European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Markov decision processes with burstiness constraints

Michal Golan, Nahum Shimkin*

Viterbi Faculty of Electrical and Computer Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 6 December 2022
Accepted 31 July 2023
Available online xxx

Keywords:
Dynamic programming
Constrained Markov decision processes
Burstiness constraints

ABSTRACT

We consider a Markov Decision Process (MDP), over a finite or infinite horizon, augmented by so-called (σ, ρ)-burstiness constraints. Such constraints, which had been introduced within the framework of network calculus, are meant to limit some additive quantity to a given rate over any time interval, plus a term which allows for occasional and limited bursts. We introduce this class of constraints for MDP models, and formulate the corresponding constrained optimization problems. Due to the burstiness constraints, constrained optimal policies are generally history-dependent. We use a recursive form of the constraints to define an augmented-state model, for which sufficiency of Markov or stationary policies is recovered and the standard theory may be applied, albeit over a larger state space. The analysis is mainly devoted to a characterization of feasible policies, followed by application to the constrained MDP optimization problem. A simple queuing example serves to illustrate some of the concepts and calculations involved.

1. Introduction

A real-valued sequence $(d_t)_{t\in\mathcal{T}}$ over a discrete interval $\mathcal{T} \subset \mathbb{Z}$ is said to satisfy a (σ, ρ)-*burstiness constraint* if the following set of inequalities is satisfied:

$$\sum_{t=t_1}^{t_2} d_t \le (t_2 - t_1 + 1)\rho + \sigma, \quad \text{for all } t_2 \ge t_1 \ : \ t_1, t_2 \in \mathcal{T}. \tag{1}$$

Here ρ is the *rate parameter*, and $\sigma \ge 0$ the *burstiness parameter*. If (d_t) is a stochastic sequence, we require this relation to hold with probability 1 (w.p. 1). In this paper we consider Markov Decision Processes that are subject to one or more burstiness constraints of this type, where d_t stands for an auxiliary running-cost function.

Generally speaking, burstiness of temporal processes is characterised by short intervals of intensive activity followed by periods of reduced or mild activity. In the context of telecommunication systems, traffic burstiness and its effect on link and network congestion have attracted continuing attention over several decades. Earlier work such as Eckberg (1985), Sriram & Whitt (1986), Heffes & Lucantoni (1986) was largely motivated by burstiness effects in telephone communication, leading to a broad interest in traffic characteristics of communication networks (Jiang & Dovrolis, 2005; Leland, Taqqu, Willinger, & Wilson, 1994; Paxson & Floyd, 1995; Willinger, Taqqu, Sherman, & Wilson, 1997) and shared computing facilities (Niu et al., 2021; Yin, Lu, Zhao, Chen, & Liu, 2014). More recent application areas include textual data streams (Kleinberg, 2003; Lappas, Arai, Platakis, Kotsakos, & Gunopulos, 2009), as well as burstiness of human dynamics and its manifestation in social networks (Barabasi, 2005; Karsai, Jo, & Kaski, 2018). Fundamental questions addressed in the literature encompass quantitative criteria for characterizing and identifying burstiness, stochastic models of bursty processes, uncovering the origin of burstiness in different areas, performance analysis of queues and networks subject to bursty traffic, and mitigating the adverse effects of burstiness via traffic shaping and control.

The definition of burstiness in this paper borrows from the literature on network calculus, where it had been introduced in the seminal paper (Cruz, 1991). Network calculus (Chang, 2000; Le Boudec & Thiran, 2001) emerged in the 1990's as a deterministic theory (with ensuing probabilistic extensions, Fidler & Rizk, 2014; Jiang & Liu, 2008; Yaron & Sidi, 1993) for quality of service analysis of packet data networks. Traffic arrivals at a networked system are bounded by upper envelope functions, and service guarantees are modeled by service curves, leading to computable performance bounds on backlog, delay, loss, output flow, etc. In this context, (σ, ρ)-burstiness constraints correspond to *affine* traffic envelopes. These constraints are also closely related to the well known leaky bucket and token bucket algorithms (e.g., Tanenbaum, 2003), which are used for traffic monitoring and shaping in packet switched networks.

Markov Decision Processes, or MDPs, are a standard model and framework for sequential optimization and control of discrete-time stochastic systems (Bellman, 1957; Bertsekas, 2012; Feinberg & Shwartz, 2002; Puterman, 1994). Recently, MDPs have drawn

* Corresponding author.
E-mail address: shimkin@ee.technion.ac.il (N. Shimkin).

https://doi.org/10.1016/j.ejor.2023.07.045
0377-2217/

considerable additional interest as the underlying model in the bustling area of Reinforcement Learning (Sutton & Barto, 2018). Constrained MDPs (CMDPs) augment the basic MDP model by adding one or more constraints that should be satisfied by a feasible policy. The constraints considered are mostly of the same type as the standard reward objectives, namely expressed as upper bounds on an expected cumulative cost, discounted cost or average cost. A useful summary of the basic theory for these models is available in the monograph (Altman, 1999). Since dynamic programming is not directly applicable to CMDPs, the major solution approaches rely on state-action frequencies, or resort to a Lagrangian formulation. More recent theoretical developments may be found, e.g., in Dufour & Prieto-Rumeau (2013), Borkar & Jain (2014), Chang (2015), Jaskiewicz & Nowak (2019), Yu (2022), Varagapriya, Singh, & Lisser (2022) and references therein. Additionally, CMDPs have recently been gaining considerable interest in the learning literature as one of the major approaches to safe reinforcement learning; see, e.g., Achiam, Held, Tamar, & Abbeel (2017), Tessler, Mankowitz, & Mannor (2018), and the reviews in Garcıa & Fernández (2015), Liu, Halev, & Liu (2021), Gu et al. (2022).

The burstiness constraints we consider in this paper are *sample-path* constraints, required to hold with probability one. In this sense they are closest to sample-path infinite-horizon average cost constraints (Ross & Varadarajan, 1989), as opposed to the more common expected-cost constraints. Indeed, it is evident that the requirements in Eq. (1) imply a rate bound of ρ over the infinite horizon. However, burstiness constraints of the type (1), which involve multiple inequalities, lead to a non-standard MDP structure. In particular, an optimal policy subject to such constraints will generally not be Markovian (hence not stationary) with respect to the original state. Nonetheless, the special structure of the constraints allows to define certain *burstiness variables* which can be computed recursively, allowing the (σ, ρ)-constraints to be expresses equivalently as bounds on these variables. By augmenting the state with these burstiness variables, a standard MDP structure may be restored, leading to the applicability of standard MDP solution methods for the augmented model. This construction is presented and analyzed in the present paper.

The reformulation of (σ, ρ)-burstiness constraints in terms of bounds on recursively-computed variables can be traced back to the equivalent formulation of the leaky bucket algorithm mentioned above, where the burstiness parameter σ corresponds to the buffer size and the burstiness constraints to the queue length. In the context of an optimization problem, that analogy has been utilized in the papers Anantharam & Konstantopoulos (1993), Konstantopoulos & Anantharam (1995), which present flow control algorithms that maximize the throughput in a queue with arbitrary input stream and whose output is subject to (σ, ρ)-burstiness constraints. These papers reformulate the burstiness constraints in terms of *virtual backlog* variables, which are complimentary to the burstiness constraints used here. We note that the latter paper considers both discrete-event processes and continuous (fluid) processes, where for the latter the tool of reflection maps is needed to formally define the backlog process. In the present discrete time context this is not needed, and an explicit recursive definition can be given, similarly to Anantharam & Konstantopoulos (1993).

The previous paragraph points to the use of burstiness constraints in the domain of network flow control, where the constrained quantities are the flows of data packets over a communication link. For applications in the broad area of operations research, burstiness constraints may pertain to various other quantities such as flows of goods, vehicles, or customers. For examples, in the area of inventory control, a contract with a supplier may cap the number of the number of ordered items (or shipments) allowed per day, but allow some slack (additional orders) to address special circumstances. Such a restriction restriction may be quantified in terms of a burstiness constraint, and standard problems of optimal inventory control may then be addressed with this additional constraint.

The main contributions of this paper are:

1. Introducing (σ, ρ)-burstiness constraints in the general context of MDP models.
2. Formulating the equivalent augmented-state problem in terms of the burstiness variables.
3. Providing a characterization of feasible policies and their calculation.
4. Utilizing this characterization for the solution of (σ, ρ)-constrained MDP problems.

We consider these issues both for a finite time horizon and an infinite one, with the former leading to the latter. Feasible states are characterized in terms of a binary-valued indicator functions, termed the feasibility functions, which are computed recursively in the finite-horizon case. An equivalent recursion is specified in terms of corresponding feasibility sets. For the infinite-horizon model, the feasibility function is characterized as the *maximal* solution of a Bellman-like fixed point equation, and a value iteration procedure is shown to converge to that solution from suitable initial conditions. Furthermore, under some mild conditions on the constraint parameters, convergence is obtained within a finite and bounded number of iterations, thus providing a finite algorithm for computing the infinite-horizon feasibility function. In either case, the set of feasible policies is characterized in terms of the feasibility functions. Given that characterization, constrained optimal policies are obtained via standard MDP theory within the augmented-state model.

The paper proceeds as follows. Section 2 formulates the basic model considered in this paper, and outlines the associated problems of feasibility and burstiness-constrained optimality. Section 3.2 reformulates the constraints in terms of the burstiness variables, and introduces the augmented-state model that will be analyzed in the rest of the paper. Feasibility is examined is Section 4 for the finite-horizon model, and in Section 5 for the infinite-horizon case. Constrained optimization problems are considered in Section 6, followed by concluding remarks in Section 8. The Appendix presents an example that illustrates some of the computations involved in feasibility determination for the infinite-horizon model.

2. Model and problem formulation

This section presents the burstiness-constrained MDP model that is considered in this paper. We start by describing the underlying standard MDP model, and follow by introducing the associated burstiness constraints in the context of this model.

2.1. The underlying MDP

Consider a discrete-time controlled Markov chain, specified through the following quantities:

- Time horizon $\mathcal{T}$, with $\mathcal{T} = \{0, 1, \dots, N-1\}$ (finite horizon) or $\mathcal{T} = \mathbb{N}_0 \triangleq \{0, 1, 2, \dots\}$ (infinite horizon).
- A finite state space S and finite action sets $\{A_s : s \in S\}$, where A_s is the set of available actions at state s.
- State transition probabilities $P = \{p(s'|a, s) : s, s' \in S, a \in A_s\}$.
- A running-reward function, specified by $r(s, a) \in \mathbb{R}$: $s \in S, a \in A_s$.

We note that in the finite horizon case the basic model quantities can be allowed to depend on time, with the results below essentially unchanged. As our main interest is in the infinite-horizon

problem, we will not make such dependence explicit to avoid notational clutter.

A deterministic policy is a sequence of mappings $\pi_t : H_t \to \cup_{s\in S} A_s$ for $t \in \mathcal{T}$, where H_t is the set of possible histories of the form $h_t = (s_0, a_0, \ldots, s_t)$, such that $\pi(h_t) \in A_{s_t}$. A general (history dependent, randomized) policy is a similar map with $\pi(h_t)$ a probability vector over A_{s_t}, from which a_t is chosen independently at random. Let Π denote the set of general policies.

Given an initial state $s_0 = s$ and a policy $\pi \in \Pi$, the description above induced a probability distribution P_s^π over the state-action sequence $(s_t, a_t)_{t\in\mathcal{T}}$ (with the addition of s_N in the finite-horizon case). Let E_s^π denote the corresponding expectation operator.

We shall refer below to the following standard objective functions, which are to be maximized (subject to the imposed constraints).

(i) Finite horizon: The expected total reward,

$$J^\pi(s) = E_s^\pi\left(\sum_{t=0}^{N-1} r(s_t, a_t) + r_N(s_N)\right). \tag{2}$$

(ii) Infinite horizon: The expected discounted reward, with discount factor $0 < \gamma < 1$,

$$J^\pi(s) = E_s^\pi\left(\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right). \tag{3}$$

2.2. Burstiness constraints

Let $(d_t)_{t\in\mathcal{T}}$ denote a real-valued sequence, and fix parameters $\sigma \geq 0$ and $\rho \in \mathbb{R}$. Following Cruz (1991), the sequence (d_t) is said to satisfy a (σ, ρ)-burstiness constraint if

$$\sum_{t=t_1}^{t_2} d_t \leq (t_2 - t_1 + 1)\rho + \sigma, \quad \text{for all } t_1 \leq t_2,\ t_1, t_2 \in \mathcal{T}.$$

Clearly, the long-term average of this sequence is bounded by the rate parameter ρ. However, the sum of a sub-sequence of consecutive elements may exceed that rate limitation by an amount determined by the burstiness parameter σ. For example, consider a $\{0, 1\}$-valued sequence, and rate parameter $\rho = 0.5$. Then the sequence $1, 0, 1, 0, 1, 0, \ldots$ can be seen to obey the constraints for $\sigma \geq 0.5$, while the sequence $1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, \ldots$ satisfies the constraint only for $\sigma \geq 1.5$. Note also that for $\sigma < 0$ the above set of constraints degenerates to the single-term requirement $d_t \leq \rho + \sigma$, hence this case need not be further considered.

Returning to the MDP model, we add $m \geq 1$ burstiness constraints to the basic model, where each constraint $i \in \{1, \ldots, m\}$ is defined through two real parameters $\sigma_i \geq 0$ and ρ_i, and a constraint function $d_i : S \times A \to \mathbb{R}$. We thus require the following set of inequalities to hold for every $i \in \{1, \ldots, m\}$:

$$\sum_{t=t_1}^{t_2} d_i(s_t, a_t) \leq (t_2 - t_1 + 1)\rho_i + \sigma_i, \text{ for all } t_1 \leq t_2,\ t_1, t_2 \in \mathcal{T}. \tag{4}$$

This may be conveniently expressed in vector form, as

$$\sum_{t=t_1}^{t_2} d(s_t, a_t) \leq (t_2 - t_1 + 1)\rho + \sigma, \quad \text{for all } t_1 \leq t_2,\ t_1, t_2 \in \mathcal{T}, \tag{5}$$

where $d(s, a) = (d_1(s.a), \ldots, d_m(s, a))$, $\rho = (\rho_1, \ldots, \rho_m)$, and $\sigma = (\sigma_1 \ldots, \sigma_m) \geq 0$. The set of constraints in (5) forms the (σ, ρ)-burstiness constraints, or just (σ, ρ)-constraints, for the MDP model.

Definition 1. A policy $\pi \in \Pi$ is (σ, ρ)-*feasible* from initial state $s \in S$ if the (σ, ρ)-constraints in (5) are satisfied with P_s^π-probability 1. Let $\Pi(\sigma, \rho; s)$ denote the set of these policies.

Definition 2. The (σ, ρ)-constraints are *feasible from initial state* s if $\Pi(\sigma, \rho; s)$ is nonempty.

The main questions that we address in this paper concern *feasibility* and *constrained optimization*.

Feasibility: Given burstiness parameters (σ, ρ), determine whether the burstiness constraints are feasible from an initial state s. If so, characterize the set of feasible policies $\Pi(\sigma, \rho; s)$.

Constrained Optimization: Given parameters (σ, ρ) and an initial state s, find a feasible policy (if one exists) that maximizes the relevant reward criterion in (2) or (3):

$$\pi^* \in \operatorname{argmax}\left\{J^\pi(s) : \pi \in \Pi(\sigma, \rho; s)\right\}.$$

We refer to such a policy as a burstiness-constrained optimal policy.

Note that the feasibility question does not involve the reward criterion, and hence can be addressed independently of it. This will be the subject of Sections 4 and 5.

3. Burstiness variables and the augmented model

Our analysis of the burstiness-constrained MDP model will rely on reformulating these constraints as constraints on suitably defined *burstiness variables*, which can be computed recursively in time. It will be then be useful to append these variables to the state of the original system, thus forming an augmented MDP model, in which the burstiness constraints are equivalently expressed as constraints on the state-variables. The analysis in this paper will mostly be carried out within this augmented model.

3.1. The burstiness variables

Similarly to Anantharam & Konstantopoulos (1993), Konstantopoulos & Anantharam (1995), we proceed to reformulate the burstiness constraints in terms of suitable recursively-computed variables. Given the sequence $(d(s_t, a_t))_{t\in\mathcal{T}}$ and constraint parameters (σ, ρ), let $y_0 = 0 \in \mathbb{R}^m$, and define

$$y_{t+1} = \left(\max_{t_1\in\{0,\ldots,t\}} \sum_{k=t_1}^{t} \big(d(s_k, a_k) - \rho\big)\right)^+, \quad t \in \mathcal{T}. \tag{6}$$

Here $(y)^+ = \max\{0, y\}$, and the maximum is taken componentwise. We refer to y_t and its components as the *burstiness variables* at time t. Note that $y_t \in \mathbb{R}_+^m = \{y \in \mathbb{R}^m : y \geq 0\}$.

Lemma 1. *Consider the sequence (y_t) as defined above.*

(i) *The set of burstiness constraints in (5) is equivalent to the inequalities*

$$y_{t+1} \leq \sigma, \quad t \in \mathcal{T}. \tag{7}$$

(ii) *The sequence (y_t) satisfies the following recursion, starting with $y_0 = 0$:*

$$y_{t+1} = \big(y_t + d(s_t, a_t) - \rho\big)^+, \quad t \in \mathcal{T}. \tag{8}$$

Proof. (*i*) Denote $d_t = d(s_t, a_t)$ for brevity. Observe the equivalence of (5) with each of the following two statements: (I) $\sum_{t=t_1}^{t_2}(d_t - \rho) \leq \sigma$, $0 \leq t_1 \leq t_2 \in \mathcal{T}$. (II) $\max_{t_1\in\{0,\ldots t_2\}} \sum_{t=t_1}^{t_2}(d_t - \rho) \leq \sigma$, $t_2 \in \mathcal{T}$. Recalling that $\sigma \geq 0$, the claim follows upon comparing the latter with the definition of y_t in (6).

(*ii*) We next show that y_t, as defined in (6), satisfies the stated recursion. For $t_1 \leq t$, denote $S(t_1, t) = \sum_{k=t_1}^{t}(d_k - \rho)$. Noting that

$S(t,t) = (d_t - \rho)$ and $S(t_1,t) = S(t_1,t-1) + (d_t - \rho)$ for $t_1 < t$, we obtain that $y_{t+1} = (z_{t+1})^+$, where

$$\begin{aligned} z_{t+1} &= \max_{t_1 \in \{0,\dots,t\}} S(t_1,t) = \max\left\{ \max_{t_1 \in \{0,\dots,t-1\}} S(t_1,t),\ (d_t - \rho) \right\} \\ &= \max\left\{ \max_{t_1 \in \{0,\dots,t-1\}} S(t_1,t-1),\ 0 \right\} + (d_t - \rho) \\ &= y_t + d_t - \rho\,. \end{aligned}$$

Hence $y_{t+1} = (z_{t+1})^+$ satisfies (8). □

Remarks:

1. As evidenced by the definition of the burstiness variables in (6) and by Lemma 1(i), these variables measure how close we came to violating the burstiness constraints. More precisely, the difference $(\sigma - y_{t+1})_i$ measures how close we came to violating the ith burstiness constraint over all sequences that extend up to time t.
2. Note that the burstiness variables depend on ρ, but not on σ.
3. The burstiness variables as defined here are complementary to the *virtual backlog* variables σ_t defined in Konstantopoulos & Anantharam (1995). That is, $y_t = \sigma - \sigma_t$. In terms of the virtual backlog variable, the feasibility requirement is $\sigma_t \geq 0$. We chose here to keep the burstiness constraint explicit, in the form of the inequalities $y_t \leq \sigma$.

3.2. The state-augmented model

Consider the burstiness-constrained MDP model specified in Section 2. Appending the burstiness variables to the state, we define a modified MDP model $\tilde{\mathcal{M}}$, with extended state:

$$\tilde{s}_t = (s_t, y_t) \in S \times \mathbb{R}_+^m\,. \tag{9}$$

(Later we will effectively restrict y_t a compact set, and possibly further to a finite set.) The action sets and reward function are inherited from the original model, namely $\tilde{A}_{s,y} = A_s$ and $\tilde{r}(s,y,a) = r(s,a)$.

Observing (8), the transition law for the extended state is given by

$$p(s_{t+1} = s' | \tilde{s}_t = (s,y), a_t = a) = p(s'|s,a), \tag{10}$$

$$y_{t+1} = \big(y_t + d(s_t,a_t) - \rho\big)^+, \tag{11}$$

with initial state (s_0, y_0). Note that the state component y_t coincides with the burstiness variables as defined in the previous subsection, provided that $y_0 = 0$.

For future reference, let us introduce the following notations:

$$S'(s,a) = \big\{ s' \in S : p(s'|s,a) > 0 \big\}, \tag{12}$$

$$y'(s,y,a) = \big(y + d(s,a) - \rho\big)^+. \tag{13}$$

Thus, $y_{t+1} = y'(s_t, y_t, a_t)$, and $s_{t+1} \in S'(s_t,a_t)$ w.p. 1.

We next formalize the relation between policies in the two models. Let $\tilde{\Pi}$ denote the set of general (history dependent, randomized) policies in $\tilde{\mathcal{M}}$. We henceforth denote the original model by $\mathcal{M}$. Two policies $\pi \in \Pi$ and $\tilde{\pi} \in \tilde{\Pi}$ are said to be *equivalent* if they induce the same probability distribution on the sequence $(s_t, y_t, a_t)_{t \in \mathcal{T}}$, for any initial condition $s_0 = s \in S$ in $\mathcal{M}$ and corresponding initial conditions $(s_0, y_0) = (s, 0)$ in $\tilde{\mathcal{M}}$.

Lemma 2. *For any policy $\pi \in \Pi$ there exists an equivalent policy $\tilde{\pi} \in \tilde{\Pi}$, and vice versa.*

Proof. Clearly, any policy $\pi \in \Pi$ can be mapped to an equivalent policy $\tilde{\pi} \in \tilde{\Pi}$ simply by ignoring the state component y_t in the latter. Conversely, a policy $\tilde{\pi} \in \tilde{\Pi}$ can be mapped to an equivalent policy $\pi \in \Pi$ in the original model, since y_t is a function of $(s_0, a_0, \dots, s_{t-1}, a_{t-1})$ only. □

We refer to the specific equivalent policy π mentioned in the proof above as the *$\mathcal{M}$-equivalent policy of $\tilde{\pi}$*, and denote it by $\tilde{\pi}/\mathcal{M}$. Note that for a Markov (or stationary) policy $\tilde{\pi}$ in $\tilde{\mathcal{M}}$, the equivalent policy $\pi = \tilde{\pi}/\mathcal{M}$ in the original model will generally not be Markov (nor stationary).

Lemma 2 implies that any optimization problem in $\mathcal{M}$ that depends on the distribution of the sequence $(s_t, y_t, a_t)_{t \in \mathcal{T}}$ can be solved for $\tilde{\mathcal{M}}$, and the resulting policy interpreted as the equivalent policy in $\mathcal{M}$. Let us specify the implications for the feasibility problem.

Recall, from Definition 1, that a policy π in the original model $\mathcal{M}$ is termed (σ,ρ)-feasible if it satisfies the burstiness constraints in (5). For the state-augmented model $\tilde{\mathcal{M}}$, the corresponding notion of feasibility is defined as follows.

Definition 3. A policy $\tilde{\pi} \in \tilde{\Pi}$ is termed *(σ,ρ)-feasible in $\tilde{\mathcal{M}}$ from initial state (s,y)* if $y = y_0 \leq \sigma$, and $y_{t+1} \leq \sigma$ holds for all $t \in \mathcal{T}$ ($P^{\tilde{\pi},(s,y)}$- *a.s.*). Accordingly, the (σ,ρ)-constraints are said to be *feasible in $\tilde{\mathcal{M}}$ from initial state* (s,y) if such a policy exists.

Proposition 3 (Equivalence of feasibility in $\mathcal{M}$ and $\tilde{\mathcal{M}}$)**.**

(i) *Let $\pi \in \Pi$ and $\tilde{\pi} \in \tilde{\Pi}$ be equivalent policies. Then π is (σ,ρ)-feasible in $\mathcal{M}$ from initial state s if, and only if, $\tilde{\pi}$ is (σ,ρ)-feasible in $\tilde{\mathcal{M}}$ from initial state $(s,0)$.*

(ii) *The (σ,ρ)-constraints are feasible in $\mathcal{M}$ from initial state $s \in S$ if, and only if, they are feasible in $\tilde{\mathcal{M}}$ from initial state $(s,0)$.*

Proof. Recall the definition of a feasible policy for $\mathcal{M}$ in Definition 1. Claim (i) follows by Lemma 1 and the definition of y_t in $\tilde{\mathcal{M}}$. Specifically, by Lemma 1, satisfying $y_{t+1} \leq \sigma$, $t \in \mathcal{T}$ in $\tilde{\mathcal{M}}$ from initial state $(s,0)$ is equivalent to satisfying the burstiness constraints in (5) from initial state s. Property (ii) is now a direct consequence of (i). □

Given the observations above, the questions posed in this paper regarding burstiness-constrained optimization may be conveniently formulated and solved within the augmented model, with obvious implications for the original model. In the rest of the paper we will mostly refer to the augmented model.

4. Feasibility analysis: Finite horizon

We turn to consider the question of feasibility of the constraints for specified values of the burstiness parameters (σ,ρ), namely existence of a policy under which the burstiness constraints are satisfied with probability 1. The analysis will also yield a characterization of feasible policies, in terms of single-stage conditions on feasible actions at each stage.

Let us precede the detailed analysis by stating the following obvious single-stage feasibility conditions, which apply both to the finite and infinite horizon problems.

Proposition 4 (Single-stage feasibility conditions)**.**

(i) *Sufficient condition: Suppose that for each state $s \in S$ there exists an action $a \in A_s$ such that $d(s,a) \leq \rho$. Then the (σ,ρ)-constraints are feasible from any initial state and for any vector $\sigma \geq 0$.*

(ii) *Necessary condition: A policy π is (σ,ρ)-feasible from initial state s_0 only if $d(s,a) \leq \rho + \sigma$ holds for any state-action pair (s,a) that is visited with positive $P^{\pi}_{s_0}$-probability over $\mathcal{T}$.*

The sufficient condition in (i) is satisfied, for example, in flow control problems with burstiness constraints on the output traffic, provided that the transmitter is allowed to halt transmission at any

time slot. Hence, the question of feasibility does not arise for such constraints. Characterizing feasible policies still remains of interest of course.

In this section we consider the finite-horizon model, namely $\mathcal{T} = \{0, 1, \dots, N-1\}$. We shall refer mainly to the augmented model $\tilde{\mathcal{M}}$, with the relevant definition or feasibility as stated in Definition 3. The implications to the original model $\mathcal{M}$ are immediate, as noted in Lemma 3.

The section is divided into four subsections. The first presents a characterization of feasibility in terms of *feasibility indicator functions*, and develops the dynamic programming equation for their recursive computation. The second subsection interprets these results in terms of corresponding *feasibility sets*. In the third we consider the single-constraint case, for which the recursive computations are simpler and explicit. The fourth subsection addresses the computational issue for the more involved multiple-constraints case; the required computations are facilitated within a discrete model, that arises by restricting the the constraint parameters and functions to lie on a common grid.

4.1. Feasibility indicator functions

Recall, from Definition 3, that the (σ, ρ)-constraints are feasible from initial state (s, y) if there exists a policy such that $y_0 \le \sigma, \dots, y_N \le \sigma$. We aim to characterize those initial conditions (s, y) for which this holds, as well as the set of feasible policies.

Consider the following indicator functions, which we term the *feasibility functions*. For $(s, y) \in S \times \mathbb{R}^m_+$, let

$$\begin{aligned} v^\pi_N(s,y) &= \mathbf{1}\Big\{\pi \in \tilde{\Pi} \text{ is feasible from initial state } (s,y)\Big\} \\ &= \mathbf{1}\Big\{y_0 \le \sigma, \dots, y_N \le \sigma,\ P^\pi_{s,y}\text{-a.s.}\Big\}. \end{aligned}$$

Here $\mathbf{1}\{\cdot\}$ is the logical indicator function, namely $\mathbf{1}\{A\}$ equals 1 if condition A is satisfied, and is 0 otherwise. Similarly, let

$$\begin{aligned} v^*_N(s,y) &= \mathbf{1}\{\text{the constraints are feasible from initial state } (s,y)\} \\ &= \mathbf{1}\{\exists \pi \in \tilde{\Pi} \text{ s.t. } y_0 \le \sigma, \dots, y_N \le \sigma,\ P^\pi_{s,y}\text{-a.s.}\}. \end{aligned} \tag{14}$$

It follows that $v^*_N(s,y) = \max_{\pi \in \tilde{\Pi}} v^\pi_N(s,y)$.

We next apply dynamic programming to compute v^*. For $t = 0, \dots, N$, define

$$\begin{aligned} v_t(s,y) = \mathbf{1}\Big\{\exists \pi \in \tilde{\Pi}_t \text{ s.t. } y_t \le \sigma, \dots, y_N \le \sigma,\ P^{\pi,t}_{s,y}\text{-a.s.}\Big\}, \\ (s,y) \in S \times \mathbb{R}^m_+. \end{aligned} \tag{15}$$

Here $\tilde{\Pi}_t$ is the set of policies in $\tilde{\mathcal{M}}$ that are started at time t rather than 0, and $P^{\pi,t}_{s,y}$ is the respective probability distribution from time t onward with initial state $(s_t, y_t) = (s, y)$. Note that $v_0 = v^*_N$.

Observe that $v_t(s,y) = 0$ if $y \not\le \sigma$, due to the requirement $y_t = y \le \sigma$. Thus, we may restrict attention to the compact set

$$Y_{[0,\sigma]} \triangleq \Big\{y \in \mathbb{R}^m_+ : y \le \sigma\Big\} = \prod_{i=1}^m [0, \sigma_i]. \tag{16}$$

Let $\mathcal{V}_B = \{v : S \times \mathbb{R}^m_+ \to \{0,1\}\}$ denote the set of binary-valued functions over the state-space of $\tilde{\mathcal{M}}$, and define the map $\mathcal{L} : \mathcal{V}_B \to \mathcal{V}_B$ as follows:

$$\begin{aligned} \mathcal{L}(v)(s,y) = \mathbf{1}\{y \in Y_{[0,\sigma]}\} \cdot \max_{a \in A_s} \min_{s' \in S'(s,a)} v(s', y'(s,y,a)), \\ (s,y) \in S \times \mathbb{R}^m_+. \end{aligned} \tag{17}$$

Recall that S' and y' were defined in (12)–(13). In more explicit terms, the above expression for $\mathcal{L}(v)$ may be interpreted as follows: $\mathcal{L}(v)(s,y) = 1$ if, and only if, $y \le \sigma$ and there exists an action $a \in A_s$ such that $v(s', y'(s,y,a)) = 1$ for every $s' \in S'(s,a)$.

We record some basic properties of the operator $\mathcal{L}$.

Lemma 5 (Properties of $\mathcal{L}$). *We say that a function $v \in \mathcal{V}_B$ is Pareto monotone-decreasing if, for every $y, z \in \mathbb{R}^m_+$ and $s \in S$, $z \le y$ implies that $v(s,z) \ge v(s,y)$. The following properties hold for the operator $\mathcal{L} : \mathcal{V}_B \to \mathcal{V}_B$ defined in (17).*

(i) *If $v \in \mathcal{V}_B$ is Pareto monotone-decreasing, then so is $\mathcal{L}(v)$.*
(ii) *For $v \in \mathcal{V}_B$, if for each $s \in S$ the set $Y_s = \{y \in \mathbb{R}^m_+ : v(s,y) = 1\}$ is closed (i.e., v is upper semi-continuous), then the same holds for $\mathcal{L}(v)$.*
(iii) *$\mathcal{L}$ is a monotone operator. That is, if $v_a \le v_b$ then $\mathcal{L}(v_a) \le \mathcal{L}(v_b)$.*

Proof. (*i*) Let $0 \le z \le y$. Recalling that $y'(s,y,a) = (y + d(s,a) - \rho)^+$, it follows that $y'(s,z,a) \le y'(s,y,a)$. Thus, if v is Pareto-monotone decreasing, then $v(s', y'(s,z,a)) \ge v(s', y'(s,y,a))$. Observe also that $0 \le z \le y$ implies $\mathbf{1}\{z \in Y_{[0,\sigma]}\} \ge \mathbf{1}\{y \in Y_{[0,\sigma]}\}$. Combining these two observations, it follows by (17) that $\mathcal{L}(v)(s,z) \ge \mathcal{L}(v)(s,y)$.

(*ii*) The claim follows by continuity of $y'(s,y,a)$ in y, which is evident. In more detail, supposing that v satisfies property (*ii*), we need to show that the following set is closed for each $s \in S$:

$$Y'_s \triangleq \Big\{y \in \mathbb{R}^m_+ : \mathcal{L}(v)(s,y) = 1\Big\}. \tag{18}$$

Let $Y_{s,a,s'} = \{y \in \mathbb{R}^m_+ : v(s', y'(s,y,a)) = 1\}$, and note by (17) that Y'_s may be expressed as

$$Y'_s = Y_{[0,\sigma]} \cap \left(\bigcup_{a \in A_s} \bigcap_{s' \in S'(s,a)} Y_{s,a,s'} \right). \tag{19}$$

Now, by continuity of y' in y and the assumed property of v, $Y_{s,a,s'}$ is a closed set. Since closedness is preserved under intersections and finite unions, it follows that Y'_s is a closed set.

(*iii*) Monotonicity of $\mathcal{L}$ follows directly by observing its definition. □

The feasibility functions (v_t) satisfy the following recursive equations, as implied by standard dynamic programming arguments.

Theorem 6 (Backward Recursion). *It holds that*

$$v_t = \mathcal{L}(v_{t+1}), \quad t = 0, 1 \dots, N-1,$$

with $v_N(s,y) = \mathbf{1}\{y \le \sigma\}$ for $(s,y) \in S \times \mathbb{R}^m_+$.

Proof. Observe first that the only requirement for v_N in (15) is that $y \le \sigma$, hence $v_N(s,y) = \mathbf{1}\{y \le \sigma\}$. Next, for $t < N$, the requirement for $v_t(s,y) = 1$ is that (for some policy, w.p. 1) $y_t \le \sigma$, and $y_{t'} \le \sigma$ for $t' \in \{t+1, \dots, N\}$. The first requirement is equivalent to $y = y_t \le \sigma$. By standard dynamic programming arguments, the second requirement is equivalent to existence of an action $a \in A_s$ such that

$$Prob\Big\{v_{t+1}(s_{t+1}, y_{t+1}) = 1 \,|\, s_t = s, y_t = y, a_t = a\Big\} = 1. \tag{20}$$

Noting that $y_{t+1} = y'(s_t, y_t, a_t)$ and $s_{t+1} \in S'(s_t, a_t)$ w.p. 1, (20) is equivalent to $v(s', y'(s,y,a)) = 1$ for every $s' \in S'(s,a)$. Since $v \in \{0,1\}$, "every" is translated to a minimum over $v(s', y'(s,y,a))$, and "exists" to a maximum. □

Computational aspects of this recursion will be discussed in the subsequent subsections. Here we observe some basic structural properties of the feasibility functions v_t.

Proposition 7 (Properties of v_t). *For each $t \in \{0, \dots, N\}$ and $s \in S$, the following properties hold.*

(i) *(Pareto-monotonicity:) $v_t(s,z) \ge v_t(s,y)$ for every $z \in \mathbb{R}^m_+$ such that $z \le y$.*
(ii) *(Continuity:) The set $\{y \in \mathbb{R}^m_+ : v_t(s,y) = 1\}$ is closed (i.e., v_t is upper semi-continuous).*

Proof. Observe that $v_N(s,y) = \mathbf{1}\{y \le \sigma\}$, which satisfies properties (*i*) and (*ii*). The respective assertions now follow by induction

from v_{t+1} to $v_t = \mathcal{L}(v_{t+1})$, by observing the properties of $\mathcal{L}$ in Lemma 5. □

We now turn to the characterization of feasible policies in terms of the feasibility functions. We first observe the following characterization of feasibility, which follows directly from the definitions.

Proposition 8 (Feasibility Characterization)**.** *Recall Definitions 2 and 3 of feasibility, as well as the definition of $v_0 = v^*$ in (15).*

(i) *The (σ,ρ)-constraints are feasible in $\tilde{\mathcal{M}}$ from initial state $(s,y) \in S \times \mathbb{R}^m_+$ if, and only if, $v_0(s,y) = 1$.*
(ii) *Consequently, the (σ,ρ)-constraints are feasible in $\mathcal{M}$ from initial state s if, and only if, $v_0(s,0) = 1$.*

To characterize feasible policies, we define the following sets of *feasible actions*, in terms of the feasibility functions. For $t = 0,\dots,N-1$ and $(s,y) \in S \times \mathbb{R}^m_+$, let

$$A_t(s,y) = \left\{ a \in A_s : y \in Y_{[0,\sigma]},\ v_{t+1}(s', y'(s,y,a)) = 1\ \forall s' \in S'(s,a) \right\}. \tag{21}$$

Note that $A_t(s,y)$ is defined as the set of actions that achieve the maximal value of 1 the definition (17) of $\mathcal{L}(v)$, with $v = v_t$. More precisely, the following properties hold.

Lemma 9 (Properties of the Feasible Actions Sets)**.** *For $(s,y) \in S \times \mathbb{R}^m_+$ and $t \in \mathcal{T}$,*

(i) *$A_t(s,x) \subset A_t(s,y)$ if $x \geq y$.*
(ii) *$A_t(s,y) \neq \emptyset$ if, and only if, $v_t(s,y) = 1$.*
(iii) *If $a_t \in A_t(s_t,y_t)$, then $v_{t+1}(s_{t+1},y_{t+1}) = 1$ (w.p. 1). Furthermore, if $y_t \in Y_{[0,\sigma]}$, these two properties are equivalent.*

Proof. Property (i) follows by the Pareto monotonicity of v_t (Proposition 7) and the Pareto-monotonicity of $y'(s,y,a) = (y + d(s,a) - \rho)^+$ in y. Property (ii) follows directly by the definition of $A_t(s,y)$ and of the recursion $v_t = \mathcal{L}(v_{t+1})$. For (iii), recall that $y_{t+1} = y'(s_t,y_t,a_t)$ and $s_{t+1} \in S'(s_t,a_t)$ (w.p. 1). But by definition of A_t, $a_t \in A_t(s,y)$ implies that $v_{t+1}(s', y'(s_t,y_t,a_t)) = 1$ for every $s' \in S'(s_t,a_t)$. Similarly, if $y_t \in Y_{[0,\sigma]}$, the opposite implication is also true. □

The set of feasible policies may now be characterized as follows.

Theorem 10 (Feasible Policies)**.**

(i) *A policy $\pi \in \tilde{\Pi}$ is (σ,ρ)-feasible from initial state (s,y) if, and only if, the following holds with $P^{\pi}_{s,y}$-probability 1:*

$$a_t \in A_t(s_t,y_t), \quad t = 0,1\dots N-1\,.$$

(ii) *Let $\pi \in \tilde{\Pi}$ be a policy that for any history $h_t = (s_0,y_0,a_1,\dots,s_t,y_t)$ chooses $a_t \in A_t(s_t,y_t)$ (possibly randomly) if the latter set is nonempty, and otherwise chooses a_t arbitrarily. Then $\pi \in \tilde{\Pi}$ is (σ,ρ)-feasible from any initial state (s,y) such that $v_0(s,y) = 1$.*

Proof. (i) We observe first that $y_t \leq \sigma$ for all $0 \leq t \leq N$ is equivalent to $v_t(s_t,y_t) = 1$ for all $0 \leq t \leq N$. The *if* part of this equivalence follows trivially since $v_t(s,y) = 0$ if $y \not\leq \sigma$, while the reverse implication follows by definition of the functions v_t, for $v_t(s_t,y_t) = 0$ implies that no policy exists such that $y_k \leq \sigma$ for all $k = t+1,\dots,N$.

Let $t = 0$, and consider $y_0 \in Y_{[0,\sigma]}$. If $A_0(s_0,y_0) = \emptyset$, then $v_1(s_1,y_1) = 0$ for any action a_0 (Lemma 9), hence there is no feasible policy. Similarly, choosing an action $a_0 \notin A_0(s_0,y_0)$ leads to $v_1(s_1,y_1) = 0$. On the other hand, choosing any action $a_0 \in A_0(s_0,y_0)$ leads to a state (s_1,y_1) such that $v_1(s_1,y_1) = 1$, and in particular $y_1 \leq \sigma$. Proceeding inductively, we obtain that $y_t \leq \sigma$ for $0 \leq t \leq N$ if, and only if, $a_t \in A_t(s_t,y_t)$ for $0 \leq t \leq N$.

(ii) Follows directly from part (i) together with Proposition 8. □

It is evident that the requirement for feasibility in the last proposition may be satisfied, in particular, by *Markov* policies in $\tilde{\mathcal{M}}$, such that a_t is a function of (s_t,y_t) only. Thus, if the constraints are feasible, there always exists a feasible deterministic Markov policy.

Remark 1. We note that feasibility determination can be formulated as an optimization problem in different ways. For example, consider the following minimization problem with an additive cost:

$$v^*(s,y) = \sup_{\pi \in \tilde{\Pi}} E^{\pi}_{s,y} \sum_{t=0}^{N} \mathbf{1}\{y_t \leq \sigma\}\,.$$

In this case, $v^*(s,y) = N+1$ would be equivalent to feasibility of the constraints. While the value functions and the associated optimality equations may vary with the formulation, they would all lead to the same *feasibility sets*, which we consider next.

4.2. Reformulation via feasibility sets

We next reformulate the results of the previous subsection in terms of feasibility sets, which are essentially those values of states (s_t,y_t) for which the (σ,ρ)-constraints can be satisfied from time t onward. As mentioned, this formulation of the feasibility conditions is independent of the specific value function used for feasibility analysis, and also serves to clarify some of the computations involved.

Recall the definition of the feasibility function v_t in (15). For $t \in \{0,\dots,N\}$, let

$$Z_t \triangleq \left\{ (s,y) \in S \times Y_{[0,\sigma]} : v_t(s,y) = 1 \right\} \tag{22}$$

$$= \left\{ (s,y) \in S \times Y_{[0,\sigma]} : \exists \pi \in \tilde{\Pi}_t \text{ s.t. } y_t \leq \sigma,\dots,y_N \leq \sigma,\ P^{t,\pi}_{s,y}\text{-a.s.} \right\}. \tag{23}$$

With this definition, v_t is indeed the indicator function of the set Z_t. Note that we restrict the y component of Z_t to the compact set $Y_{[0,\sigma]} = \{y \in \mathbb{R}^m_+ : y \leq \sigma\}$, rather than $\mathbb{R}^m_+$, since $v_t(s,y) = 0$ for $y \not\leq \sigma$.

It will be convenient to denote $Z_t(s) = \{y \in Y_{[0,\sigma]} : (s,y) \in Z_t\}$. Thus, Z_t may be identified with the collection $\{Z_t(s), s \in S\}$.

We may now repeat the main results of Section 4.1 in terms of the feasibility sets. The following properties are direct consequences of Proposition 7 and of (22).

Corollary 11 (Properties of Z_t)**.** *For each $t \in \{0,\dots,N\}$ and $s \in S$,*

(i) *(Pareto-monotonicity:) If $y \in Z_t(s)$, then $\{x \in Y_{[0,\sigma]} : x \leq y\} \subset Z_t(s)$.*
(ii) *(Closedness:) $Z_t(s)$ is a closed set.*

Let us present recursive equations for Z_t. To that end, we define the set operator $\mathcal{F}$ that parallels the operator $\mathcal{L}$ in (17). Let $\mathcal{Z}$ denote the collection of sets $\mathcal{Z} = \{Z \subset S \times Y_{[0,\sigma]}\}$. As before, for $Z \in \mathcal{Z}$ we denote $Z(s) = \{y \in Y_{[0,\sigma]} : (s,y) \in Z\}$. Thus, the set Z may be identified with the collection $\{Z(s),\ s \in S\}$.

Define a map $\mathcal{F} : Z \in \mathcal{Z} \mapsto \mathcal{F}(Z) \in \mathcal{Z}$ as follows:

$$\mathcal{F}(Z)(s) = \bigcup_{a \in A_s} \bigcap_{s' \in S'(s,a)} Y(s,a,s'), \quad s \in S, \tag{24}$$

where

$$Y(s,a,s') = \left\{ y \in Y_{[0,\sigma]} : y'(s,y,a) \in Z(s') \right\}. \tag{25}$$

Then $\mathcal{F}$ is the equivalent of the operator $\mathcal{L}$ in the sense described in the next lemma. For $v \in \mathcal{V}_B$ and $Z \in \mathcal{Z}$, we say that v is the indicator of Z if $Z = \{(s,y) \in S \times Y_{[0,\sigma]} : v(s,y) = 1\}$.

Lemma 12. *If v is the indicator of Z, then $\mathcal{L}(v)$ is the indicator of $\mathcal{F}(Z)$.*

Proof. Let v be the indicator of Z. By definition of $\mathcal{L}$ in (17), $\mathcal{L}(v)(s,y) = 1$ is equivalent to the following: $y \in Y_{[0,\sigma]}$, and there exists an action $a \in A_s$ such that $v(s', y'(s,y,a)) = 1$ for every $s' \in S'(s,a)$. Now, $v(s', y'(s,y,a)) = 1$ is equivalent to $y'(s,y,a) \in Z(s')$. Translating the qualifiers (and, exists, for all) in the above statement to set operations, we obtain

$$\mathcal{F}(Z)(s) = Y_{[0,\sigma]} \cap \left(\bigcup_{a \in A_s} \bigcap_{s' \in S'(s,a)} \left\{ y \in \mathbb{R}^m_+ : y'(s,y,a) \in Z(s') \right\} \right).$$

Rearranging yields (24)–(25). □

Corollary 13 (Backward Recursion)**.** *Let $\mathcal{F}$ be the operator defined in (24)–(25). Then*

$$Z_t = \mathcal{F}(Z_{t+1}), \quad t = 0, 1, \ldots, N-1,$$

and $Z_N = S \times Y_{[0,\sigma]}$.

Proof. The claim follows from Proposition 6, upon noting that v_N is the indicator of Y_N, and applying Lemma 12. □

As a corollary to Propositions 8 and 10, we have the following characterization of feasibility in terms of the feasibility sets.

Corollary 14 (Feasibility and feasible policies)**.** *Recall Definitions 2 and 3 of feasibility.*

(i) *The (σ,ρ)-constraints are feasible in $\tilde{\mathcal{M}}$ from initial state $(s,y) \in S \times \mathbb{R}^m_+$ if, and only if, $y \in Z_0(s)$. Consequently, the (σ,ρ)-constraints are feasible in $\mathcal{M}$ from initial state s if, and only if, $0 \in Z_0(s)$, or equivalently $Z_0(s) \neq \emptyset$.*

(ii) *The characterization of feasible policies in Theorem 10 remains valid with the feasible actions sets $A_t(s,y)$ defined, equivalently, as*

$$A_t(s,y) = \left\{ a \in A_s \,:\, y'(s,y,a) \in Z_{t+1}(s'),\ \forall s' \in S'(s,a) \right\}, \tag{26}$$

and $v_0(s,y) = 1$ replaced by $y \in Z_0(s)$.

Note that the equivalence of $0 \in Z_0(s)$ and $Z_0(s) \neq \emptyset$ in item (*i*) follows from the Pareto-monotonicity property in Corollary 11, since 0 is dominated by any vector in $Y_{[0,\sigma]}$.

4.3. The single-constraint case

Let us consider the special case of a single burstiness constraint, namely $m = 1$. Thus, the constraint parameters σ, ρ are scalar-valued, as is the constraint function $d(s,a)$. In that case, the computation of the feasibility sets may be considerably simplified.

When specialized to the scalar case, properties (*i*) – (*ii*) of Corollary 11 (Pareto monotonicity and closedness) imply that $Z_t(s)$ is either an empty set, or a closed interval of the form $[0, y^*_t(s)]$ for some scalar $y^*_t(s) \in [0,\sigma]$. Hence, it suffices to compute the scalar variables $y^*_t(s)$ to determine the sets $Z_t(s)$.

It will be convenient to define $y^*_t(s) = -\infty$ if $Z_t(s) = \emptyset$. Thus,

$$y^*_t(s) = \sup Z_t(s) = \begin{cases} \max Z_t(s) & : \quad Z_t(s) \neq \emptyset \\ -\infty & : \quad \text{otherwise.} \end{cases} \tag{27}$$

We refer to the scalars $y^*_t(s) \in [0,\sigma] \cup \{-\infty\}$ as the *feasibility variables*. Corollary 13 implies the following explicit recursive formulae for computing these variables.

Proposition 15. *Let $m = 1$. Then $y^*_N(s) = \sigma$, $s \in S$, and for $t = N-1, \ldots, 0$,*

$$y^*_t(s) = \max_{a \in A_s} \min_{s' \in S'(s,a)} y^*_t\big(s,a,s'\big), \quad s \in S, \tag{28}$$

where $y^*_t(s,a,s') \triangleq f_\sigma(y^*_{t+1}(s') - d(s,a) + \rho)$, and $f_\sigma(x) = \min\{x,\sigma\}$ if $x \geq 0$, while $f_\sigma(x) = -\infty$ if $x < 0$.

Proof. We specialize Corollary 13 to the scalar case. Clearly $Y^*_N(s) = [0,\sigma]$, so that $y^*_N(s) = \sigma$.

Consider $t < N$, and recall that $Z_t = \mathcal{F}(Z_{t+1})$ by Corollary 13. That is, by definition of $\mathcal{F}$,

$$Z_t(s) = \bigcup_{a \in A_s} \bigcap_{s' \in S'(s,a)} Y_t\big(s,a,s'\big), \tag{29}$$

$$Y_t(s,a,s') = \left\{ y \in [0,\sigma] : y'(s,y,a) \in Z_{t+1}(s') \right\}. \tag{30}$$

For convenience, define $[0,-\infty] = \emptyset$. Let us first establish that $Y_t(s,a,s') = [0, y^*_t(s,a,s')]$. Suppose $Z_{t+1}(s') = \emptyset$, so that $y^*_{t+1}(s') = -\infty$ by (27). Then, by (30) and the definition of $y^*_t(s,a,s')$, respectively, we have that $Y_t(s,a,s') = \emptyset$ and $y^*_t(s,a,s') = -\infty$. Suppose next that $Z_{t+1}(s') = [0, y^*_{t+1}(s')] \neq \emptyset$. Then, by (30) and (13), $Y_t(s,a,s') = \{y \in [0,\sigma] : y + d(s,a) - \rho \leq y^*_{t+1}(s')\}$, which yields, by rearranging, $Y_t(s,a,s') = [0, f_\sigma(y^*_{t+1}(s') - d(s,a) + \rho)] \triangleq [0, y^*_t(s,a,s')]$.

Eq. (28) now follows from (24) upon noting that $[0,a] \cap [0,b] = [0, \min\{a,b\}]$ and $[0,a] \cup [0,b] = [0, \max\{a,b\}]$. □

Proposition 15 provides an explicit backward recursion for computing the scalar feasibility variables $y^*_t(s)$, hence the feasibility sets $Z_t(s) = [0, y^*_t(s)]$. The computational complexity is similar to that of finite-horizon dynamic programming over the original MDP model $\mathcal{M}$ with state space S.

4.4. Computational aspects: The discrete-state model

We next consider computation of the feasibility sets $Z_t \subset S \times Y_{[0,\sigma]}$ in the general case of $m \geq 2$ burstiness constraints. Here the elements of $Z_t(s)$ are m-dimensional vectors in the continuous set $Y_{[0,\sigma]}$. While these sets can be represented by their Pareto front (given their Pareto-monotone structure, per Corollary 11), this front is still a surface in the m-dimensional space. Thus, this property is not as useful as in the single-constraint case, and does not lead to an explicit calculation over a bounded number of points in general. We therefore specialize to the case where the constraint parameters take values in a discrete grid. Under this mild assumption, the y component of the state becomes discrete, and all calculations need to be carried out over a finite grid.

We thus specify here the following assumption.

Assumption 1 (Grid-valued burstiness constraints)**.** For each constraint $i = 1, \ldots, m$, the constraint parameters σ_i, ρ_i and the values of the constraint function $d_i(s,a)$ are all integer multiples of a common constant $\beta_i > 0$. Denote $n_i = 1 + \sigma_i/\beta_i$.

Recall that the sequence of burstiness variables $\{y_t\}$ (as defined Section 3.1) starts with the initial condition $y_0 = 0$. Hence, by (8) and Assumption 1, y_t remains in the positive grid

$$Y^G_+ = \prod_{i=1}^m \left\{ k_i \beta_i,\ k_i \in \mathbb{N}_0 \right\} \subset \mathbb{R}^m_+ . \tag{31}$$

We can therefore restrict attention to burstiness variables that take values in Y^G_+, and to corresponding burstiness sets

$$Z_t(s) \subset Y_{[0,\sigma]} \cap Y^G_+ \triangleq Y^G_{[0,\sigma]} .$$

Recalling the notation $n_i = 1 + \sigma_i/\beta_i$, the cardinality of the set $Y^G_{[0,\sigma]}$ is $\prod_{i=1}^m n_i$, which is finite, but grows exponentially in m.

The results of Section 4.2 remain valid, with $Y_{[0,\sigma]}$ replaced by its discrete version $Y^G_{[0,\sigma]}$. The backward recursion over Z_t is carried out with the same stage-complexity as standard finite-horizon dynamic programming, however at each stage the computations involve sets of cardinality $|Y^G_{[0,\sigma]}|$ rather than scalars.

Remark 2. If the given model data do not satisfy Assumption 1, then, observing the constraint Eq. (4), a conservative approximation for feasibility properties can be obtained by rounding up the values of $d_i(s,a)$ to a larger grid point, and rounding down the values of ρ_i and σ_i to a smaller grid point. The level of approximation can naturally be controlled through the selected grid density.

5. Feasibility analysis: Infinite horizon

We turn to consider the question of (σ,ρ)-feasibility over the infinite time horizon, $\mathcal{T}=\mathbb{N}_0$. The development proceeds similarly to that of the previous section: We first consider the feasibility indicator functions, and develop the main results in these terms. We then follow by reformulating the results in terms of corresponding feasibility sets, and close by considering the computational complexity under the discrete model that is induced by Assumption 1.

As may be expected, the feasibility indicator function for the infinite horizon case is characterized as a solution to a fixed-point dynamic programming equation. While the solution to this equation is not unique, the required solution will be characterized as the *maximal* solution of this equation, and a value iteration procedure that converges to this solution will be established. We note the these properties are similar to those of MDPs with expected total reward criterion, and specifically with the so-called negative models in this category (Puterman, 1994, Chapter 7.3). These relations will be further discussed below.

5.1. The limiting feasibility function

Extending the definition of the feasibility indicator function v^*_N in (14) to the infinite-horizon case, let

$$\begin{aligned} v^*(s,y) &= \mathbf{1}\{\text{the } (\sigma,\rho)\text{-constraints are feasible from initial state } (s,y)\} \\ &= \mathbf{1}\{\exists \pi\in\tilde{\Pi} \text{ s.t. } y_t\le\sigma \ \ \forall t\ge 0,\ P^{\pi}_{s,y}\text{-a.s.}\}, \\ &\qquad (s,y)\in S\times\mathbb{R}^m_+ . \end{aligned}$$

We refer to v^* as the limiting feasibility function.

Recall the definition of the operator $\mathcal{L}:\mathcal{V}_B\to\mathcal{V}_B$ in (17), which is repeated here for convenience:

$$\mathcal{L}(v)(s,y)=\mathbf{1}\{y\in Y_{[0,\sigma]}\}\cdot\max_{a\in A_s}\min_{s'\in S'(s,a)} v(s',y'(s,y,a)),\quad (s,y)\in S\times\mathbb{R}^m_+ . \tag{32}$$

We proceed to characterize v^* as a solution of the fixed point equation

$$v=\mathcal{L}(v)\,. \tag{33}$$

This equation is analogous to the standard Bellman optimality equation for MDPs. In the present context, we refer to it as the *feasibility equation*.

Notably, the solution to the feasibility equation above is not unique; in particular, $v=0$ is always a solution. We therefore resort to the following characterization of v^*.

Theorem 16. *The limiting feasibility function v^* is the maximal solution of the feasibility equation $v=\mathcal{L}(v)$ in $\mathcal{V}_B$. That is, v^* is a solution of this equation, and $v^*\ge v$ (point-wise) for any other solution $v\in\mathcal{V}_B$.*

Proof. We first demonstrate that v^* is a solution of $v=\mathcal{L}(v)$, which follows by standard arguments: Let v^*_1 be defined similarly to v^*, but starting from $t=1$ rather than $t=0$. Since the problem data is stationary and the horizon is infinite, we have that $v^*_1=v^*$. On the other hand, arguing as in Proposition 6, $v^*=\mathcal{L}(v^*_1)$. Hence $v^*=\mathcal{L}(v^*)$.

We next claim the following: If $v\in\mathcal{V}_B$ satisfies $v\le\mathcal{L}(v)$, then $v\le v^*$. To that end, define $v_k=\mathcal{L}^k(v)$ for $k\ge 0$. Noting the monotonicity property of $\mathcal{L}$ in Lemma 5 (*iii*), $v\le\mathcal{L}(v)$ implies that $v_k\le v_{k+1}$. Consider a state $(s,y)\in S\times\mathbb{R}^m_+$. By monotonicity of the sequence (v_k), and recalling that each function v_k takes values in $\{0,1\}$, we have that $v(s,y)=1$ implies that $v_k(s,y)=1$ for all $k\ge 0$. Next, as shown below, $v_k(s,y)=1$ for $k\ge 0$ implies that the (σ,ρ)-constraints are feasible from initial state (s,y). By definition of v^*, it follows that $v^*(s,y)=1$. Thus, for any state (s,y), $v(s,y)=1$ implies $v^*(s,y)=1$, hence $v\le v^*$.

Since the last claim applies, in particular, to any solution of $v=\mathcal{L}(v)$, it follows that v^* is the largest solution of $v=\mathcal{L}(v)$, as claimed.

It remains to show that, indeed, $v_k(s,y)\equiv\mathcal{L}^k(v)(s,y)=1$ for $k\ge 0$ implies that the (σ,ρ)-constraints are feasible from initial state $(s_0,y_0)=(s,y)$. We argue similarly to the proof of Theorem 10. By definition of $\mathcal{L}$, $\mathcal{L}(v)(s_0,y_0)=1$ implies that $y_0\in Y_{[0,\sigma]}$, and there exists an action a_0 such that $v(s_1,y_1)=1$ w.p. 1. Continuing by induction, it follows that there exists a policy such that (w.p. 1) $v(s_t,y_t)=1$ for $t\ge 0$, which implies that $y_t\in Y_{[0,\sigma]}$ for $t\ge 0$. Thus, the (σ,ρ)-constraints are feasible from the initial state (s,y). □

We may now introduce a value iteration procedure that converges to v^*. As the solution to $v=\mathcal{L}v$ is not unique, the required convergence must depend on a suitable choice of initial conditions.

Theorem 17 (Value Iteration)**.** *Denote by $v_{\max}$ the function $v_{\max}(s,y)=\mathbf{1}\{y\in Y_{[0,\sigma]}\}$, $(s,y)\in S\times\mathbb{R}^m_+$.*

(i) *Let $v_0=v_{\max}$, and let $v_{k+1}=\mathcal{L}(v_k)$ for $k\ge 0$. Then $v_{k+1}\le v_k$, and $\lim_{k\to\infty}v_k=v^*$.*
(ii) *Consequently, if $v^*\le v_0\le v_{\max}$, then $\lim_{k\to\infty}v_k=v^*$ (not necessarily monotonically).*

Proof. (*i*) Observe that $\mathcal{L}(v)(s,y)=0$ for $y\notin\sigma$, it follows that $v_0=v_{\max}\ge\mathcal{L}(v_0)=v_1$, and similarly that $v_0\ge v^*$. Noting the monotonicity property of $\mathcal{L}$ from Lemma 5 (*iii*), it follows that $v_{k+1}\le v_k$ and $v_k\ge v^*$; indeed, $v_{k+1}=\mathcal{L}^k(v_1)\le\mathcal{L}^k(v_0)=v_k$, and $v_k=\mathcal{L}^k(v_0)\ge\mathcal{L}^k(v^*)=v^*$. Now, by the monotone convergence theorem it follows that v_k converges (point-wise) to a limit $v_\infty\ge v^*$. As we show below, the limit v_∞ satisfies the feasibility equation. Since v^* is the maximal solution of this equation, it follows that $v_\infty=v^*$, as claimed.

It remains to show that v_∞ satisfies the feasibility equation, namely, $v_\infty=\mathcal{L}(v_\infty)$. Since $v_{k+1}=\mathcal{L}(v_k)$, then $v_\infty=\lim_{k\to\infty}\mathcal{L}(v_k)$. We argue that the latter limit equals $\mathcal{L}(v_\infty)$. First note that since $v_k(s,y)\in\{0,1\}$, then $v_k\to v_\infty$ implies finite-time convergence at each point (s,y), i.e., $v_k(s,y)=v_\infty(s,y)$ for $k\ge k_{s,y}$ for some finite number $k_{s,y}$. Recalling the definition of $\mathcal{L}$, it may be seen that for each point (s,y), $\mathcal{L}(v)(s,y)$ depends on the value of v at a *finite* number of points (s'_i,y'_i). Observing the finite-time convergence noted above, it follows that $v_k(s'_i,y'_i)=v_\infty(s'_i,y'_i)$ beyond some finite time k_0, which implies that $\mathcal{L}(v_k)(s,y)\to\mathcal{L}(v_\infty)(s,y)$. Hence $\mathcal{L}(v_k)\to\mathcal{L}(v_\infty)$, which concludes the proof of part (*i*).

(*ii*) By monontonicity of $\mathcal{L}$, $v^*\le v_0\le v_{\max}$ implies that $v^*\le\mathcal{L}^k(v_0)\le\mathcal{L}^k(v_{\max})$, and the claimed convergence follows by part (*i*). □

Remark 3. It may actually be shown that the convergence of v_k to v^* is uniform. Specifically, denoting $A_k=\{y\in\mathbb{R}^m_+ : v_k(s,y)=v^*(s,y)\ \forall s\in S\}$, it may be argued that $d(y,A_k)$ converges to 0 uniformly in y (here d is the Euclidean point-to-set distance), by resorting to Dini's uniform convergence theorem. However, these

convergence results are not of practical use in absence of effective bounds on the distance to the solution, such as those available in discounted dynamic programming. We shall therefore turn our attention later to the discrete-state model, where convergence is guaranteed in a bounded number of steps.

The properties of the finite-time feasibility functions noted in Proposition 7 carry over to the limiting feasibility function.

Proposition 18 (Basic Structural Properties)**.** *The following properties are satisfied by v^*. For $s \in S$:*

(i) *(Pareto monotonicity:) $v^*(s,z) \geq v^*(s,y)$ for $z \leq y$, $z \in \mathbb{R}^m_+$.*
(ii) *(Continuity:) The set $\{y \in \mathbb{R}^m_+ : v^*(s,y) = 1\}$ is closed.*

Proof. Let $v_k = \mathcal{L}^k(v_k)$, with $v_0 = v_{\max}$ as in Proposition 17(i). By proposition 7, both properties hold for each v_k. Observing that the sequence v_k is non-increasing, both properties carry over to the limiting function v^*. □

Let us turn to the characterization of feasibility in terms of the limiting feasibility functions. Define the following sets of feasible actions:

$$A^*(s,y) = \left\{a \in A_s : v(s', y'(s,y,a)) = 1,\ \forall s' \in S'(s,a)\right\}, \quad (s,y) \in S \times \mathbb{R}^m_+ . \tag{34}$$

Lemma 19 (Properties of A^*)**.** *For $(s,y) \in S \times \mathbb{R}^m_+$,*

(i) *$A^*(s,x) \subset A^*(s,y)$ if $x \geq y$.*
(ii) *$A^*(s,y) \neq \emptyset$ if, and only if, $v^*(s,y) = 1$.*
(iii) *If $a_t \in A^*(s_t, y_t)$, then $v^*(s_{t+1}, y_{t+1}) = 1$ (w.p. 1). Furthermore, if $y_t \in Y_{[0,\sigma]}$, these two properties are equivalent.*

These properties are analogous to those in Lemma 9, and their proof is similar.

Theorem 20 (Feasibility and feasible policies)**.** *Proposition 8 and Theorem 10 remain valid for the infinite-horizon model, with v_0 replaced by v^*, A_t replaced by A^*, and $t \in \{0, 1 \ldots, N-1\}$ replaced by $t \in \mathbb{N}_0$. That is,*

(i) *The (σ, ρ)-constraints are feasible in $\tilde{\mathcal{M}}$ from initial state $(s,y) \in S \times \mathbb{R}^m_+$ if, and only if, $v^*(s,y) = 1$. Consequently, the (σ,ρ)-constraints are feasible in $\mathcal{M}$ from initial state s if, and only if, $v^*(s,0) = 1$.*
(ii) *A policy $\pi \in \tilde{\Pi}$ is (σ,ρ)-feasible from initial state (s,y) if, and only if, the following holds with $P^\pi_{s,y}$-probability 1:*

$$a_t \in A^*(s_t, y_t), \quad t \in \mathbb{N}_0 . \tag{35}$$

(iii) *Let $\tilde{\pi} \in \tilde{\Pi}$ be a policy that for any history $h_t = (s_0, y_0, a_1, \ldots, s_t, y_t)$ chooses $a_t \in A^*(s_t, y_t)$ if the latter set is nonempty, and otherwise chooses a_t arbitrarily. Then $\tilde{\pi}$ is (σ,ρ)-feasible from any initial state (s,y) such that $v^*(s,y) = 1$.*

Proof. Item (i) follows from the definitions. Item (ii) and (iii) that parallel Theorem 10 follow by the the same line of argument given there, extended to all $t \geq 0$. □

It is evident that the requirement (35) for a feasible policy may be satisfied by a stationary policy in $\tilde{\mathcal{M}}$ with respect to the augmented state, namely, $a_t = f(s_t, y_t)$.

Remark 4. As noted at the beginning of this section, the obtained results concerning the limiting feasibility function v^* parallel those for the class of MDPs with the expected total reward criterion and negative rewards (Puterman, 1994, Chapter 7.3). To see the connection, consider $\log v^*(s,y) \in \{0, -\infty\}$, which can formally be expressed as

$$\log v^*(s,y) = \max_{\pi \in \tilde{\Pi}} E^\pi_{s,y} \sum_{t=0}^{\infty} r_\sigma(y_t),$$

where $r_\sigma(y) = 0$ if $y \leq \sigma$ and $r_\sigma(y) = -\infty$ if $y \not\leq \sigma$, and where we let $0 \cdot (-\infty) \triangleq 0$. This maximization problem conforms with the above-mentioned MDP class, with $-\infty$ added to the allowed reward values.

5.2. Limiting feasibility sets

We briefly interpret the main results of the previous subsection in terms of the corresponding feasibility sets. Let

$$Z^* = \left\{(s,y) \in S \times Y_{[0,\sigma]} : v^*(s,y) = 1\right\}. \tag{36}$$

Thus, v^* is the indicator function of Z^*. Denote $Z^*(s) = \{y \in Y_{[0,\sigma]} : (s,y) \in Z^*\}$.

Recall the operator $\mathcal{F}$ defined in (24)–(25), and its correspondence to the operator $\mathcal{L}$ as described in Lemma 12. Using this correspondence, the following results follow directly from those of the previous subsection.

Proposition 21.

(i) *(Structural properties:) Z^* satisfies the two properties noted in Corollary 11, namely Pareto-monotonicity and closedness.*
(ii) *(Fixed-point solution:) Z^* is the maximal solution of the fixed point equation $Z = \mathcal{F}(Z)$. That is, any other solution Z' of that equation satisfies $Z' \subset Z^*$.*
(iii) *(Value Iteration:) Let $Z_0 = S \times Y_{[0,\sigma]}$, and let $Z_{k+1} = \mathcal{F}(Z_k)$, $k \geq 0$. Then $Z_{k+1} \subset Z_k$, and[1] $Z^* = \lim_{k\to\infty} Z_k$. Consequently, if $Z^* \subset Z_0 \subset S \times Y_{[0,\sigma]}$, then $Z^* = \lim_{k\to\infty} Z_k$.*
(iv) *(Feasibility:) In Theorem 20, the feasibility condition $v_0(s,y) = 1$ is equivalent to $y \in Z^*(s)$, and the feasible actions sets $A^*(s,y)$ may be equivalently expressed as*

$$A^*(s,y) = \left\{a \in A_s : y'(s,y,a) \in Z^*(s'),\ s' \in S\right\}.$$

We next comment on the single-constraint case, $m = 1$. Similarly to Section 4.3, it follows by Proposition 21(i) that $Z^*(s)$ is either empty, or a closed interval of the form $[0, y^*(s)]$ with $y^*(s) \in [0, \sigma]$. If $Z^*(s) = \emptyset$ set $y^*(s) = -\infty$. Then, the value iteration procedure in Proposition 21(iii) may be expressed in terms of these scalar variables $\{y^*_k(s)\}$ using a similar iteration to that in Eq. (28), namely: For $k = 0, 1, 2, \ldots,$

$$y_{k+1}(s) = \max_{a \in A_s} \min_{s' \in S'(s,a)} f_\sigma(y_k(s') - d(s,a) + \rho), \quad s \in S, \tag{37}$$

with initial conditions $y_0(s) = \sigma$, $s \in S$. Consequently, for each $s \in S$ the sequence $(y_k(s))_{k\geq 0}$ is monotone non-increasing, and converges to $y^*(s)$.

5.3. Finite-time convergence

We finally turn our attention to the discrete-state model, which arises under Assumption 1. As discussed in Section 4.4, under this assumption the state space for the $\tilde{\mathcal{M}}$ model reduces to a discrete set $S \times Y^G_+$, and the feasibility sets may be restricted to a finite set:

$$Z^*(s) \subset Y^G_{[0,\sigma]} = \prod_{i=1}^{M} \left\{0, \beta_i, \ldots, (n_i - 1)\beta_i\right\},$$

with cardinality $|Y^G_{[0,\sigma]}| = \prod_{i=1}^m n_i$. The results of the previous two subsections now remain valid with $Y_{[0,\sigma]}$ replaced by $Y^G_{[0,\sigma]}$ throughout.

[1] Recall that the elements of a set limit $Z^* = \lim_{k\to\infty} Z_k$ are such that $z \in Z^*$ if, and only if, there exists a finite integer $m = m(z)$ such that $z \in Z_n$ for all $n \geq m$. If the sequence (Z_k) is monotone decreasing, then the limit is guaranteed to exist, and $\lim_k Z_k = \cap_k Z_k$.

As a consequence, we obtain the following convergence result.

Theorem 22. *Suppose the model data satisfy Assumption 1 (see Section 4.4). Consider the value iteration algorithm $Z_{k+1} = \mathcal{F}(Z_k)$, $k \geq 0$, with initial conditions $Z_0 = S \times Y^G_+$. Then Z_k converges to Z^* in a finite number of steps, which is bounded above by $k_{\max} = |S| \cdot \prod_{i=1}^m n_i$.*

Proof. By Proposition 21, $Z_{k+1} \subset Z_k$, and $Z_{k+1} = Z_k$ implies that $Z_k = Z^*$. Since Z_0 is a finite set with cardinality $k_{\max}$, the conclusion follows. □

A similar conclusion holds of course if the iteration is carried out in terms of the feasibility indicator functions, namely as in Theorem 17, with initial conditions $v_{\max}(s,y) = \mathbf{1}\{y \in Y^G_{[0,\sigma]}\}$, $(s,y) \in S \times Y^G_+$.

We note that finite-time convergence is not typical in value iteration algorithms. It holds here due to discrete ($\{0,1\}$-valued) nature of the value functions.

While the derived upper bound on the computation complexity may be substantial, this result does provide a finite algorithm for computing the feasibility sets, from which we can obtain the sets of feasible actions $A^*(s,y)$ that characterize feasible policies. Of course, the number of iterations to convergence may well be smaller than the bound in specific instances (e.g., see the illustrative example in the Appendix).

6. Constrained optimization

We proceed to consider the constrained optimization problem, namely the optimization of standard reward functionals such as (2)-(3), subject to the imposed burstiness constraints. Our approach here builds upon the previous results that provide a characterization of feasible policies, in terms of allowed actions in each augmented state $\tilde{s} = (s,y)$.

To apply this approach, it is required to compute the feasible actions sets, namely $A_t(s,y)$ or $A^*(s,y)$. Finite algorithms for this computation were derived only for the discrete-state model, in which the constraint parameters satisfy Assumption 1. While we do not explicitly impose Assumption 1 in the following, it is implicit in that a practical implementation requires this assumption. (Alternatively, if the given model parameters do not satisfy this assumption, then conservative estimates on the feasible actions sets can be computed by slightly modifying the parameters to fit this assumption; see Remark 2 in Section 4.4.)

When Assumption 1 is satisfied, the continuous set $Y_{[0,\sigma]}$ in the following may be replaced by the finite set $Y^G_{[0,\sigma]}$.

6.1. Finite-horizon optimal policies

Consider the problem of maximizing the reward functional (2), over the set of (σ,ρ)-feasible policies over the finite time horizon $\mathcal{T} = \{0,1\ldots,N-1\}$. As before, it will be convenient to treat the equivalent problem within the state-augmented model $\tilde{\mathcal{M}}$, namely

$$J^*_N(s,y) = \max_{\pi \in \tilde{\Pi}_{\sigma,\rho}(s,y)} E^\pi_{s,y}\left(\sum_{t=0}^{N-1} r(s_t,a_t) + r_N(s_N)\right), \tag{38}$$

where $(s,y) \in S \times Y_{[0,\sigma]}$, and $\tilde{\Pi}_{\sigma,\rho}(s,y)$ is the set of (σ,ρ)-feasible policies in $\tilde{\mathcal{M}}$ from initial state (s,y), as per Definition 3. The implications for the original model $\mathcal{M}$ are immediate, as observed in Section 3.2. We recall that for correspondence with the original problem in M, it suffices to consider the initial value $y = 0$.

Suppose that the sets $Z_t \subset S \times Y_{[0,\sigma]}$ and $A_t(s,y)$, $(s,y) \in S \times Y_{[0,\sigma]}$ have been computed and are available for all $t \in \mathcal{T}$. Consider the following finite-horizon dynamic programming recursion for computing the constrained value functions $V_N,\ldots,V_0$: Let $V_N(s,y) = r_N(s)$, $(s,y) \in Z_N$, and for $t = N-1,\ldots,0$,

$$V_t(s,y) = \max_{a \in A_t(s,y)} Q_t(s,y,a), \quad (s,y) \in Z_t,$$

where

$$Q_t(s,y,a) = r(s,a) + \sum_{s' \in S} p(s'|s,a) V_{t+1}(s', y'(s,y,a)).$$

Recall that the (σ,ρ) constraints are feasible from initial state (s,y) if, and only if, $(s,y) \in Z_0$ (Corollary 14). The following result provides the constrained optimal policies for feasible initial states.

Theorem 23. *Let $(s,y) \in Z_0$. Then $J^*_N(s,y) = V_0(s,y)$, and any policy that satisfies the following is an optimal policy in $\tilde{\Pi}_{\sigma,\rho}(s,y)$: If $(s_t,y_t) = (s,y) \in Z_t$, choose*

$$a_t \in \operatorname*{argmax}_{a \in A_t(s,y)} Q_t(s,y,a).$$

If $(s_t,y_t) \notin Z_t$, then a_t may be chosen arbitrarily (since the constraints cannot be satisfied from such states).

Proof. The characterization of feasible policies in Theorem 10 and Corollary 14 implies that a policy is feasible if, and only if, it chooses $a_t \in A_t(s_t,y_t)$ in every stage. Note that, by Lemma 9, such a choice implies that $(s_{t+1},y_{t+1}) \in Z_{t+1}$ (w.p. 1), so that $A_{t+1}(s_{t+1},y_{t+1})$ is nonempty. Thus, w.p. 1, no state is encountered for which $A_t(s_t,y_t) = \emptyset$. Restricting attention to actions $a_t \in A_t(s_t,y_t)$, the optimality result now follows by standard dynamic programming arguments. □

It is clear that the requirement in Theorem 23 can be satisfied by a *deterministic Markov policy* with respect to the augmented state $\tilde{s} = (s,y)$. Thus, the set of deterministic Markov policies is sufficient for the finite-horizon constrained optimality problem.

We close this subsection by observing that the constrained value functions are Pareto-monotone decreasing, namely $V_t(s,y) \leq V_t(s,x)$ if $y \geq x$, which follows from the corresponding monotonicity property of the actions sets $A_t(s,y)$, stated in Lemma 9(i).

6.2. Infinite-horizon optimal policies

We consider next the infinite-horizon constrained optimization problem, namely the maximization of a reward functional such as (3), subject to (σ,ρ)-burstiness constraints over the time horizon $\mathcal{T} = \mathbb{N}_0$.

For the infinite horizon problem it will be useful to first *prune* the model, namely, remove states and actions that are not in the feasible sets. The pruned model already adheres to the burstiness constraints, so that the constrained optimization problem becomes a standard, unconstrained MDP problem over this reduced model.

Recall the definitions of Z^* and A^* in (36) and (20). Consider the augmented-state model $\tilde{\mathcal{M}}$, and define a pruned version $\check{\mathcal{M}}$ thereof by restricting the state and actions sets as follows:

$$\check{S} = Z^* = \{(s,y) : y \in Z^*(s)\} \subset S \times Y_{[0,\sigma]}.$$

$$\check{A}_{s,y} = A^*(s,y) \subset A_s, \quad (s,y) \in \check{S}.$$

Observe that the pruned model $\check{\mathcal{M}}$ is well defined, in the sense that for any state $(s,y) \in \check{S}$: (I) The action set $\check{A}_{s,y}$ is nonempty. (II) If $a_t \in \check{A}_{s,y}$, then $(s_{t+1},y_{t+1}) \in \check{S}$ (w.p. 1). Both properties follow by Lemma 19 (recall that v^* is the indicator of Z^*).

Let $\check{\Pi}$ denote the set of general policies for the model $\check{\mathcal{M}}$, and recall that $\tilde{\Pi}$ is the set of policies in the model $\tilde{\mathcal{M}}$.

Definition 4. We say that a policy $\pi \in \tilde{\Pi}$ coincides with a policy $\check{\pi} \in \check{\Pi}$ if $\pi(h_t) = \check{\pi}(h_t)$ for any history h_t of states and actions in $\check{\mathcal{M}}$, namely $h_t \in K^{t-1} \times \check{S}$, $t \geq 0$, where $K = \{(s,y,a) : (s,y) \in \check{S}, a \in \check{A}_{s,y}\}$.

Lemma 24. *Let $\tilde{\Pi}(s,y;\sigma,\rho) \subset \tilde{\Pi}$ denote the set of (σ,ρ)-feasible policies from initial state (s,y) in $\tilde{\mathcal{M}}$. Then $\pi \in \tilde{\Pi}(s,y;\sigma,\rho)$ if, and only if, it coincides with some policy $\check{\pi} \in \check{\Pi}$.*

Proof. The claim follows from the characterization of feasible policies in Theorem 20 (recall that v^* is the indicator of Z^*). □

Observe that $\tilde{\Pi}(s,y;\sigma,\rho)$ is nonempty if, and only if, $(s,y) \in Z^*$. Consider the constrained optimization problem over $\tilde{\mathcal{M}}$:

$$\tilde{J}(s,y) = \max_{\pi \in \tilde{\Pi}(s,y;\sigma,\rho)} E^{\pi}_{s,y}\left(\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\right), \quad (s,y) \in Z^*. \tag{39}$$

Consider further the following (unconstrained) optimization problem over $\check{\mathcal{M}}$:

$$\check{J}(s,y) = \max_{\check{\pi} \in \check{\Pi}} E^{\check{\pi}}_{s,y}\left(\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\right), \quad (s,y) \in \check{S} = Z^*. \tag{40}$$

The following result follows directly from Lemma 24.

Theorem 25. *A policy $\pi \in \tilde{\Pi}(s,y;\sigma,\rho)$ is optimal for the constrained optimization problem in (39) if, and only if, it coincides with a policy $\check{\pi} \in \check{\Pi}$ that is optimal for the (unconstrained) optimization problem (40) over the pruned model $\check{\mathcal{M}}$.*

The optimization problem in (40) is a standard MDP problem, admitting an optimal policy that is stationary and deterministic. This implies that an optimal solution to the constrained problem (39) is obtained by such a policy over the augmented state space $\tilde{S}$. The solution to (40) may be found by standard dynamic programming algorithm, namely value iteration, policy iteration, or linear programming.

The above discussion and conclusions apply equally well to other infinite-horizon reward criteria, such as the expected average reward or risk-sensitive problems. In either case, the burstiness-constrained problem may be solved as an unconstrained MDP problem over the pruned model, with the obtained policy transferred to the constrained model.

7. An illustrative example

We present here a simple example to illustrate the feasibility-related computations in the infinite-horizon case. The specific queuing model was chosen specifically so that some of the results are apparent, which serves to validate the computational scheme. Hence, this model is not meant to represent a realistic situation. We start by considering feasibility computations for a single burstiness constraint, and then add a second one. The third subsection address a constrained optimization problem, namely optimal admission control under a burstiness constraint on the number of dropped arrivals. All calculations were implemented in MATLAB®.

7.1. A single burstiness constraint

We consider a transmission-queue model, as illustrated in Fig. 1. Packets arrive to a transmission system in discrete time slots $t = 0, 1, \ldots$, with their number b_t determined by an i.i.d. distribution p_B with finite support $\mathrm{supp}(p_B)$. A router decides how many of these arrivals may be admitted to the queue, with the rest to be discarded. This decision is made before observing the batch size b_t. The accepted packets are queued in a transmission buffer, with a maximal capacity of $Q_{\max}$ packets. Let q_t denote the number of packets in the buffer at time t. The transmitter transmits a single packet per time slot, as long as the buffer is not empty. Note that the transmitter must transmit if there is a packet waiting in the buffer. The sequence (Tx_t) of transmitted packets is subject to a single burstiness constraint with parameters (σ_1, ρ_1).

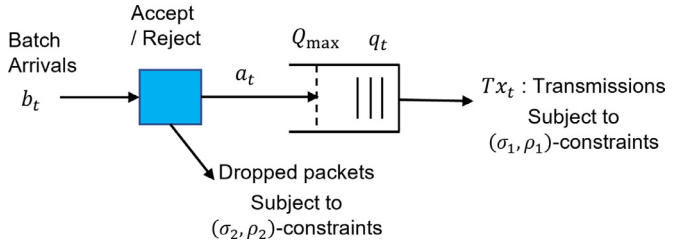

Fig. 1. A transmission queue with two burstiness constraints.

The system state at time t is $s_t = q_t$, where $q_t \in \{0, \ldots, Q_{\max}\}$ is the buffer content. Recall that $b_t \in \mathrm{supp}(p_B)$ denotes the number of arrivals at that time slot. The resulting state dynamics is described by

$$q_{t+1} = q_t - \mathbf{1}\{q_t > 0\} + \min(a_t, b_t), \quad b_t \sim p_B,$$

where the action $0 \le a_t \le \min(Q_{\max} - q_t, b_{\max})$ is the allowed number of admitted packets, which is upper bounded by the buffer capacity and as well as the maximal batch size. The actual number admitted is the minimum between a_t and the actual batch size b_t.

The transmission sequence is given by $Tx_t = \mathbf{1}\{q_t > 0\} \in \{0, 1\}$, and is subject to the constraints

$$\sum_{t=t_1}^{t_2} Tx_t \le (t_2 - t_1 + 1)\rho_1 + \sigma_1, \quad t_2 \ge t_1 \ge 0.$$

Note that in our standard notation, $d_1(s_t, a_t) = Tx_t$. The burstiness variables y_t are accordingly are accordingly given by $y_{t+1} = (y_t + Tx_t - \rho_1)^+$.

Prior to simulation, let us determine analytically the set of feasible augmented states $\tilde{s} = (q, y)$ for this model. To avoid trivialities, suppose that $\rho_1 \le 1$ and $E(b_t) > 0$. Recall that feasibility of the constraint requires $y_t \le \sigma_1$ for all $t \ge 0$. Suppose that at time t, $s_t = q$ and $y_t = y$. Even if no further packets are admitted, the transmitter must transmit at least the q packets waiting in the buffer, over the time period $t, \ldots, t + q - 1$. At the end of this period, we will have $y_{t+q} = y + (1 - \rho_1)q$. This implies the feasibility condition $y + (1 - \rho_1)q \le \sigma_1$, and the feasibility set is accordingly given by

$$Z^*(q) = \{y \in [0, \sigma_1] : y \le \sigma_1 - (1 - \rho_1)q\}.$$

This allows to confirm the simulation results. Note that the arrival distribution p_B does not affect the feasibility set.

A specific set of parameters used in this example is: $p_B(b) = 0.5$ for $b \in \{0, 2\}$, $Q_{\max} = 10$, $\rho_1 = 0.6$ and $\sigma_1 = 5$. Observe that Assumption 1 is satisfied, with grid separation $\beta_1 = 0.2$. Thus, the burstiness variables y_t can be limited to the discrete set

$$Y^G_{[0,\sigma]} = \{0, 0.2, 0.4, \ldots, 5\} \subset Y_{[0,\sigma]} = [0, 5].$$

The feasibility value iteration algorithm of Theorem 17 was implemented over the state space $S \times Y^G_{[0,\sigma]}$ of cardinality $11 * 21$. The algorithm converged to the anticipated feasibility function, namely,

$$v^*(q, y) = \mathbf{1}\{y + (1 - 0.6)q \le 5\}, \quad q \in \{0, \ldots, 10\}, \ y \in Y^G_{[0,\sigma]}.$$

More explicitly, the obtained values of $v^*(q, y)$ are shown below, where the vertical axis corresponds to $q = 0, 1, \ldots, 10$, and the horizontal to $y = 0,\ 0.2,\ 0.4, \ldots, 5.0$.

Convergence was reached after 11 iterations, which equals the number of queue positions $(Q_{\max} + 1)$. The same equality was observed for other values of the system parameters. That can be explained by noting that the optimal action (with respect to the constraint) is always $a_t = 0$. With these actions the queue empties from any initial state after $Q_{\max} + 1$ steps at most, reaching the

Table 1
Obtained values of the feasibility function $v(q, y)$. The '1' entries indicate the feasible states.

q \y	0 ← $y = 0.0, 0.2, \ldots, 5.0$ → 5																						
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2
Computation of the feasibility threshold variables $y_k(q, b)$, for $b = 0$. Here k denotes the iteration number, till convergence. The final values $y^*(q, b)$ are shown in bold.

q \k	0	1	2	3	4	5	6	7	8	9	10	11
0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	**5.0**
1	5.0	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	**5.0**
2	5.0	4.6	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	4.2	**4.2**
3	5.0	4.6	4.2	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8	**3.8**
4	5.0	4.6	4.2	3.8	3.4	3.4	3.4	3.4	3.4	3.4	3.4	**3.4**
5	5.0	4.6	4.2	3.8	3.4	3.0	3.0	3.0	3.0	3.0	3.0	**3.0**
6	5.0	4.6	4.2	3.8	3.4	3.0	2.6	2.6	2.6	2.6	2.6	**2.6**
7	5.0	4.6	4.2	3.8	3.4	3.0	2.6	2.2	2.2	2.2	2.2	**2.2**
8	5.0	4.6	4.2	3.8	3.4	3.0	2.6	2.2	1.8	1.8	1.8	**1.8**
9	5.0	4.6	4.2	3.8	3.4	3.0	2.6	2.2	1.8	1.4	1.4	**1.4**
10	5.0	4.6	4.2	3.8	3.4	3.0	2.6	2.2	1.8	1.4	1.0	**1.0**

empty queue state which is absorbing. This is of course specific to the present model.

For the same parameters, computation of the feasible set was repeated using the simpler single-constraint procedure in Eq. (37). Here the data is more compact and we can show the progression of the computation up to convergence. Table 2 shows the feasibility threshold variables $y_k(q)$, where the vertical axis again corresponds to $q = 0, 1, \ldots, 10$, and the horizontal axis to the iteration number $k = 0, \ldots, 11$. The final value $y^*(q)$ is shown in bold. The obtained results can be explained as in the previous paragraph.

7.2. Two burstiness constraints

We next consider the same example with the addition of a second burstiness constraint, that limits the number of rejected packets at the entrance (see Fig. 1). Specifically, the number of rejections at time t is $b_t - a_t$, and the rejection sequence is subject to the (σ_2, ρ_2) burstiness constraint

$$\sum_{t=t_1}^{t_2} (b_t - a_t) \le (t_2 - t_1 + 1)\rho_2 + \sigma_2, \quad t_2 \ge t_1 \ge 0\,.$$

In our standard notation, $d_2(s, a) = d_2((q, b), a) = b - a$.

The results here are harder to predict, and we focus here on the convergence time. The same set of parameters was used as before, with the addition of the second constraint parameters $\sigma_2 = 5$, $1 \le \rho_2 \le 2$, with grid separation $\beta_2 = 0.2$. Here $Y^G_{[0,\sigma]} = \{0, 0.2, 0.4, \ldots, 5\}^2$, and the augmented state space $S \times Y^G_{[0,\sigma]}$ is of cardinality $11 * 21 * 21$.

The number of iterations to convergence, k^*, is shown for several values of the rate parameter ρ_2:

$\rho_2 =$	2.0	1.8	1.6	1.4	1.2	1.0
$k^* =$	10	13	13	15	37	27

For $\rho_2 = 2$, the (σ_2, ρ_2)-constraint has no effect (as the maximal number of arrivals is 2), and as expected the results are identical to those of the previous subsection. As ρ_2 is decreased, the limitation of this constraint enters into play, more elements of v^* turn from 1 to 0, and the number of iterations to convergence somewhat increases. Finally, for $\rho_2 = 1.2$ the two constraints become contradictory, and the iteration converges to the trivial solution $v^* \equiv 0$, i.e., no state is feasible. Decreasing ρ_2 further to $\rho_2 = 1$ leads to the same trivial solution, except that it is reached faster due to the stricter limitation imposed at each step.

Evidently, the number of iterations required for convergence is orders of magnitude lower than the bound in Theorem 22. While this may well be due to the special structure of the present model, it does indicate that an orderly model structure as induced by specific applications may well lead to a shorter convergence time of the proposed value iteration algorithm.

7.3. Burstiness-constrained optimal admission control

We return to the single-constraint model of Section 7.1, and consider a problem of optimal admission control under this constraint. Admission control to a single queue is a classical and well studied problem (e.g., Naor, 1969; Stidham, 1985). With a reward function that balances throughput versus delay, the optimal admission policy typically turns out to be a threshold policy. As noted, our purpose here is to illustrate the computation procedure and the form of the results rather than the analysis of a full-fledged model.

The instantaneous reward function considered is of the standard form $r_t = Tx_t - cq_t$, where the first term denotes the number of transmitted packets at t, accounting for throughput, and the second term is a constant $c > 0$ times the queue size, which accounts for queueing delay (or holding cost). Since $Tx_t = \mathbf{1}\{q > 0\}$ in our model, we obtain the reward function (with state $s = q$)

$$r(q, a) = \mathbf{1}\{q > 0\} - cq\,.$$

We aim to maximize the discounted reward in (3), subject to the (ρ_1, σ_1)-burstiness constraint on the transmission sequence (Tx_t) (see Fig. 1). We use the same set of model parameters as before, namely $p_B(b) = 0.5$ for $b \in \{0, 2\}$, $Q_{\max} = 10$, $\rho_1 = 0.6$ and $\sigma_1 = 5$. We set here the discount factor to $\gamma = 0.98$ and the cost parameter to $c = 0.1$.

For reference, we first calculate optimal unconstrained policy, which is shown below.

q:	0	1	2	3	4	5	6	7	8	9	10
$a^*(q)$:	2	2	2	2	2	2	2	2	1	0	0

As may be expected this turns out to be a threshold policy, with a threshold $\theta = 9$. Thus, for $q \le 7$ the optimal number of allowed admissions is 2 (the maximal value possible), for $q = 8$ one admission is optimal, and for $q \ge 9$ no admissions are allowed.

We next consider the optimal policy subject the (σ_1, ρ_1)-burstiness constraints. We follow the procedure outlined in Section 6.2. We first compute the feasible augmented states $\tilde{s} = (q, y)$, which was already done above (see Table 1). We next trim the model by restricting the state space to the feasible augmented states, and further removing all actions that lead from a feasible state to an infeasible one.

The resulting model is a standard discounted MDP, which can be solved using standard methods. We used policy iteration, which converged within 5 iterations. The optimal policy is shown in Table 3.

It may be seen that for most fixed values of y, the obtained policy is a threshold policy, with the threshold value deceasing in y. That is, the admission policy becomes less aggressive as

Table 3
The optimal constrained policy. The numeric values are the optimal actions $a^*(q, y) \in \{0, 1, 2\}$, and the infeasible states are indicated here by a '*'.

q \y | 0 ←	$y = 0.0, 0.2, \ldots, 5.0$	→ 5
0	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 0 0 * * *
2	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 0 0 0 0 * * * * *
3	2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 0 0 0 0 0 * * * * * *
4	2 2 2 2 2 2 1 1 1 1 1 1 1 0 0 0 0 * * * * * * * * *
5	2 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 * * * * * * * * * *
6	1 1 0 0 0 0 0 0 0 0 0 0 0 0 * * * * * * * * * * * *
7	0 0 0 0 0 0 0 0 0 0 0 0 * * * * * * * * * * * * * *
8	0 0 0 0 0 0 0 0 0 * * * * * * * * * * * * * * * * *
9	0 0 0 0 0 0 0 * * * * * * * * * * * * * * * * * * *
10	0 0 0 0 0 *

the transmission burstiness variable increases. An exception to the threshold structure may be seen for the higher values of y (last 5 columns), which may be attributed to some relevant actions becoming infeasible. For example, for $q = 1$ and $y = 4.4$, the only feasible action is $a = 0$, which is hence the one chosen.

8. Conclusion

We considered in this paper the incorporation of (σ, ρ)-type burstiness constraints within the framework of finite state and action Markov Decision Processes. Utilising the equivalent formulation of the constraints in terms of burstiness variables, the burstiness constraints were expresses as limitations on the state values in an augmented-state model. The analysis of this model provided a characterization of feasible policies, which allowed to find burstiness-constrained optimal policies using standard dynamic programming algorithms within the augmented model.

For the infinite-horizon model, the characterization of feasibility relies on a converging value iteration procedure. By requiring the constraint parameters to lie on a common grid, the procedure was shown to converge in a bounded number of steps, thus leading to a finite algorithm. While the derived bound on the number of iterations to convergence is large, the actual number may well be smaller in specific cases, as hinted here by a structured example.

An apparent limitation of the presented results is the increase in the state-space due to the incorporation of the burstiness variables. With multiple inter-dependent constraints, the state space essentially increases exponentially in the number of constraints. For large problems, some sub-optimal procedures may be needed to implement the constraints, possibly utilizing the system structure and weak inter-dependence among different constraints.

The example studied in Section 7 has hinted that certain structural properties of optimal unconstrained policies (here the threshold structure of admission policies) may carry over to the burstiness-constrained case. Is should be of interest investigate this property theoretically for specific systems.

Finally. the burstiness constraints studied here are hard constraints, namely are required to hold with probability one. An obvious extension is to consider soft constraints, which need to hold with high probability. The precise formulation of high-probability (σ, ρ)-type burstiness constraints and the relevant analysis are left as topics for further study.

References

Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In *International conference on machine learning* (pp. 22–31).
Altman, E. (1999). *Constrained markov decision processes*. CRC Press.
Anantharam, V., & Konstantopoulos, T. (1993). An optimal flow control scheme that regulates the burstiness of traffic subject to delay constraints. In *Proceedings of 32nd IEEE conference on decision and control* (pp. 3606–3610).
Barabasi, A.-L. (2005). The origin of bursts and heavy tails in human dynamics. *Nature, 435*(7039), 207–211.
Bellman, R. E. (1957). *Dynamic programming*. Princeton University Press.
Bertsekas, D. P. (2012). *Dynamic programming and optimal control*: Vol. 1 & 2. Athena Scientific.
Borkar, V., & Jain, R. (2014). Risk-constrained Markov decision processes. *IEEE Transactions on Automatic Control, 59*(9), 2574–2579.
Chang, C.-S. (2000). *Performance guarantees in communication networks*. Springer.
Chang, H. S. (2015). Random search for constrained Markov decision processes with multi-policy improvement. *Automatica, 58*, 127–130.
Cruz, R. L. (1991). A calculus for network delay. I. Network elements in isolation. *IEEE Transactions on Information Theory, 37*(1), 114–131.
Dufour, F., & Prieto-Rumeau, T. (2013). Finite linear programming approximations of constrained discounted Markov decision processes. *SIAM Journal on Control and Optimization, 51*(2), 1298–1324.
Eckberg, A. E. (1985). Approximations for bursty (and smoothed) arrival queueing delays based on generalized peakedness. In *Proc. 11th Int. teletraffic congress* (pp. 331–335).
(2002). In E. A. Feinberg, & A. Shwartz (Eds.), *Handbook of Markov decision processes: methods and applications*. Kluwer Academic.
Fidler, M., & Rizk, A. (2014). A guide to the stochastic network calculus. *IEEE Communications Surveys & Tutorials, 17*(1), 92–105.
Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research, 16*(1), 1437–1480.
Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Yang, Y., & Knoll, A. (2022). A review of safe reinforcement learning: Methods, theory and applications. arXiv:2205.10330.
Heffes, H., & Lucantoni, D. (1986). A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications, 4*(6), 856–868.
Jaskiewicz, A., & Nowak, A. S. (2019). Constrained Markov decision processes with expected total reward criteria. *SIAM Journal on Control and Optimization, 57*(5), 3118–3136.
Jiang, H., & Dovrolis, C. (2005). Why is the internet traffic bursty in short time scales? In *Proceedings of the 2005 ACM SIGMETRICS international conference on measurement and modeling of computer systems* (pp. 241–252).
Jiang, Y., & Liu, Y. (2008). *Stochastic network calculus*: Vol. 1. Springer.
Karsai, M., Jo, H.-H., & Kaski, K. (2018). *Bursty human dynamics*. Springer.
Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery, 7*(4), 373–397.
Konstantopoulos, T., & Anantharam, V. (1995). Optimal flow control schemes that regulate the burstiness of traffic. *IEEE/ACM Transactions on Networking, 3*(4), 423–432.
Lappas, T., Arai, B., Platakis, M., Kotsakos, D., & Gunopulos, D. (2009). On burstiness-aware search for document sequences. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 477–486).
Le Boudec, J.-Y., & Thiran, P. (2001). *Network calculus: A theory of deterministic queuing systems for the internet*. Springer.
Leland, W. E., Taqqu, M. S., Willinger, W., & Wilson, D. V. (1994). On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking, 2*(1), 1–15.
Liu, Y., Halev, A., & Liu, X. (2021). Policy learning with constraints in model-free reinforcement learning: A survey. In *Ijcai* (pp. 4508–4515).
Naor, P. (1969). The regulation of queue size by levying tolls. *Econometrica: Journal of the Econometric Society*, 15–24.
Niu, Y., Jin, P., Guo, J., Xiao, Y., Shi, R., Liu, F., … Wang, Y. (2021). Postman: Rapidly mitigating bursty traffic via on-demand offloading of packet processing. *IEEE Transactions on Parallel and Distributed Systems, 33*(2), 374–387.
Paxson, V., & Floyd, S. (1995). Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking, 3*(3), 226–244.
Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley.
Ross, K. W., & Varadarajan, R. (1989). Markov decision processes with sample path constraints: The communicating case. *Operations Research, 37*(5), 780–790.
Sriram, K., & Whitt, W. (1986). Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications, 4*(6), 833–846.
Stidham, S. (1985). Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control, 30*(8), 705–713.
Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.
Tanenbaum, A. S. (2003). *Computer networks* (4th ed.). Prentice-Hall.
Tessler, C., Mankowitz, D. J., & Mannor, S. (2018). Reward constrained policy optimization. In *International conference on learning representations*.
Varagapriya, V., Singh, V. V., & Lisser, A. (2022). Constrained Markov decision processes with uncertain costs. *Operations Research Letters, 50*(2), 218–223.
Willinger, W., Taqqu, M. S., Sherman, R., & Wilson, D. V. (1997). Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking, 5*(1), 71–86.
Yaron, O., & Sidi, M. (1993). Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Transactions on Networking, 1*(3), 372–385.
Yin, J., Lu, X., Zhao, X., Chen, H., & Liu, X. (2014). Burse: A bursty and self-similar workload generator for cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*(3), 668–680.
Yu, H. (2022). On linear programming for constrained and unconstrained average–cost Markov decision processes with countable action spaces and strictly unbounded costs. *Mathematics of Operations Research, 47*(2), 1474–1499.