on complex reward engineering. Their method employs indicator-based cost functions, enabling intuitive interpretations such as "avoid lava at least 99% of the time." Furthermore, they address instability in Lagrangian optimization by introducing normalized multipliers that prevent difficult constraints from dominating the optimization process. However, in their formulation, constraint thresholds remain fixed throughout training, limiting adaptability as the agent learns.

Our work departs from this static view by introducing dynamic constraint thresholds that evolve during training. This idea is aided by a certain formulation called Budgeted Markov Decision Process (BMDP) where the constraint threshold is treated as a budget incorporated into the state space. Unlike CMDPs, which must be reformulated whenever the constraint changes, BMDPs allow for the seamless accommodation of dynamic budgets, enabling policies that are explicitly budget-aware and adaptive to changing safety levels. This makes them well-suited for curriculum-based training regimes where safety requirements are gradually adjusted.

While [2] demonstrated the feasibility of BMDPs in continuous domains, their approach relies on randomly initialized episode budgets, sampled uniformly at the start of each episode. Although this randomness exposes the agent to varying safety levels, it also leaves the balance between exploration and caution largely to chance, limiting control over the resulting policy's behavior.

To address this, we propose introducing structured budget evolution throughout training, replacing random initialization with deliberate scheduling. Specifically, we investigate budget curricula that control how the initial budget changes over time. Curriculum learning has been successfully used in safe RL to shape agent behavior and exploration strategies [4, 6–8], but prior applications have been limited to CMDPs, where every threshold change requires reformulating the environment. In contrast, BMDPs embed the budget as part of the state, allowing efficient and flexible curriculum design. We evaluate two complementary budget curricula. The first begins with conservative budgets that gradually increase, emphasizing safety early in training and encouraging the development of robust, low-risk behaviors before systematically introducing riskier, higher-reward regimes. The second follows the opposite trajectory, starting with generous budgets that are progressively reduced to promote rapid acquisition of high-return strategies early on, followed by refinement toward safer, more stable behavior. Through this comparison, we aim to understand how structured budget evolution influences the safety–performance trade-off and whether curriculum-based budget adaptation can produce agents that are both robust and high-performing.

Ultimately, our goal is to provide a framework for controlled budget adaptation in BMDPs, bridging the gap between static constraint formulations and adaptive, curriculum-driven safe RL.

2 Related Works

Safe RL in the domain of CMDPs has been approached with a variety of constraint handling techniques. Among the most widely adopted methods are policy gradient methods with constraint handling. Constrained Policy Optimization (Achiam et al., 2017) extends the trust region optimization with the purpose of enforcing near constraint satisfaction during each update, providing empirical stability but lacking scalability. Similarly, primal dual methods (Tessler et al., 2019) introduce Lagrangian multipliers to balance the reward and cost dynamically. These approaches illustrate that constraint satisfaction is achievable in practice, however, doing so while fixed to a fixed threshold. As a result, the problem must be reformulated whenever the safety requirements change, making them ill suited for settings where constraints evolve during training.

Several works have attempted to improve the robustness of constraint handing. Roy et al. (2022) proposes direct behavior specification, leveraging indicator based cost functions, which binarily encode whether constraints are satisfied. Consequently, constraints returns can be interpreted as behaviour frequencies (i.e., "avoid unsafe states at least 99% of the time"). This approach offers an interpretable and scalable alternative to reward engineering introduced in CPOs. Moreover, to address the instability of Lagrangian updates, they introduce normalized multiples, preventing infeasible constraints from dominating learning. However, their formulation still relies on static thresholds, which leaves unexplored how to handle safety requirements that evolve over time.

BMDPs address the static nature of previous methods by embedding the budget directly into the state, making policies explicitly budget aware. This enables agents to adapt to varying thresholds without requiring a redefining of the problem. Carrara et al. (2019) developed the first budgeted RL algorithms in continuous spaces, introducing the Budgeted Bellman operator and propagation of both reward and cost signals. This work depicted risk sensitive exploration and laid the foundation for budget adaptive policies. Nonetheless, this approach assumes uniform random sampling of budgets across episodes. In particular, Carrara et al. (2019) left unexplored whether sequencing budgets in a meaningful order could accelerate learning, improve constraint satisfaction, or reduce unsafe exploration.

Finally, curriculum learning has been proposed as a general strategy to improve RL performance by progressively structuring training tasks (Bengio et al., 2009). This was subsequently extended to safe RL, where curricula is applied to accelerate learning speeds (Portelas el al., 2020) and to enhance safe exploration (Shperberg et al., 2024). However, existing methods have been designed within the CMDP domain, whereas the constraint threshold requires the reformulation of the entire optimization problem. This requirement significantly limits scalability and undermines the efficiency gains promised by curriculum learning. To date, no work has combined the flexibility of budget-aware formulations with curriculum-based training, leaving unexplored how structured budget sequences might improve both safety and learning dynamics.

3 Background

3.1 Budgeted Reinforcement Learning

Safe control under explicit constraints is often modelled as a Constrained MDP (CMDP) [1], where the agent maximizes return while keeping the expected cumulative cost below a threshold β. Changing β typically requires reformulating the optimization, which is inconvenient when safety levels vary across training.

Budgeted MDPs (BMDPs) [2] address this limitation by treating the budget as part of the state. The augmented state $\tilde{s} = (s, \beta)$ and action $\tilde{a} = (a, \beta_a)$ define a kernel

$$\tilde{P}\big((s', \beta') \mid (s, \beta), (a, \beta_a)\big) = P(s' \mid s, a)\, \delta(\beta' - \beta_a),$$

so a budget-aware policy $\pi(a, \beta_a \mid s, \beta)$ chooses both the environment action and the next-step budget. A single policy can thus adapt to different safety levels without changing the problem specification. The agent is feasible at initialization if $\mathbb{E}[G_c^\pi \mid \tilde{s}_0] \leq \beta_0$, with G_r^π and G_c^π the discounted reward and cost.

We learn vector action-values $Q = (Q_r, Q_c)$ using the budgeted optimality operator

$$\mathcal{T}Q(\tilde{s}, \tilde{a}) = R(\tilde{s}, \tilde{a}) + \gamma\, \mathbb{E}_{\tilde{s}'}\big[Q\big(\tilde{s}',\ \pi_{\text{hull}}(\tilde{s}'; Q)\big)\ \big],$$

where π_{hull} is the greedy policy constrained to the convex hull of the (Q_c, Q_r) frontier to respect the current budget. Following [2], Budgeted Fitted Q-Iteration (BFTQ) approximates the fixed point of $\mathcal{T}$ via supervised regression on transitions collected with an ε-greedy mixture of a random budgeted policy and the current greedy policy.

3.2 Curriculum Learning for RL

Curriculum Learning organizes experience as an ordered sequence of tasks to improve efficiency and stability [?]. In our setting, a "task" is the episode's initial budget β_0. We therefore define a curriculum as a schedule $\mathrm{C} : \mathbb{N} \longrightarrow [\beta_{\min}, \beta_{\max}]$ that maps the episode index e to $\beta_0^{(e)} = \mathrm{C}(e)$. We focus on *static* curricula (fixed schedule), which is sufficient to test whether the ordering of safety levels affects learning speed and final performance.

We study three families that project to $[\beta_{\min}, \beta_{\max}]$:

- **Linear (inc./dec.)**: $\beta_0^{(e)} = \beta_{\min} + (\beta_{\max} - \beta_{\min})\frac{e}{E}$ or $\beta_{\max} - (\beta_{\max} - \beta_{\min})\frac{e}{E}$.
- **Exponential (inc./dec.)**: $\beta_0^{(e)} = \beta_{\min} + (\beta_{\max} - \beta_{\min})\big(1 - e^{-\kappa e}\big)$ or the mirrored decay, with $\kappa > 0$.
- **Cosine (periodic)**: $\beta_0^{(e)} = \bar{\beta} + \Delta\beta \cos(2\pi e/T)$, with $\bar{\beta} = \frac{1}{2}(\beta_{\min} + \beta_{\max})$ and $0 < \Delta\beta \leq \frac{1}{2}(\beta_{\max} - \beta_{\min})$.

We evaluate curricula using three standard metrics: time-to-threshold, asymptotic performance, and sample efficiency. This framing isolates the effect of *initial budget ordering* while keeping the within-episode budget dynamics of BMDPs unchanged.

3.3 Positioning

Prior safe-RL work in CMDPs (e.g., policy-gradient methods with Lagrangian updates or trust-region constraints) offers practical mechanisms for constraint satisfaction but assumes fixed thresholds and requires re-specification when safety levels change. Direct behavior specification [5] improves interpretability yet remains tied to static constraints. Our setting leverages BMDPs to keep the problem fixed while we study whether structured *budget curricula* can accelerate learning and improve the safety–performance trade-off.

4 Background

4.1 Budgeted Reinforcement Learning

In Budgeted Reinforcement Learning (BRL), we rely on the framework of *Budgeted Markov Decision Processes* (BMDPs), which can be viewed as an extension of *Constrained Markov Decision Processes* (CMDPs). In this formulation, a budget variable is introduced as part of the state space, representing a threshold for the cumulative cost that the agent is allowed to incur. By incorporating this additional variable, the agent learns policies that explicitly account for risk while exploring.

In a BMDP, the policy not only selects an action but also determines the immediate next budget associated with that action. This mechanism allows the agent to dynamically balance reward maximization with satisfaction of the budget constraint at every timestep.

A key advantage of this formulation is that it enables a *single policy* to generalize across different safety levels. Unlike CMDPs, which must be reformulated whenever constraint thresholds change, BMDPs naturally handle variable budgets, supporting both static and adaptive safety requirements. This property is particularly relevant in environments where constraints evolve during training or deployment—such as in robotic control, energy management, or autonomous driving.

A major method for solving BMDPs was introduced by [2] through the *Budgeted Fitted Q-Iteration* (BFTQ) algorithm. Fitted Q-Iteration uses sampled transitions to iteratively approximate the Bellman target; since the Bellman operator is a contraction mapping, this procedure converges to the optimal Q-function given sufficient samples. In BMDPs, this operator is extended to act over two value dimensions—reward and cost—rather than one. The learning objective becomes minimizing a regression loss over both dimensions, which can be achieved with linear approximators, regression trees, or neural networks.

4.2 Curriculum Learning for Reinforcement Learning

Our second theoretical reference is [?], *"Curriculum Learning for Reinforcement Learning Domains"*, which introduces a principled approach to structuring experience acquisition during agent training. The main idea is to control *how* and *in what order* experience is presented to the agent, thereby improving efficiency and stability of learning.

In Reinforcement Learning (RL), a curriculum can be interpreted as an ordered sequence of tasks or samples—designed to shape the evolution of difficulty or complexity across training. In our specific case, each "task" corresponds to a distinct initial budget value β_0. Rather than sampling budgets uniformly at random, we define a structured schedule over time—linear, exponential, or cyclic—to study how such ordering affects learning.

The concept of a curriculum draws inspiration from human and animal learning: we typically master simpler tasks before progressing to more complex ones. An *increasing* curriculum mirrors this behavior, starting with easy, low-risk conditions and gradually exposing the agent to harder, riskier environments.

4.2.1 Core Components of Curriculum Learning. Following [?], a curriculum is defined by three key components:

Task Generation. This component defines which tasks or conditions the agent will face. Practically, this means specifying the set of "worlds" or parameter configurations describing each step in the curriculum. For example, in a game environment, the agent might begin with few obstacles and gradually face more complex challenges. In our case, task generation corresponds to defining the different initial budget values β_0 encountered across episodes.

Sequencing. Sequencing determines the order in which tasks are presented. It can be fixed (e.g., strictly increasing or decreasing) or adaptive, changing based on the agent's performance. For instance, if the agent consistently performs well at a given difficulty, the next episode can move to a harder one. In our setting, sequencing defines whether the initial budget increases, decreases, or oscillates over time—and at what rate.

Transfer. This component captures how knowledge from one task transfers to the next. In RL, transfer typically occurs through shared model parameters or updates to the Q-function. For example, a robot that learns to walk slowly on flat terrain can transfer that experience to learn faster on rough terrain or when running. Transfer ensures that early experiences contribute to efficient adaptation later in the curriculum.

Together, these components structure a progression that guides the agent from safer, simpler experiences to more complex, risk-sensitive ones, promoting stability and generalization.

4.2.2 Types of Curricula: Static and Dynamic. Narvekar et al. distinguish between two main categories of curricula:

- **Static curriculum:** the sequence of tasks (or budgets) is predefined and does not depend on the agent's performance.
- **Dynamic curriculum:** the difficulty evolves based on the agent's success, allowing adaptive progression.

In our BMDP context, this distinction translates to how the initial budget β_0 is determined:

- In a *static* curriculum, β_0 follows a fixed, predefined schedule.
- In a *dynamic* curriculum, β_0 depends on performance metrics from previous episodes.

It is important to note that this adaptability affects only the initialization of each episode; within an episode, the budget evolves step-by-step according to BMDP dynamics.

4.2.3 Metrics for Evaluating Curricula. To evaluate a curriculum's effectiveness, [?] propose three key metrics:

Time-to-Threshold. Measures the number of interactions required for the agent to reach a target performance level. A well-designed curriculum reduces this time, accelerating learning.

Asymptotic Performance. Indicates the final steady-state performance achieved after convergence. It reflects whether the curriculum enables the agent to reach higher or more stable outcomes compared to uniform training.

Sample Efficiency. Quantifies how much useful information the agent extracts per training sample. An efficient curriculum allows strong performance with fewer environment interactions, which is crucial in real-world, costly-to-simulate domains.

5 Problem Statement

Ensuring safety in reinforcement learning (RL) requires agents to maximize their expected cumulative rewards while simultaneously satisfying explicit safety or resource-based constraints. In the framework of Constrained Markov Decision Processes (CMDPs) [1], such constraints are expressed through an auxiliary cost function $C(s, a)$ with an upper bound β on its expected return. This leads to the following optimization problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right], \\ \text{s.t.} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t C(s_t, a_t)\right] \leq \beta. \end{aligned} \tag{1}$$

However, in many real-world scenarios, safety requirements and operational tolerances evolve dynamically during training or deployment. Examples include adaptive mission risk limits, varying energy resources, or changing environmental hazards. CMDPs are not well suited for such cases since each change in β requires reformulating the optimization problem, limiting their scalability in dynamic environments.

5.1 Notation

Basic symbols.

$\mathcal{S}, \mathcal{A}$	State and action spaces
$P(s' \mid s, a)$	Transition kernel
$R_r(s, a)$, $R_c(s, a)$	Immediate reward and cost
$\gamma \in [0, 1)$	Discount factor
$\beta \in \mathcal{B} \subset \mathbb{R}_+$	Budget (constraint threshold)
$\tilde{s} = (s, \beta)$, $\tilde{a} = (a, \beta_a)$	Augmented state and action
$\tilde{\mathcal{S}} = \mathcal{S} \times \mathcal{B}$, $\tilde{\mathcal{A}} = \mathcal{A} \times \mathcal{B}$	Augmented spaces
G_r^π, G_c^π	Discounted returns of reward and cost
$Q = (Q_r, Q_c)$	Vector action-value function
$e \in \mathbb{N}$	Episode index ($\mathbb{N}$: natural numbers)
$\mathrm{C} : \mathbb{N} \to \mathcal{B}$	Curriculum schedule for initial budgets
$\beta_{\min}$, $\beta_{\max}$	Minimum and maximum budgets

Budgeted dynamics. We work with a Budgeted Markov Decision Process (BMDP) in which the transition kernel embeds the budget update:

$$\tilde{P}\big((s', \beta') \mid (s, \beta), (a, \beta_a)\big) \;=\; P(s' \mid s, a)\, \delta(\beta' - \beta_a),$$

so that the policy $\pi(a, \beta_a \mid s, \beta)$ selects both an environment action and the next-step budget. Within an episode, the budget evolves as $\beta_{t+1} = \beta_a$.

Objectives. For any initial augmented state (s_0, β_0), the reward and cost returns are

$$G_r^\pi = \sum_{t \geq 0} \gamma^t R_r(s_t, a_t), \qquad G_c^\pi = \sum_{t \geq 0} \gamma^t R_c(s_t, a_t),$$

and the budgeted feasibility constraint requires $\mathbb{E}[G_c^\pi \mid \tilde{s}_0] \leq \beta_0$.

Budgeted Bellman operator. We learn $Q = (Q_r, Q_c)$ over $\tilde{\mathcal{S}} \times \tilde{\mathcal{A}}$ via the budgeted optimality operator

$$\mathcal{T}Q(\tilde{s}, \tilde{a}) \;=\; R(\tilde{s}, \tilde{a}) + \gamma\, \mathbb{E}_{\tilde{s}' \sim \tilde{P}}\big[Q\big(\tilde{s}',\; \pi_{\text{hull}}(\tilde{s}'; Q)\big)\big],$$

where π_{hull} denotes the greedy budget-feasible policy that selects an action (or mixture) lying on the convex hull of the (Q_c, Q_r) frontier, thereby respecting the current budget.

BFTQ (learning rule). Budgeted Fitted Q-Iteration (BFTQ) approximates the fixed point of $\mathcal{T}$ via regression on a dataset $\mathcal{D}$ of transitions:

$$\theta_{k+1} = \arg\min_{\theta} \sum_{(\tilde{s}, \tilde{a}, r, c, \tilde{s}') \in \mathcal{D}} \left\| Q_\theta(\tilde{s}, \tilde{a}) - \hat{\mathcal{T}}\, Q_{\theta_k}(\tilde{s}, \tilde{a}, \tilde{s}') \right\|_2^2.$$

We use ε-greedy exploration with a schedule over episodes, mixing a random budgeted policy and the current greedy policy.

Curriculum over initial budgets. Instead of sampling β_0 uniformly, we initialize each episode with

$$\beta_0^{(e)} \;=\; \mathrm{C}(e) \;\in\; [\beta_{\min}, \beta_{\max}],$$

according to one of the schedules studied below (linear, exponential, cosine). Typical parametrizations include a growth/decay rate $\kappa > 0$ (exponential) and a mean/amplitude $(\bar{\beta}, \Delta\beta)$ (cosine).

Evaluation metrics. We report (i) *time-to-threshold* (episodes to reach a target return), (ii) *asymptotic performance* (converged return under budget), and (iii) *sample efficiency* (performance per interaction).

5.2 Budgeted MDP Formulation

Budgeted Markov Decision Processes (BMDPs) [2] extend CMDPs by embedding the budget directly into the state representation, forming augmented tuples

$$\tilde{s} = (s, \beta) \in \tilde{\mathcal{S}} = \mathcal{S} \times \mathcal{B}, \quad \tilde{a} = (a, \beta_a) \in \tilde{\mathcal{A}} = \mathcal{A} \times \mathcal{B}.$$

The transition kernel becomes:

$$\tilde{P}\big((s', \beta') \mid (s, \beta), (a, \beta_a)\big) = P(s' \mid s, a)\, \delta(\beta' - \beta_a),$$

embedding budget dynamics into the environment itself. A policy in this setting outputs both the action and the next budget:

$$\pi(a, \beta_a \mid s, \beta).$$

This formulation enables a single policy to generalize across a continuous range of budgets, allowing dynamic constraint handling without costly reformulations.

For any initial augmented state (s_0, β_0), the learning objective becomes:

$$\max_{\pi} \; \mathbb{E}[G_r^\pi \mid \tilde{s}_0 = (s_0, \beta_0)] \quad \text{s.t.} \quad \mathbb{E}[G_c^\pi \mid \tilde{s}_0 = (s_0, \beta_0)] \le \beta_0,$$

where $G_r = \sum_{t \ge 0} \gamma^t R_r(s_t, a_t)$ and $G_c = \sum_{t \ge 0} \gamma^t R_c(s_t, a_t)$ denote the discounted returns for reward and cost respectively.

5.3 Budgeted Bellman Operator and BFTQ

We learn vector action-values $Q = (Q_r, Q_c)$ over the augmented space $\tilde{\mathcal{S}} \times \tilde{\mathcal{A}}$. The Budgeted Bellman optimality operator is defined as:

$$\mathcal{T}Q(\tilde{s}, \tilde{a}) = R(\tilde{s}, \tilde{a}) + \gamma\, \mathbb{E}_{\tilde{s}' \sim \tilde{P}}\big[Q\big(\tilde{s}',\; \pi_{\text{hull}}(\tilde{s}'; Q)\big)\big],$$

where π_{hull} denotes the greedy policy constrained by the convex hull of the (Q_c, Q_r) frontier, selecting the optimal action mixture that respects the budget.

The Budgeted Fitted-Q (BFTQ) algorithm iteratively approximates the fixed point of this operator via regression:

$$\theta_{k+1} = \arg\min_{\theta} \sum_{(\tilde{s}, \tilde{a}, r, \tilde{s}') \in \mathcal{D}} \left\| Q_\theta(\tilde{s}, \tilde{a}) - \hat{\mathcal{T}}\, Q_{\theta_k}(\tilde{s}, \tilde{a}, \tilde{s}') \right\|_2^2,$$

where $\mathcal{D}$ is a dataset of transitions collected during exploration.

5.4 Risk-Sensitive Exploration

To collect $\mathcal{D}$, a risk-sensitive exploration strategy mixes a random budgeted policy π_{rand} and a greedy policy π_{greedy}. Each episode begins with an initial budget β_0 sampled uniformly:

$$\beta_0 \sim \mathcal{U}(\mathcal{B}),$$

ensuring exposure to various risk levels. During training, exploration actions (a, β_a) are sampled uniformly from a feasible simplex $\Delta_{\mathcal{A} \times \mathcal{B}}$ satisfying $\mathbb{E}[\beta_a] \le \beta$, while exploitation relies on π_{greedy}. Within an episode, the budget evolves step-by-step as $\beta_{t+1} = \beta_a$.

While uniform sampling promotes broad coverage, it introduces inefficiencies: episodes may begin at infeasible or uninformative budget values, wasting training samples and potentially destabilizing learning.

5.5 Applying Curriculum Learning to BMDPs

Our main contribution lies in integrating Curriculum Learning principles within the BMDP framework. Specifically, we modify the *BFTQ* algorithm such that the initial budget β_0 at the beginning of each episode is not sampled uniformly, but rather determined by a structured or adaptive *curriculum schedule*.

In this setting, the budget itself becomes both the constraint and the single "task" operated on by the curriculum. We explore multiple curriculum configurations—each defining a different ordering or evolution of β_0—to address our central research question:

> *Can the ordering of initial budgets influence the agent's learning process? If so, how and to what extent?*

To investigate this, we study several curriculum types:

- **Increasing curriculum:** the budget starts small (restrictive) and gradually increases, allowing the agent to master safe behavior before exploring riskier strategies.
- **Decreasing curriculum:** the budget starts large and progressively shrinks, forcing the agent to become more efficient as safety margins tighten.
- **Periodic curriculum:** the budget oscillates cyclically, mimicking environments where available resources fluctuate over time (e.g., seasonal or operational variations).
- **Structured random curriculum:** introduces semi-random orderings to test whether regularity or unpredictability better supports adaptation.

For example, a *decreasing* curriculum could represent a startup with an initial investment that diminishes over time, requiring increasingly efficient decisions. Conversely, an *increasing* curriculum could represent a system that begins under strict safety constraints and earns more flexibility as it performs well—mirroring the gradual trust humans gain in autonomous systems.

Beyond addressing our research question, these curricula aim to improve sample efficiency, enable smooth adaptation to dynamic constraints, and establish a structured connection between CMDPs, BMDPs, and Curriculum Learning. In essence, the curriculum provides a principled framework for organizing the agent's exposure to varying risk regimes during training, offering new insights into how the sequencing of safety budgets shapes learning and performance.

5.6 Curriculum on the Initial Budget

We propose to replace the random initialization of β_0 with a structured *curriculum function*

$$\mathrm{C} : \mathbb{N} \to \mathrm{B}, \qquad \beta_0^{(e)} = \mathrm{C}(e),$$

which deterministically schedules the episode's starting budget based on its index e. This provides a controlled and interpretable progression of budget values—either gradually increasing, decreasing, or cyclic—allowing the agent to experience different safety levels in a structured manner.

This modification aligns with the principles of curriculum learning, where the order and structure of training experiences can accelerate convergence and improve policy stability. By systematically varying the budget, the agent can progressively adapt from conservative to risk-tolerant behaviors (or vice versa), promoting both safety and efficiency in exploration.

We explore three curriculum schedules, each projected to $[\beta_{\min}, \beta_{\max}]$:

Linear (increasing / decreasing).

$$\begin{aligned} \beta_0^{(e)} &= \beta_{\min} + (\beta_{\max} - \beta_{\min})\,\tfrac{e}{E}, \qquad &&\text{(increasing)},\\ \beta_0^{(e)} &= \beta_{\max} - (\beta_{\max} - \beta_{\min})\,\tfrac{e}{E}, \qquad &&\text{(decreasing)}. \end{aligned}$$

This linear curriculum offers a smooth and predictable transition between budget extremes. It is particularly useful in environments where safety tolerance should evolve steadily over time—for example, in robotic manipulation tasks where the agent gradually learns to apply larger forces after mastering safe low-force interactions.

Exponential (increasing / decreasing).

$$\begin{aligned} \beta_0^{(e)} &= \beta_{\min} + (\beta_{\max} - \beta_{\min})\big(1 - e^{-\kappa e}\big), \qquad &&\text{(increasing)},\\ \beta_0^{(e)} &= \beta_{\max} - (\beta_{\max} - \beta_{\min})\big(1 - e^{-\kappa e}\big), \qquad &&\text{(decreasing)}, \end{aligned} \qquad \kappa > 0.$$

The exponential curriculum allows for rapid adaptation at the start of training while slowing the rate of change as the agent stabilizes. Such a schedule is well-suited for safety-critical navigation domains where initial fast exploration under tight constraints quickly transitions into steady, fine-grained learning of safe yet efficient behaviors.

Cosine (periodic increase–decrease).

$$\begin{aligned} \beta_0^{(e)} &= \bar{\beta} + \Delta\beta \cos\!\big(2\pi e/T\big),\\ \bar{\beta} &= \tfrac{1}{2}(\beta_{\min} + \beta_{\max}), \qquad 0 < \Delta\beta \le \tfrac{1}{2}(\beta_{\max} - \beta_{\min}). \end{aligned}$$

Algorithm 1: Risk-sensitive exploration (BFTQ) with Curriculum Budget Initialization

```
Data: An environment $\mathcal{E}$, a BFTQ solver, $W$ CPU workers,
      curriculum $C$
Result: A batch of transitions $\mathcal{D}$
$\mathcal{D} \leftarrow \emptyset$
for each intermediate batch do
    split episodes between $W$ workers
    for each episode $e$ in batch in parallel do
        sample initial budget $\beta \leftarrow C(e)$   // Curriculum
            modification (replaces $\beta \sim \mathcal{U}(\mathcal{B})$)
        while episode not done do
            update $\varepsilon$ from schedule
            sample $z \sim \mathcal{U}([0,1])$
            if $z < \varepsilon$ then
                sample $(a, \beta_a) \sim \mathcal{U}(\Delta_{\mathcal{A}\times\mathcal{B}})$   // Explore
            else
                sample $(a, \beta_a) \sim \pi_{\text{greedy}}(a, \beta_a \mid s, \beta; Q^*)$
                 // Exploit
            append transition $(s, \beta, a, \beta_a, R, C, s')$ to batch $\mathcal{D}$
            step episode budget $\beta \leftarrow \beta_a$
    $\pi_{\text{greedy}}(\cdot \mid \cdot; Q^*) \leftarrow$ BFTQ($\mathcal{D}$)
return $\mathcal{D}$
```

This cyclic curriculum alternates between conservative and liberal budgets, exposing the agent to both high-risk and low-risk phases repeatedly. It is motivated by real-world processes with alternating operational modes—such as energy-constrained drones that alternate between exploration (high budget) and precision landing (low budget) phases—encouraging robustness to fluctuating safety conditions.

Across all curricula, our objective is to study whether the order and shape of budget progression influence learning speed, sample efficiency, and long-term adherence to safety constraints.

5.7 Modified Algorithm: Risk-Sensitive BFTQ with Curriculum Budgeting

5.8 Objective and Evaluation

Formally, we aim to determine an optimal curriculum C^* that improves learning efficiency while maintaining constraint satisfaction:

$$C^* = \arg\min_{C}\ \mathbb{E}\left[L(\pi_C)\right], \quad \text{s.t.} \quad \forall t,\ \mathbb{E}_{\pi_C}\left[C_t\right] \le \beta_t, \qquad (2)$$

where $L(\pi_C)$ denotes the training loss (e.g., Bellman residual or regret) under curriculum C.

We evaluate performance using three metrics: (i) **Time-to-threshold** — number of episodes required to reach a target return, (ii) **Asymptotic performance** — final converged reward under budget, and (iii) **Sample efficiency** — the amount of useful information gained per interaction.

6 Methodology

Our methodology integrates the principles of *Budgeted Reinforcement Learning* (BRL) with structured *curriculum learning* to examine how different budget initialization schedules influence learning dynamics, safety adherence, and policy robustness. Specifically, we adapt the *Budgeted Fitted Q-Iteration* (BFTQ) algorithm [2] by replacing uniform random initialization of budgets with deterministic or structured curriculum schedules. This section describes the formulation, curriculum design, training pipeline, and evaluation protocol.

6.1 Overall Framework

The training loop proceeds as follows:

(1) Initialize the environment and policy network.
(2) At the beginning of each episode e, select an initial budget $\beta_0^{(e)}$ from the curriculum function $\mathrm{C}(e)$ instead of sampling uniformly.
(3) Run the BFTQ algorithm with ε-greedy exploration, collecting transitions over augmented states $\tilde{s} = (s, \beta)$ and actions $\tilde{a} = (a, \beta_a)$.
(4) Update the Q-function using supervised regression on collected data, applying the budgeted Bellman operator to ensure safety constraints are enforced.
(5) Evaluate policy performance periodically on held-out budget levels.

7 Acknowledgments

...

8 Appendices

...

Acknowledgments

To Robert, for the bagels and explaining CMYK and color spaces.

References

[1] Eitan Altman. 1999. *Constrained Markov Decision Processes.* Chapman & Hall/CRC, Boca Raton.

[2] Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. 2019. Budgeted Reinforcement Learning in Continuous State Space. arXiv:1903.01004 [cs.LG] https://arxiv.org/abs/1903.01004 arXiv:1903.01004.

[3] Javier García and Fernando Fernández. 2015. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research* 16, 42 (2015), 1437–1480. http://jmlr.org/papers/v16/garcia15a.html

[4] Cevahir Koprulu, Thiago D. Simão, Nils Jansen, and ufuk topcu. 2025. Safety-Prioritizing Curricula for Constrained Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations.* https://openreview.net/forum?id=f3QR9TEERH

[5] Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher Pal. 2022. Direct Behavior Specification via Constrained Reinforcement Learning. arXiv:2112.12228 [cs.LG] https://arxiv.org/abs/2112.12228 arXiv:2112.12228.

[6] Shahaf S. Shperberg, Bo Liu, and Peter Stone. 2023. Relaxed Exploration Constrained Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2821–2823.

[7] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. 2020. Safe Reinforcement Learning via Curriculum Induction. *CoRR* abs/2006.12136 (2020). arXiv:2006.12136 https://arxiv.org/abs/2006.12136

[8] Q. Yang. 2023. *Risk Aversion and Guided Exploration in Safety-Constrained Reinforcement Learning.* Dissertation (TU Delft). Delft University of Technology. doi:10.4233/uuid:ca5a81c2-f895-4638-bce5-1423a5943381

Received not yet; revised not yet; accepted not yet