Burstiness-Constrained Markov Decision Processes

Michal Golan

Burstiness-Constrained Markov Decision Processes

Research Thesis

Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Electrical Engineering

Michal Golan

Submitted to the Senate
of the Technion — Israel Institute of Technology
Kislev 5777 Haifa December 2016

This research was carried out under the supervision of Prof. Nahum Shimkin, in the Department of Electrical Engineering.

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS GRATEFULLY ACKNOWLEDGED.

Contents

List of Figures, Tables and Algorithms

Figures

Tables

Algorithms

Abstract

Burstiness of a dynamic process is the behavior of anomalous changes in the process's volume or frequency of occurrence. Bursty processes are encountered, for example, in communication networks, storage systems, and cloud-computing systems. Presence of bursty signals can cause severe network congestion and load unbalancing in clouds, which degrades the overall system performance. We consider the problem of Burstiness-Constrained MDPs (BCPs): Markov Decision Processes with a sample-path constraint on the burstiness of an associated cost.

The burstiness model we use requires any sum of consecutive cost elements to be no greater than a quantity that is proportional to the number of summed elements, plus a constant. Due to the burstiness constraint's form, feasible BCP policies are generally history-dependent. Thus, determining the problem's feasibility is prohibitive in its original form. Additionally, we cannot solve it with standard MDP algorithms such as backwards induction, value iteration, policy iteration and linear programming formulation.

We propose algorithms to determine the feasibility of BCPs, and to find their optimal feasible policies, in both finite and infinite time horizon settings. This is done by reformulating the burstiness constraint and posing it as a state constraint. We then characterize the conditions for feasibility of every state, action and policy, and develop procedures to determine whether these conditions are met. Next, we augment the BCP's state variable with an additional term which encompasses the past information regarding satisfaction of the burstiness constraint. The feasible policies are then Markov in the augmented state-space. Restricting the reformulated BCP to only feasible states and actions yields an unconstrained MDP with an augmented state-space, which can be treated with the aforementioned tools.

List of Abbreviations

MDP	Markov Decision Process
CMDP	Constrained MDP
BCP	Burstiness-Constrained MDP
BCF	BCP feasibility problem
BCP	BCP optimization problem
RBCP	Reformaulted BCP
RBCF	RBCP feasibility problem
RBCP	RBCP optimization problem
LP	Linear Programming
BC	Burstiness Constraint
ea	expected average (objective function)

List of Notations

Chapter 2	
$\|\cdot\|_\infty$	Maximum norm
a_t	Action preformed at time t
$\mathcal{A}$	Action space
$A_t(s)$	Action set at time t and state s
$A(s)$	Action set at state s in stationary MDP
$\mathbb{E}^{\pi,s}[\cdot]$	Expectation operator induced by policy π and initial state s
h_t	History sequence at time t
H_t	History-space at time t
$J_N^\pi(\cdot)$	Total expected reward function of policy π
$J_\gamma^\pi(\cdot)$	Expected discounted reward function of policy π
$J_{\text{ea}}^\pi(\cdot)$	Expected average reward function of policy π
$J_o^*(\cdot)$	Optimal value for objective type $o \in \{N, \gamma, \text{ea}\}$
J_γ^π	Expected discounted reward vector of policy π
J_γ^*	Optimal expected discounted reward vector
N	Time horizon length
$p_t(\cdot\|\cdot,\cdot)$	Transition probability function at time t
$p(\cdot\|\cdot,\cdot)$	Transition probability function in stationary MDP
$\mathcal{P}$	State-transition probability kernel
$\mathscr{P}(X)$	Set of probability distributions over a set X
$\mathbb{P}^{\pi,s}(\cdot)$	Probability measure induced by policy π and initial state s
P^π	State-transition probability matrix under policy π
$r_t(\cdot,\cdot)$	Reward function at time t
$r(\cdot,\cdot)$	Reward function in stationary MDP
r^π	Immediate reward vector under policy π
$\mathcal{R}$	Reward structure
s_t	State at time t
$\mathcal{S}$	State space
S_t	State set at time t
S	State set in stationary MDP

t	Time index
$\mathcal{T}$	Time horizon
V	Space of real-valued functions on S
x^+	Positive part of x, $\max\{x, 0\}$
$x(\cdot,\cdot)$	State-action frequency function
$x^*(\cdot,\cdot)$	Optimal state-action frequency function
$\beta(\cdot)$	Initial state probability distribution
γ	Discount factor
π	A control policy
π_o^*	Optimal policy for objective type $o \in \{N, \gamma, \text{ea}\}$
π_ϵ^*	ϵ-optimal policy for expected discounted reward objective
π_x	Stationary randomized policy induced by state-action frequency function x
Π^{HR}	Space of history-dependent, randomized control policies
Π^{MR}	Space of Markov randomized control policies
Π^{MD}	Space of Markov deterministic control policies
Π^{SR}	Space of stationary randomized control policies
Π^{SD}	Space ofstationary deterministic control policies
Ω	Sample space
Chapter 3	
$B_{\sigma,\rho}$	Event of obeying the (σ, ρ) burstiness constraint
$d_t(\cdot,\cdot)$	Cost function at time t
$d(\cdot,\cdot)$	Cost function in stationary BCP
$\mathcal{D}$	Cost structure
K	Dimension of cost fuction
$J_o^{F*}(\cdot)$	Optimal feasible value for objective type o
S_0^F	Set of feasible initial states
π_o^{F*}	Optimal feasible policy for objective type o
ρ	Burstiness constraint coefficients
σ	Burstiness constraint coefficients
$\Phi(\cdot)$	Feasibility indicator function
Chapter 4	
$B(y)$	Extended burstiness constraint on BCP
$\tilde{B}$	Burstiness constraint on RBCP
$\tilde{\mathbb{E}}^{\tilde{\pi},\tilde{s}}[\cdot]$	Expectation operator induced on RBCP by policy $\tilde{\pi}$ and initial state $\tilde{s}$
$\tilde{h}_t$	RBCP history sequence at time t
$\tilde{H}_t$	RBCP history-space at time t
$\tilde{J}_N^{\tilde{\pi}}(\cdot)$	Total expected reward function of RBCP policy $\tilde{\pi}$
$\tilde{J}_\gamma^{\tilde{\pi}}(\cdot)$	Expected discounted reward function of RBCP policy $\tilde{\pi}$

$\tilde{J}_o^{F^*}(\cdot)$	Optimal feasible RBCP value for objective type o
$\tilde{p}_t(\cdot\|\cdot,\cdot)$	RBCP state-transition probability function at time t
$\tilde{p}(\cdot\|\cdot,\cdot)$	RBCP state-transition probability function in the stationary case
$\tilde{\mathcal{P}}$	RBCP state-transition probability kernel
$\tilde{\mathbb{P}}^{\tilde{\pi},\tilde{s}}(\cdot)$	Probability measure induced on RBCP by policy $\tilde{\pi}$ and initial state $\tilde{s}$
$\tilde{s}_t = (s_t, y_t)$	RBCP state at time t
$\tilde{\mathcal{S}}$	RBCP state space
$\tilde{S}_t$	RBCP state set at time t
$\tilde{S}$	RBCP state set in the stationary case
$\tilde{S}_t^F$	Set of feasible RBCP states at time t
$\tilde{S}^F$	Set of feasible RBCP states in stationary BCP
y_t	Maximum deficit accumulated by time t
$Y_t(s)$	Set of possible y_t values at time t and state s
$Y(s)$	Set of possible y_t values at state s in stationary BCP
$y^*(\cdot)$	RBCP feasibility threshold function
$\tilde{\pi}$	RBCP control policy
$\tilde{\pi}_o^{F^*}$	Optimal feasible RBCP policy for objective type o
$\tilde{\Pi}^{HR}$	Space of history-dependent, randomized RBCP control policies
$\tilde{\Pi}^{MD}$	Space of Markov deterministic RBCP control policies
$\tilde{\Phi}(\cdot,\cdot)$	RBCP feasibility indicator function
$\tilde{\Omega}$	RBCP sample space
Chapter 5	
$\emptyset$	The empty set
$A_t^F(s,y)$	Set of feasible RBCP actions at time t and state (s,y)
$A^F(\cdot,\cdot)$	Set of feasible RBCP actions at state (s,y) in stationary BCP
$\tilde{B}_t(\cdot)$	Event of RBCP adherence to burstiness constraint from time t onward
$\mathcal{F}\{\cdot\}$	One-step feasibility operator
$y_t^*(\cdot)$	RBCP feasibility threshold function for time t
$\mathcal{Y}$	Set of eigen-functions of $\mathcal{F}\{\cdot\}$
$y(s,a), f_y(s,a)$	Feasibility operator auxiliary functions
$y^*(\cdot,\cdot), f_{y^*}(\cdot,\cdot)$	Auxiliary functions corresponding to $y^*(\cdot)$ in stationary BCP
$\tilde{\Pi}^{HR,F}$	Space of feasible history-dependent, randomized RBCP control policies
$\tilde{\Pi}^{MD,F}$	Space of feasible Markov deterministic RBCP control policies
$\tilde{\Pi}^{SR,F}$	Space of feasible stationary randomized RBCP control policies
$\tilde{\Pi}^{SD,F}$	Space of feasible stationary deterministic RBCP control policies
ϕ	Non-numerical value for computational use

Chapter 1

Introduction

Burstiness of a dynamic process is the behavior of anomalous changes in the process's volume or frequency of occurrence. A common example of a bursty process is the traffic in communication networks, where large files (e.g., audio, video and interactive data) may be sent at irregular times. Bursty workloads are also experienced in production grids, storage systems, file systems and multi-tier computing architectures. Burtsiness of a process may be caused by internet flash-crowds and traffic surges, or built up due to the topology of the underlying network. Presence of bursty signals can cause severe network congestion and load unbalancing in clouds, which degrades the overall system performance [KA95], [MOSM90], [TZL11].

In this work, we use the (σ, ρ) formulation of burstiness, which was presented by Cruz in [Cru91]: given coefficients $\sigma, \rho \geq 0$, a sequence $(d_t)_{t \in \mathcal{T}}$ over some time horizon $\mathcal{T}$ is said to be (σ, ρ)-burstiness constrained if it satisfies

$$\sum_{t=t_1}^{t_2} d_t \leq \rho(t_2 - t_1 + 1) + \sigma, \quad \forall t_1, t_2 \in \mathcal{T} : \ t_1 \leq t_2. \tag{1.1}$$

This constraint requires any sum of consecutive elements of d_t to be no greater than ρ times the number of summed elements, plus σ. By this formulation, ρ defines an upper bound to the long term average of the sequence, and σ supports occasional deviations from this bound.

This work is focused on discrete-time control of a stochastic process under a sample-path constraint on its burstiness. We introduce the Burstiness-Constrained MDP (BCP) setting, which consists of an MDP with an additional function of the state and action sequence, $d_t(s_t, a_t)$ (where s_t and a_t are the system's state and the action performed at time t, respectively). A control policy is feasible for the BCP if, given this policy, the sequence $(d_t(s_t, a_t))_{t \in \mathcal{T}}$ obeys the (σ, ρ)-burstiness constraint (1.1) *with probability 1.* In this setting, we would like to optimize some objective function of the BCP over all

such feasible policies. We look at both finite horizon and infinite horizon BCPs.

As an example, consider a queue sending jobs to a server, which we control by determining the number of jobs sent by the queue at every time point. The system's state is then the number of jobs in the queue at every time point, $s_t \in S$, which adheres to a certain dynamics. The performed action, a_t, is the number of jobs sent to the server, so that the set of available actions for state $s \in S$ is $A(s) = \{0, ..., s\}$. The system is illustrated in Figure 1.1.

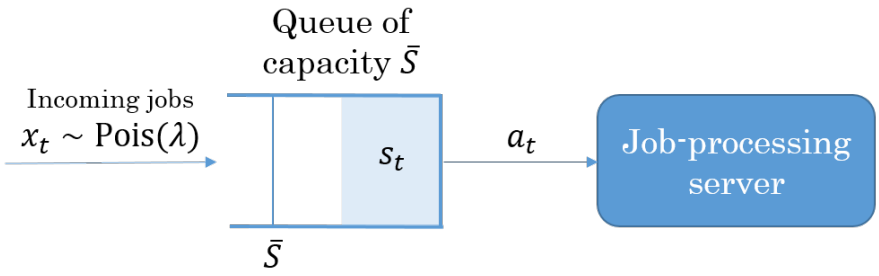

Figure 1.1: Job queue for a server

At every time point, a cost of C_1 is incurred per every job held in queue and of C_2 per every job sent to the server, i.e., the cost at time t is $d_t = d(s_t, a_t) = C_1 s_t + C_2 a_t$. We would like to maximize the expected discounted number of jobs sent to the server, whilst maintaining (σ, ρ) burstiness constraints on the cost sequence, $(d_t)_{t \in \mathcal{T}}$. The reward function is thus $r(s_t, a_t) = a_t$, and the resulting optimization problem is

$$\begin{aligned} \max_{\pi \in \Pi^{HR}} \quad & \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \\ \text{s.t.} \quad & \sum_{t=t_1}^{t_2} d(s_t, a_t) \leq \rho(t_2 - t_1) + \sigma, \quad \forall t_1, t_2 \in \{0, 1, ...\} : \; t_1 \leq t_2, \quad \text{w.p. } 1, \end{aligned}$$

where Π^{HR} is the set of history-dependent randomized policies on this MDP, and $\mathbb{E}^{\pi,s}[\cdot]$ is the expectation operator induced by policy π and initial state s.

Burstiness constraints may be imposed on other measures such as network load, transmission delays, or probability of loss of data packets in a network, or in other settings, e.g., inventory management and supply chains.

In order to solve this type of problem, we must first find an efficient way to identify the feasible policies, if any exist. Determining whether a policy is feasible, and solving the BCP optimization problem, pose several challenges. First, note that the inequalities

in the burstiness constraint (1.1) aren't posed on each d_t element independently. Rather, they couple between different time points: each value of d_t affects several inequalities. Thus, when $d_t = d_t(s_t, a_t)$, deciding whether each action a_t allows satisfaction of the burstiness constraint, depends on past and future values of the sequence $(d_t(s_t, a_t))_{t\in\mathcal{T}}$, and consequently on the MDP's entire sample path. Consequently, the feasible policies are generally history-dependent. Moreover, in order for a policy to obey the burstiness constraint with probability 1, *any possible* sample-path under this policy must obey the constraint. In addition, in the finite horizon case, the number of inequalities in the burstiness constraint is of the order of $N^2/2$ (N being the length of the time-horizon), and infinite in the inifinite horizon case. Thus, an exhaustive inspection for each policy's feasibility is prohibitive. Lastly, since the BCP is not in the form of a conventional MDP, we cannot apply to it the standard MDP tools such as Backwards Induction, Value Iteration, Policy Iteration and Linear Programming, and we must find efficient algorithms to solve it.

However, we find that a quantity which we call the *maximum possible backlog* at each time point, and denote by y_t, encapsulates the entire information required in order to determine which actions would preserve compliance with the burstiness constraint. By augmenting the BCP's state-space with this term, the burstiness constraint is transformed into the simpler form of a *state constraint*. We characterize the backlog's behavior, and come up with simple algorithms to determine the feasibility of the BCP in the finite and the inifinite horizon cases. Additionally, in the augmented state-space, the feasible policies are no longer history-dependent but are rather Markov (and stationary, in the infinite horizon case). This allows us to use the standard MDP tools in order to solve the BCP optimization problem, with certain extensions.

This thesis is organized as follows. We start with a scientific background on the MDP setting and its key results, constrained MDPs, and the burstiness constraint, in Chapter 2. In Chapter 3 we give an exact formulation of the Burstiness-Constrained MDP (BCP) feasibility and optimization problems, explain the complexities of solving these problems, and demostrate them. A reformulation of the BCP as a state-constrained MDP is given in Chapter 4. Following this reformulation, algorithms for solving the BCP feasibility and optimization problems are presented in Chapters 5 and 6 for the finite and infinite horiozon cases, respectively. In Chapter 7 we demonstrate the application of our algorithms on BCP examples. Conclusions and possible directions for further research are given in Chapter 8.

Chapter 2

Scientific Background and Literature Survey

In this chapter we give an overview of the mathematical framework for burstiness-constrained MDPs: MDPs, constrained MDPs, and the burstiness constraint. We introduce several mathematical notations that are used in this work in Section **??**. Section 2.1 describes the structure of MDPs, and several of their key results and solution algorithms. Sections 2.2 and 2.3 present constrained MDPs and the burstiness constraint, respectively, with a focus on key results which are relevant to our discussion on burstiness-constrained MDPs.

In the work presented herein, we use the following mathematical notations:

- $\mathscr{P}(X)$ denotes the unit simplex over a discrete set X, that is, the set of probability distributions over X, $\mathscr{P}(X) \triangleq \left\{ f : X \to [0,1] \text{ s.t. } \sum_{x \in X} f(x) = 1 \right\}$.
- x^+ denotes the positive part of x, i.e., $x^+ = \max\{x, 0\}$ where $x \in \mathbb{R}$.
- Mathematical operators such as *max* and inequalities are extended to operate on vectors element-wise. In this context, recall that when x and y are multidimensional vectors, $x \nleqslant y$ does not imply $x > y$.

2.1 Markov Decision Processes

2.1.1 MDP Model and Notations

MDPs provide a simple mathematical framework for modelling dynamic, stochastic environments which allow some degree of control. For a thorough review and definition of

MDPs, see Puterman [Put14]. In this work we focus on discrete-time systems with finite state and action spaces. Such MDPs are defined by a 5-tuple of the form $\langle \mathcal{T}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, where:

- $\mathcal{T}$ is the *time horizon*, a set of *time points* which may be either finite, $\mathcal{T} \triangleq \{0, 1, ..., N\}$ where $N \in \mathbb{N}$, or infinite, $\mathcal{T} \triangleq \{0, 1, ...\}$.
- $\mathcal{S} \triangleq (S_t)_{t \in \mathcal{T}}$ is the *state-space*, a sequence of finite sets of available *states* for every time point. The state at time t is denoted by s_t.
- $\mathcal{A} \triangleq (A_t(s))_{t \in \mathcal{T},\ s \in S_t}$ is the *action-space*, a sequence of finite sets of available *actions* for every decision epoch and state. At each time point (except for the last one, in finite horizon MDPs), an action is performed, denoted by $a_t \in A_t(s_t)$.
- $\mathcal{P} \triangleq (p_t(s'|s,a))_{t \in \mathcal{T}}$ is the *state transition probability kernel*, a sequence of state-transition probability functions for every decision epoch, where $s \in S_t,\ a \in A_t(s),\ s' \in S_{t+1}$, and $p_t(\cdot|s,a) \in \mathscr{P}(S_{t+1})$. The probability measure of the resulting sample space is denoted by $\mathbb{P}(\cdot)$, such that $\mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a) = p_t(s'|s,a)$.
- $\mathcal{R} \triangleq (r_t(s,a))_{t \in \mathcal{T}}$ is the *reward structure*, a sequence of reward functions for every time point, where $s \in S_t,\ a \in A_t(s)$ and $r_t(s,a) \in \mathbb{R}$. At each time point a reward $r_t(s_t, a_t)$ is accrued. The MDP's goal is to maximize some objective function of the accrued rewards. In finite horizon MDPs, the reward in the last time point depends only on the current state: $r_N(s),\ s \in S_N$.

In infinite horizon MDPs, we assume stationarity of the model, i.e., the state-sets, action sets, transition probabilities and immediate rewards are time-invariant, and accordingly the time index is omitted from $S_t,\ A_t,\ p_t$ and r_t, yielding the following notations:

$$S_t \equiv S, \quad A_t(\cdot) \equiv A(\cdot), \quad p_t(\cdot|\cdot,\cdot) \equiv p(\cdot|\cdot,\cdot), \quad r_t(\cdot,\cdot) \equiv r(\cdot,\cdot).$$

For every $t \in \mathcal{T}$, any possible sequence of past states and actions $(s_0, a_0, s_1, a_1, ..., s_t)$ is called a *t-history*. Denote by H_t the set of all such t-histories:

$$H_t \triangleq \{(s_0, a_0, ..., s_t) :\ s_{t'} \in S_{t'},\ a_{t'} \in A_{t'}(s_{t'}),\ \forall t' \leq t\}, \quad t \in \mathcal{T}.$$

The sample space of the stochastic process generated by the MDP is therefore $\Omega \triangleq H_N$ in finite horizon MDPs, and $\Omega \triangleq H_\infty$ in infinite horizon MDPs, where

$$H_\infty \triangleq \{(s_0, a_0, ...) :\ s_t \in S_t,\ a_t \in A_t(s_t),\ \forall t \in \{0, 1, ...\}\}.$$

2.1.2 MDP Policies

A *control policy* is used to determine the actions performed at every decision epoch, $(a_t)_{t\in\mathcal{T}}$. We define the following classes of policies:

- Π^{HR}, the set of history-dependent randomized policies, such that any $\pi \in \Pi^{HR}$ is a sequence of decision rules for every decision epoch, $\pi = (\pi_t(a|h))_{t\in\mathcal{T}}$, where $h = (..., s) \in H_t,\ a \in A_t(s),\ \pi_t(\cdot|h) \in \mathscr{P}(A_t(s))$, and $a_t \sim \pi_t(\cdot|s_0, a_0, ..., s_t)$.
- Π^{MR}, the set of Markov randomized policies, such that any $\pi \in \Pi^{MR}$ is a sequence of decision rules for every decision epoch, $\pi = (\pi_t(a|s))_{t\in\mathcal{T}}$, where $s \in S_t,\ a \in A_t(s),\ \pi_t(\cdot|s) \in \mathscr{P}(A_t(s))$, and $a_t \sim \pi_t(\cdot|s_t)$.
- Π^{MD}, the set of Markov deterministic policies, such that any $\pi \in \Pi^{MD}$ is a sequence of decision rules for every decision epoch, $\pi = (\pi_t(s))_{t\in\mathcal{T}}$, where $s \in S_t,\ \pi_t(s) \in A_t(s)$, and $a_t = \pi_t(s_t)$.

Observe that $\Pi^{MD} \subset \Pi^{MR} \subset \Pi^{HR}$.

In stationary infinite horizon MDPs, the following classes of policies are of special interest:

- Π^{SR}, the set of stationary randomized policies, such that any $\pi \in \Pi^{SR}$ is a function of the form $\pi(a|s)$ where $s \in S,\ a \in A(s),\ \pi(\cdot|s) \in \mathscr{P}(A(s))$, and $a_t \sim \pi(\cdot|s_t)$.
- Π^{SD}, the set of stationary deterministic policies, such that any $\pi \in \Pi^{SD}$ is a function of the form $\pi(s) \in A(s)$, and $a_t = \pi(s_t)$.

Observe that $\Pi^{SD} \subset \Pi^{SR} \subset \Pi^{MR}$ and $\Pi^{SD} \subset \Pi^{MD}$.

$\mathbb{P}^{\pi,s}(\cdot)$ and $\mathbb{E}^{\pi,s}[\cdot]$ denote the probability measure and expectation operator, respectively, that are indcued when the MDP starts with initial state $s \in S_0$ and uses the control policy π.

2.1.3 MDP Optimization Problem

In this setting, one would like to optimize some objective function of the acquired rewards, over all history-dependent randomized control policies. Given a control policy, π, and initial state, $s \in S_0$, we may focus on the following objectives:

- The *expected total reward*, for a finite horizon MDP,

$$J_N^{\pi}(s) \triangleq \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{N-1} r_t(s_t, a_t) + r_N(s_N)\right].$$

- The *expected discounted reward*, for an infinite horizon MDP with $\gamma \in (0,1)$,

$$J_{\gamma}^{\pi}(s) \triangleq \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right].$$

- The *expected long-term average reward*, for an infinite horizon MDP,

$$J_{\mathrm{ea}}^{\pi}(s) \triangleq \liminf_{N\to\infty} \left\{ \tfrac{1}{N+1}\mathbb{E}^{\pi,s}\left[\sum_{t=0}^{N} r(s_t, a_t)\right]\right\}.$$

We refer to $J_o^{\pi}(\cdot)$ as the *value function* of the control policy π, where $o \in \{N, \gamma, \mathrm{ea}\}$ denotes the objective's type.

The MDP's optimization problem is formulated as follows.

MDP: Given an MDP and objective $o \in \{N, \gamma, \mathrm{ea}\}$, find $\pi_o^* \in \Pi^{HR}$ such that

$$J_o^{\pi_o^*}(s) = \max_{\pi\in\Pi^{HR}} \{J_o^{\pi}(s)\},\ \forall s \in S_0. \tag{2.1}$$

The optimal value function is denoted by $J_o^*(s) \triangleq J_o^{\pi_o^*}(s)$, and is also called the problem's value function.

2.1.4 Properties of Optimal Policies

An important property of MDPs is that the optimization over Π^{HR} may be reduced to subclasses of simpler policies. This is due to the fact that the state dynamics is Markov, and the objective functions in question are composed of sums of terms of the form $\mathbb{E}[r_t(s_t, a_t)]$. Consequently, any two control policies that induce the same marginal probability distribution over the state-action pairs (s_t, a_t) reach the same value of the objective function. Since any marginal probability distribution on (s_t, a_t) can be induced by a randomized Markov policy, the optimization in equation (2.1) may be limited to Π^{MR} without changing the optimal value. When the state and action spaces are finite, the optimization in (2.1) may be further reduced to Π^{MD}.

In stationary infinite horizon MDPs, we can reduce the optimization to Π^{SR}, and when the state and action spaces are finite, there exists an optimal policy in Π^{SD} (i.e., stationary and deterministic).

2.1.5 Finite Horizon MDPs with a Total Expected Reward Objective

2.1.5.1 Bellman's Optimality Equations

For a finite horizon MDP with a total expected reward objective, Bellman's optimality equations state that the respective value function is $J_N^*(s) = v_0^*(s),\ \forall s \in S_0$, and the optimal policy is $\pi_N^* = \left(\pi_t^*(s)\right)_{t\in\mathcal{T}, s\in S_t} \in \Pi^{MD}$, where

$$
\begin{aligned}
v_t^*(s) &\triangleq \max_{a\in A_t(s)} Q_t(s,a) \ \text{ and} \\
\pi_t^*(s) &\in \operatorname*{argmax}_{a\in A_t(s)} Q_t(s,a), \quad \forall t \in \{0,...,N-1\},\ s \in S_t; \\
Q_t(s,a) &\triangleq r_t(s,a) + \sum_{s'\in S_{t+1}} p_t(s'|s,a) v_{t+1}^*(s'), \quad \forall t \in \{0,...,N-1\},\ s\in S_t,\ a \in A_t(s); \\
v_N^*(s) &\triangleq r_N(s), \quad s \in S_N.
\end{aligned}
$$

2.1.5.2 Backward Induction Algorithm

Following the equations in Section 2.1.5.1, we obtain a dynamic programming algorithm which efficiently solves finite horizon MDPs with an expected total reward objective:

1. For $s \in S_N$, set $v_N^*(s) = r_N(s)$.

2. For $t = N-1, ..., 0$,
   For $s \in S_t$,
   For $a \in A_t(s)$, set $Q_t(s,a) = r_t(s,a) + \sum_{s'\in S_{t+1}} p_t(s'|s,a) \cdot v_{t+1}^*(s')$;
   Set $\pi_t^*(s) \in \operatorname{argmax}_{a\in A_t(s)} Q_t(s,a)$;
   Set $v_t^*(s) = \max_{a\in A_t(s)} Q_t(s,a)$.

3. The policy $\pi_N^* = (\pi_t^*)_{t=0}^{N-1} \in \Pi^{MD}$ optimizes **MDP**, and the optimal value function is $J_N^*(s) = v_0^*(s),\ \forall s \in S_0$.

2.1.6 Infinite Horizon Notations

For stationary infinite horizon MDPs, we further introduce the following notations:

Denote by V the space of real-valued functions on S, $V \triangleq \{v : S \to \mathbb{R}\}$. Since S is finite, V is equivalent to the space of $|S|$-vectors on $\mathbb{R}$, i.e., $\mathbb{R}^{|S|}$. Hence, we can interchangeably refer to any $v \in V$ as a function on S and as a vector in $\mathbb{R}^{|S|}$. In particular, observe that $J_o^\pi \in V$.

$P^\pi \in [0,1]^{|S|\times|S|}$ and $r^\pi \in \mathbb{R}^{|S|}$ are, respectively, the state-transition matrix and immediate reward vector induced when using a policy $\pi \in \Pi^{SD}$, such that $P^\pi_{s,s'} \triangleq p(s'|s,\pi(s))$ and $r^\pi_s \triangleq r(s,\pi(s))$.

2.1.7 Stationary Infinite Horizon MDPs with an Expected Discounted Reward Objective

2.1.7.1 Bellman's Equations

Given a stationary infinite horizon MDP with an expected discounted reward objective, for any control policy $\pi \in \Pi^{SD}$, its value function, $J^\pi_\gamma(\cdot)$, satisfies the following Bellman-type equation:

$$J^\pi_\gamma = r^\pi + \gamma P^\pi J^\pi_\gamma. \tag{2.2}$$

The optimal value function, $J^*_\gamma(\cdot)$, satisfies Bellman's optimality equation:

$$J^*_\gamma(s) = \max_{a\in A(s)} \left\{ r(s,a) + \gamma \sum_{s'\in S} p(s'|s,a) J^*_\gamma(s') \right\},\ \forall s \in S. \tag{2.3}$$

2.1.7.2 Value Iteration Algorithm

Given some $\epsilon > 0$, a control policy π and its corresponding value function, $J^\pi_o(s),\ s \in S$, are called ϵ-*optimal*, if $\|J^\pi_o - J^*_o\|_\infty \leq \epsilon$, where $\|\cdot\|_\infty$ is the *max* norm operator.

Based on Bellman's optimality equation, the value iteration algorithm computes an ϵ-optimal policy and value function for stationary infinite horizon MDPs with the expected discounted reward objective. That is, it computes a policy $\pi^*_\epsilon \in \Pi^{SD}$ whose value fucntion, $J^{\pi^*_\epsilon}_\gamma(s),\ s \in S$, satisfies $\|J^{\pi^*_\epsilon}_\gamma - J^*_\gamma\|_\infty \leq \epsilon$ for any $\epsilon > 0$.

The algorithm is based on successive applications of a contraction operator on some initial guess of the problem's value, which converges to the problem's true value. Since the problem's value-space is non-discrete ($\mathbb{R}^{|S|}$), the algorithm can only guarantee an ϵ-optimal result.

1. Select some $v^0(s),\ \forall s \in S$, and specify $\epsilon > 0$.

2. For $n = 1, 2, ...,$

   For $s \in S$,

   For $a \in A(s)$, set $Q^n(s,a) = r(s,a) + \gamma \sum_{s'\in S} p(s'|s,a) \cdot v^{n-1}(s')$;

Set $v^n(s) = \max_{a \in A(s)} Q^n(s,a)$.

If $\max_{s \in S}\{v^n(s) - v^{n-1}(s)\} < \epsilon(1-\gamma)/2\gamma$, continue to step 3.

3. The policy $\pi^*_\epsilon \in \Pi^{SD}$, where $\pi^*_\epsilon(s) \in \text{argmax}_{a \in A(s)}\, Q^n(s,a),\ \forall s \in S$, is ϵ-optimal for **MDP**, and $J^{\pi^*_\epsilon}_\gamma(s) = v^n(s),\ \forall s \in S$ is an ϵ-optimal approximation of its value function, i.e., $\|J^{\pi^*_\epsilon}_\gamma - J^*_\gamma\|_\infty \leq \epsilon$.

2.1.7.3 Policy Iteration Algorithm

Rather than improve our estimate of the problem's *value*, as done in the value iteration algorithm, the policy iteration algorithm sequentially improves our estimate of the optimal *policy*, using Equation (2.2).

1. Select some $\pi^0 \in \Pi^{SD}$.

2. For $n = 0, 1, ...,$

   Policy evaluation: Solve $(I - \gamma P^{\pi^n})v^n = r^{\pi^n}$ for $v^n \in \mathbb{R}^{|S|}$.

   Policy improvement: For $s \in S$, choose

   $\pi^{n+1}(s) \in \text{argmax}_{a \in A(s)}\{r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v^n_{s'}\}$,

   setting $\pi^{n+1}(s) = \pi^n(s)$ if possible.

   If $\pi^{n+1} = \pi^n$, continue to step 3.

3. The policy $\pi^*_\gamma = \pi^n \in \Pi^{SD}$ optimizes **MDP**, and the optimal value function is $J^*_\gamma(s) = v^n_s,\ \forall s \in S$.

Since the policy space is finite, the policy improvement step eventually yields the actual optimal policy, rather than a nearly-optimal policy. However, when the state and action spaces are very large, solving the linear equation in step 2 may be prohibitive.

2.1.7.4 Linear Programming Algorithm

A third method of solving the stationary infinite horizon MDP with an expected discounted reward is via a linear program.

We first observe that for any policy $\pi \in \Pi^{HR}$, the MDP's objective function can be expressed as a linear combination of π's *state-action frequencies*:

$$J^\pi_\gamma(s) = \sum_{s' \in S} \sum_{a' \in A(s')} r(s',a') \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi,s}(s_t = s', a_t = a'),\ \forall s \in S.$$

In addition, the optimal policy can be found by

$$\pi^*_\gamma \in \underset{\pi\in\Pi^{HR}}{\operatorname{argmax}} \sum_{s\in S} \alpha(s) J^\pi_\gamma(s),$$

for any arbitrary $\alpha(s) > 0,\ \forall s \in S$ (note that $\alpha(\cdot)$ can be viewed as the initial state's probability distribution).

Thus, we can write the MDP optimization problem as follows:

$$\pi^*_\gamma \in \underset{\pi\in\Pi^{HR}}{\operatorname{argmax}} \sum_{s'\in S} \sum_{a'\in A(s')} r(s',a') x^\pi(s',a'),$$

where

$$x^\pi(s',a') = \sum_{s\in S} \alpha(s) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi,s}(s_t = s', a_t = a'),\ \forall \pi \in \Pi^{HR},\ s' \in S,\ a' \in A(s').$$

In order to formulate the problem as a linear program, we recall that in the stationary case, we can narrow the search for an optimal policy to stationary randomized policies:

$$\pi^*_\gamma \in \underset{\pi\in\Pi^{SR}}{\operatorname{argmax}} \sum_{s'\in S} \sum_{a'\in A(s')} r(s',a') x^\pi(s',a').$$

This property is useful since the set of functions $x^\pi(s,a)$ that correspond to stationary randomized policies, $\pi \in \Pi^{SR}$, can be expressed in linear form (as opposed to Π^{HR}), and is also convex (as opposed to Π^{SD}, for example):

For any $\pi \in \Pi^{SR}$, $x^\pi(s,a)$ satisfies the following equations:

$$\begin{aligned} &\sum_{a'\in A(s')} x(s',a') - \gamma \sum_{s\in S} \sum_{a\in A(s)} p(s'|s,a) \cdot x(s,a) = \alpha(s'), \quad \forall s' \in S. \\ &x(s,a) \geq 0,\ \forall s \in S,\ a \in A(s). \end{aligned} \tag{2.4}$$

In addition, any function $x(s,a)$ which satisfies Equations (2.4) corresponds to a stationary randomized policy $\pi_x \in \Pi^{SR}$:

$$\pi_x(a|s) = \frac{x(s,a)}{\sum_{a'\in A(s)} x(s,a')},\ \forall s \in S,\ a \in A(s),$$

such that $x(s,a) = x^{\pi_x}(s,a),\ \forall s \in S,\ a \in A(s)$.

Therefore, **MDP** can be solved using the following steps:

1. Solve the following linear program:

$$x^* = \underset{x(\cdot,\cdot)}{\operatorname{argmax}} \sum_{s\in S}\sum_{a\in A(s)} r(s,a)\cdot x(s,a),$$

subject to

$$\sum_{a'\in A(s')} x(s',a') - \gamma \sum_{s\in S}\sum_{a\in A(s)} p(s'|s,a)\cdot x(s,a) = \alpha(s'), \quad \forall s' \in S,$$

$$x(s,a) \geq 0, \quad \forall s \in S,\ a \in A(s). \tag{2.5}$$

2. The policy $\pi^*_\gamma = \pi_{x^*} \in \Pi^{SR}$, where

$$\pi_{x^*}(a|s) = \frac{x^*(s,a)}{\sum_{a'\in A(s)} x^*(s,a')},\ \forall s \in S,\ a \in A(s),$$

optimizes **MDP**, and its optimal value is $J^*_\gamma = J^{\pi_{x^*}}_\gamma$.

This is a useful approach to solving MDPs, as linear programs are a well-researched domain in optimization theory and have a variety of efficient solution algorithms.

2.2 Constrained MDPs

A useful variant of Markov Decision Processes are Constrained MDPs (CMDPs), where the optimization problem is accompanied by constraints on some functionals of the state and action sequence, for example

$$\mathbb{E}^{\pi,s}\left[\sum_{t=0}^{\infty} \gamma^t d(s_t,a_t)\right] \leq D, \tag{2.6}$$

where $d(\cdot,\cdot)$ is some cost function of the curent state and action, D is a constant, and $\gamma \in [0,1]$. Such situations arise, for example, when so dictated by the MDP's environment, or when the decision maker has several objectives and she prefers to optimize one objective while setting specific bounds on the others, rather than to optimize some combination of the various objectives.

In such problems, we search for a control policy which optimizes the objective function whilst obeying the CMDP's constraints. An example of a CMDP problem is a stationary infinite horizon system where we would like to maximize the expected

discounted reward, while constraining the expected discounted cost:

$$\max_{\pi \in \Pi^{HR}} \mathbb{E}^{\pi,\beta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \quad \text{s.t.} \quad \mathbb{E}^{\pi,\beta}\left[\sum_{t=0}^{\infty} \gamma^t d(s_t, a_t)\right] \leq D, \tag{2.7}$$

where $\beta(\cdot) \in \mathscr{P}(S)$ is the initial state's probability distribution.

See Chapter 1 of Altman [Alt99] for an overview of solution approaches for various CMDPs.

A common type of constraint is a bound on the *expectation* of some functional of the state and action sequence, such as in Equation (2.6) above. When the constrained functional takes the same form as the objective function (as in Problem (2.7)), we can use the fact that both can be written as a linear combination of the selected control policy's state-action frequencies. Altman [Alt99] discusses solution methods for such CMDPs, e.g. Problem (2.7) and the corresponding expected average reward/cost version:

$$\begin{aligned} \max_{\pi \in \Pi^{HR}} \quad & \liminf_{N \to \infty} \left\{ \tfrac{1}{N+1} \mathbb{E}^{\pi,\beta}\left[\sum_{t=0}^{N} r(s_t, a_t)\right] \right\} \\ \text{s.t.} \quad & \liminf_{N \to \infty} \left\{ \tfrac{1}{N+1} \mathbb{E}^{\pi,\beta}\left[\sum_{t=0}^{N} d(s_t, a_t)\right] \right\} \leq D. \end{aligned} \tag{2.8}$$

One solution approach to this type of problems is to write the problem in linear program form (as seen in Section 2.1.7.4), with added constraints to express the constrained functionals in terms of their state-action frequency vector. For Problem (2.7), for example, we get the next LP (Altman [Alt99], Section 1.7):

$$\begin{aligned} &\max_{x(\cdot,\cdot)} \sum_{s' \in S} \sum_{a' \in A(s')} r(s', a') \cdot x(s', a'), \\ &\text{subject to} \\ &\quad \sum_{a' \in A(s')} x(s', a') - \gamma \sum_{s \in S} \sum_{a \in A(s)} p(s'|s, a) \cdot x(s, a) = \beta(s'), \quad \forall s' \in S, \\ &\quad x(s, a) \geq 0, \quad \forall s \in S, \ a \in A(s), \\ &\quad \sum_{s \in S} \sum_{a \in A(s)} d(s, a) \cdot x(s, a) \leq D. \end{aligned} \tag{2.9}$$

The optimal policy in this case is stationary, but depends on the initial state distribution, $\beta(s)$. Hordijk and Kallenberg [HK84] consider Problem (2.8), and show that in the general multichain case, there does not exist a stationary policy which is optimal for any initial state distribution. However, when the CMDP is unichain, there exists such a policy.

In the case of K such constraints, Altman [Alt99] (Section 3.5) and Ross [Ros89] show that there exists an optimal stationary policy with no more than K randomizations, for the expected discounted reward and expected average reward objectives, respectively.

Another solution approach is to write the problem as a Lagrangian optimization problem. We then reverse the order of optimization on policies and Lagrange multipliers, thus transforming the problem into an unconstrained MDP optimization problem, with the Lagrange multipliers as parameters. Finally, we may obtain a linear program formulation to the problem, which is the dual of the LP from the previous approach (see Altman [Alt99] (Sections 3.3 and 3.4) for application of this approach on Problem (2.7)).

Another type of constraints demand that some functional of the state and action sequence be bounded with probability 1, e.g.,

$$\mathbb{P}^{\pi,s}\left(\liminf_{N\to\infty}\left\{\tfrac{1}{N+1}\sum_{t=0}^{N} d(s_t,a_t)\right\}\leq D\right)=1.$$

In such *sample-path* constraints, any possible realization of the state and action sequence, given a selected control policy, should obey the required bound.

Ross and Varadarajan [RV89] consider the problem of maximizing the expected average reward with a sample-path constraint on the average cost, in a communicating CMDP structure:

$$\begin{aligned}\max_{\pi\in\Pi^{HR}} \quad & \mathbb{E}^{\pi}\left[\liminf_{N\to\infty}\left\{\tfrac{1}{N+1}\sum_{t=0}^{N} r(s_t,a_t)\right\}\right] \\ \text{s.t.} \quad & \mathbb{P}^{\pi}\left(\liminf_{N\to\infty}\left\{\tfrac{1}{N+1}\sum_{t=0}^{N} d(s_t,a_t)\right\}\leq D\right)=1. \end{aligned} \tag{2.10}$$

As opposed to the expectation-type constraint case (see Problem (2.8) above), they show that under sample-path constraints there exist ϵ-optimal stationary policies for multichain CMDPs, and give an LP formulation for the problem's solution. They also show that in the unichain case, sample-path constraints behave like expectation-type constraints. In [RV91] they extend the algorithm to the general, non-communicating case, by decomposing the MDP into maximal recurrent classes of states and a set of transient states.

Another work on MDPs with a sample-path constraint was presented by Piunovskiy

[Piu06], where the following problem was considered:

$$\begin{aligned} \inf_{\pi \in \Pi^{HR}} \quad & \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty} \gamma_0^{t-1} r_t(s_{t-1}, a_t)\right] \\ \text{s.t.} \quad & \mathbb{P}^{\pi}\left(\sum_{t=1}^{\infty} \gamma_1^{t-1} d_t(s_{t-1}, a_t) \leq D\right) = 1. \end{aligned} \tag{2.11}$$

The solution approach towards this problem is similar to the one we use in the present work. The MDP's state-space is extended with an additional state variable, which encompasses the past information regarding satisfaction of the constraint. Thus, the problem transforms into a state-constrained MDP problem, which is easier to handle with standard MDP tools.

Notice that the burstiness constraint (see Equation (1.1)), which is the center of the present study, introduces a higher level of complexity than the above constraint. Whereas the latter limits only the long-term total discounted cost and includes one inequality, the former puts a bound on any sum of consecutive elements, yielding an infinite amount of inequalities that should be satisfied. Aside from the different problems that are solved, in our work we discuss the conditions for the problem's feasibility, and propose algorithms to determine whether or not a feasible policy exists.

Sample-path constraints are useful when one would like the optimal policy to be more sensitive to deviation of the cost from its expected value. Altman [Alt99] (Section 1.3) mentions several other types of constraints, and see also White [Whi88], Filar, Krass and Ross [FKR95] and Chamie and Açikmeşe [ECA15].

2.3 The Burstiness Constraint

Bursty processes cannot be analyzed with the traditionally employed models such as Poisson or renewal processes [KA95]. Several models for burstiness have been proposed, e.g., the peakedness formulation by Eckberg and Lucantoni [EL90], the (σ, ρ)-regulated process by Cruz [Cru91], and the peak-to-median ratio by Chen, Alspaugh and Katz [CAK12]. Starobinsky and Sidi [StaS99] addressed processes whose burstiness is stochastically bounded by general decreasing functions, and developed a network calculus for this setting.

Recall the (σ, ρ) burstiness formulation, which was presented in the Introduction (Chapter 1):

$$\sum_{t=t_1}^{t_2} d_t \leq \rho(t_2 - t_1 + 1) + \sigma, \quad \forall t_1, t_2 \in \mathcal{T} : \; t_1 \leq t_2. \tag{2.12}$$

This forumlation is closely tied to the *leaky bucket* congestion control scheme, which has several alternative forms [Tur86], [Val01]. In general, the model consists of a "bucket" of tokens, which starts with σ tokens. At every time point, ρ tokens are added to the bucket. In order to insert a packet of data into the queue, a packet-dependent number of tokens must be taken from the bucket. If the bucket lacks the required amount of tokens, then the packet is either discarded or delayed until a proper amount of tokens is generated. The bucket can only hold up to σ tokens at any given point, and any exceeding tokens are dropped. Eckberg and Lucantoni [EL90] model the performance of the leaky bucket throughtput-burstiness filter, in terms of the peakedness of the output data, given a Markov-modulated Poisson process (MMPP) input. Low and Varaiya [LV91] show that the leaky bucket model reduces burstiness of the input data.

Clearly, a sequence which obeys constraint (2.12) would pass the leaky bucket filter without dropping a single packet, when d_t describes the number of tokens required at each time point by the incoming data packet. Alternatively, any (σ, ρ)-burstiness constrained sequence can be viewed as the output of a leaky bucket filter for some input data.

Cruz in [Cru91] analyzes the impact of various network elements, such as a constant delay line, a receive buffer and a (σ, ρ) regulator, on the burstiness of their output data, given that the input is (σ, ρ)-burst constrained.

Konstantopoulos and Anantharam [KA95] establish the optimality of the leaky bucket scheme for regulating traffic burstiness. They show that a process's burstiness can be regulated by measuring the bucket's backlog using reflection mappings, and keeping it below σ. Note that in the setting considered by Konstantopoulos and Anantharam, a burstiness-constrained output can always be obtained, since the possibility of discarding or delaying the input data packet always exists. This is equivalent to setting $d_t = 0$ at those time points. The goal, in their case, is to minimize the number of discarded packets (or maximize the system's throughput), while obeying burstiness constraints on the data.

In this work, we consider a more general setting. First, the system consists of two sequences: a reward sequence, $(r_t(s_t, a_t))_{t \in \mathcal{T}}$, which we would like to maximize some function of, and a cost sequence, $(d_t(s_t, a_t))_{t \in \mathcal{T}}$, which is required to obey burstiness constraints. In addition, the option of discarding or delaying an element d_t does not exist: the sets of available actions $A_t(s_t)$ are pre-determined, and there does not necessarily exist an action for which $d_t(s, a) = 0$. Thus, a control policy under which d_t is (σ, ρ)-burst constrained does not necessarily exist, and it is up to us to determine whether or not the problem is at all feasible, and to find the optimal policy under this constraint.

Chapter 3

BCP Problem Formulation

In this chapter we introduce Burstiness-Constrained MDPs (BCPs). Section 3.1 presents the setting of a BCP. Sections 3.2 and 3.3 formulate the feasibility and optimization problems under discussion, respectively. In Section 3.4 we describe the challenges of solving BCP problems, which we then demonstrate in Section 3.6.

3.1 BCP Model and Notations

The BCP setting is similar to an MDP's, with the addition of a set of cost functions and burstiness coefficients. A BCP is defined by an 8-tuple of the form $\langle \mathcal{T}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}; \mathcal{D}, \sigma, \rho \rangle$, where $\langle \mathcal{T}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ constitute a discrete-time MDP with finite state and action spaces (see section 2.1). The additional terms are defined as follows.

- $\mathcal{D} \triangleq (d_t(s,a))_{t \in \mathcal{T}}$ is a *cost structure*, consisting of a sequence of K-dimensional cost functions for every time point, where $s \in S_t,\ a \in A_t(s)$ and $d_t(s,a) \in \mathbb{R}^K$. At every time point, a cost $d_t(s_t, a_t)$ is incurred on the system. Burstiness constraints are imposed on the vector-sequence of incurred costs, $(d_t(s_t,a_t))_{t \in \mathcal{T}}$. In finite horizon BCPs, the cost in the last time point depends only on the current state: $d_N(s),\ s \in S_N$.

- $\sigma \in \mathbb{R}^K$ and $\rho \in \mathbb{R}^K$ are K-dimensional *burstiness coefficient vectors*, used to characterize the burstiness constraint on $\mathcal{D}$.

In infinite horizon BCPs, we assume stationarity of the model, so that the cost function is time-invariant:

$$d_t(\cdot,\cdot) \equiv d(\cdot,\cdot).$$

3.2 BCP Feasibility

When the sequence of instantaneous cost vectors, $(d_t(s_t, a_t))_{t\in\mathcal{T}}$, satisfies the burstiness constraint (1.1) component-wise w.r.t. the burstiness coefficient vectors (σ, ρ), we say that *the burstiness constraint is obeyed*. Given an initial state $s \in S_0$ and policy π, adherence to the burstiness constraint is an event (that is, a set of possible outcomes), which we denote by

$$B_{\sigma,\rho} \triangleq \left\{ \omega \in \Omega : \quad \sum_{t=t_1}^{t_2} d_t(s_t, a_t) \leq \rho(t_2 - t_1 + 1) + \sigma, \quad \forall t_1, t_2 \in \mathcal{T} : t_1 \leq t_2 \right\}. \tag{3.1}$$

Given an initial state $s \in S_0$, a policy π is said to be feasible for s if $B_{\sigma,\rho}$ has probability 1 given π and s, or equivalently, $\mathbb{P}^{\pi,s}(B_{\sigma,\rho}) = 1$. An initial state $s \in S_0$ is called feasible if there exists a feasible policy for it. Denote by S_0^F the set of feasible initial states,

$$S_0^F \triangleq \left\{ s \in S_0 : \ \exists \pi \in \Pi^{HR} \text{ s.t. } \mathbb{P}^{\pi,s}(B_{\sigma,\rho}) = 1 \right\}.$$

The problem of determining whether or not a feasible policy exists for any initial state is articulated via the following feasibility problem.

BCF: Given a BCP, find $\Phi(s)$ for any $s \in S_0$, where

$$\Phi(s) \triangleq \begin{cases} 1, & \text{if } \exists \pi \in \Pi^{HR} : \ \mathbb{P}^{\pi,s}(B_{\sigma,\rho}) = 1; \\ 0, & \text{otherwise.} \end{cases}$$

We refer to $\Phi(s)$ as the BCP's *feasibility indicator function*. Observe that

$$S_0^F = \left\{ s \in S_0 : \ \Phi(s) = 1 \right\}.$$

3.3 BCP Optimization Problem

Assuming that there exist feasible policies for the BCP, we would like to optimize some objective function of the acquired rewards, over all feasible history-dependent randomized control policies. The objective function may take any of the standard forms: In finite horizon BCPs,

$$J_N^\pi(s) \triangleq \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{N-1} r_t(s_t, a_t) + r_N(s_N) \right],$$

and in infinite horizon BCPs,

$$J_{\gamma}^{\pi}(s) \triangleq \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t)\right], \text{ or } J_{\text{ea}}^{\pi}(s) \triangleq \liminf_{N\to\infty}\left\{\tfrac{1}{N+1}\mathbb{E}^{\pi,s}\left[\sum_{t=0}^{N}r(s_t,a_t)\right]\right\},$$

where π is the control policy, $s \in S_0$ is the initial state, and $\gamma \in (0,1)$.

The BCP optimization problem is formulated as follows.

BCP: Given a BCP and objective $o \in \{N, \gamma, \text{ea}\}$, find $\pi_o^{F^*} \in \Pi^{HR}$ such that

$$J_o^{\pi_o^{F^*}}(s) = \max_{\pi\in\Pi^{HR}} \left\{J_o^{\pi}(s) : \ \mathbb{P}^{\pi,s}(B_{\sigma,\rho}) = 1\right\}, \ \forall s \in S_0^F. \tag{3.2}$$

The parameter o denotes the objective's type. The optimal objective function is denoted by $J_o^{F^*}(s) \triangleq J_o^{\pi_o^{F^*}}(s)$, and referred to as the *value* of the problem.

3.4 Remarks on BCP Feasibility and Optimization

We make several observations about the **BCF** and **BCP** problems.

1. In order to determine whether or not each initial state is feasible, we need a way to determine whether or not any possible control policy is feasible.

2. Since feasibility of a policy requires that the burstiness constraint hold with probability 1, a policy is feasible only if *any possible* instantiation of the sample path, given the policy and initial state, obeys the burstiness constraint.

3. The number of inequalities in the burstiness constraint in finite horizon MDPs is of the order of $N^2/2$ (N being the length of the time-horizon), and infinite in inifinite horizon MDPs. Thus, an exhaustive inspection for each policy's feasibility is prohibitive.

4. Due to the constraint's complicated form, we cannot solve **BCP** with the standard **MDP** solution tools such as backward induction, value iteration, policy iteration and linear programming algorithms.

5. As the constraints are imposed on the system's *actual* sample-path, rather than on the *expectation* of some fucntion of the sample-path, we cannot express the constraint as a linear combination of state-action frequencies, such as seen in the examples of Section 2.2.

Despite the difficulties described above, the burstiness constraint can be reformulated as a state constraint, which enables simple methods to determine the feasibility of states and policies. By posing the BCP as a state-constrained MDP, we can extend the standard MDP tools appropriately to our case. We lay out this reformulation in the next chapter.

3.5 Simple Necessary Conditions and Sufficient Conditions for Feasiblity

Despite the complex structure of the burstiness constraint, we can formulate simple sufficient conditions and (separate) necessary conditions for feasibility of the BCP. We present these conditions below. Tighter (necessary and sufficient) conditions will be derived in the following chapters.

Claim 3.1 (Necessary Condition)**.** *In a fully-communicating BCP (i.e., when there is a nonzero probability of reaching any state from any state under any policy), a necessary condition for feasibility is that for every time point $t \in \mathcal{T}$ and state $s \in S_t$ there exists some action $a \in A_t(s)$ s.t. $d_t(s,a) \leq \rho + \sigma$.*

Proof. Observe from Equation (3.1) that if a policy π is feasible for some initial state $s_0 \in S_0$, then in particular it should satisfy

$$\mathbb{P}^{\pi,s}(d_t(s_t,a_t) \leq \rho + \sigma, \quad \forall t \in \mathcal{T}) = 1.$$

Recall that in a fully-communicating BCP, any state $s_t \in S_t$ is accessible from any initial state under any policy. Thus, in order for there to exist a feasible policy in such a BCP, in particular at any time point $t \in \mathcal{T}$ and state $s \in S_t$ there should be an action $a \in A_t(s)$ for which $d_t(s,a) \leq \rho + \sigma$.

■

Claim 3.2 (Sufficient Condition)**.** *Given a BCP, if for every time point $t \in \mathcal{T}$ and state $s \in S_t$ there exists an action $a \in A_t(s)$ s.t. $d_t(s,a) \leq \rho$, then a feasible policy exists.*

Proof. Assume that for every time point $t \in \mathcal{T}$ and every state $s \in S_t$ there exists an action $a \in A_t$ s.t. $d_t(s,a) \leq \rho$, and consider the following policy: at any time point

$t \in \mathcal{T}$, choose the action $a_t = a \in A_t(s_t)$ for which $d_t(s_t, a) \leq \rho$. Under this policy, any achievable outcome $\omega \in \Omega$ will satisfy the burstiness constraint,

$$\sum_{t=t_1}^{t_2} d_t(s_t, a_t) \leq \rho(t_2 - t_1 + 1) + \sigma, \quad \forall t_1, t_2 \in \mathcal{T} : \ t_1 \leq t_2.$$

Therefore, under this condition there exists a feasible policy for the BCP.

■

3.6 Examples

The following examples demonstrate the difficulties presented in the previous section, and show that stationary and/or deterministic policies are insufficient for the problem in its current form. First, consider a stationary infinite horizon BCP with one state and two available actions, as depicted in Figure 3.1. The cost function, d, is scalar, and $\sigma = \rho = 1$. Obviously, at time $t = 0$ we cannot perform $a_0 = a^{(1)}$, otherwise the BC

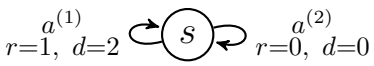

Figure 3.1: Infinite horizon example.

would be violated. Observe that constantly performing action $a^{(2)}$ would lead to an expected discounted reward of 0. Conversely, a periodic control policy, where action $a^{(1)}$ is performed once in every 3 time steps, and otherwise action $a^{(2)}$ is performed, would result with an expected discounted reward of $\frac{1}{1-\gamma^3}$. We conclude that we cannot narrow the search for an optimal feasible policy to stationary policies alone, but rather that the optimal feasible policy is time (or history) dependent.

The second example is set in a non-stationary environment. Consider the finite horizon BCP portrayed in Figure 3.2. Again, we use a scalar cost function d, and $\sigma = \rho = 1$.

The burstiness constraint for this BCP consists of the following inequalities:

$$\begin{aligned} d_0 &\leq 2, \quad d_0 + d_1 \leq 3, \\ d_1 &\leq 2, \quad d_1 + d_2 \leq 3, \\ d_2 &\leq 2, \quad d_0 + d_1 + d_2 \leq 4. \end{aligned}$$

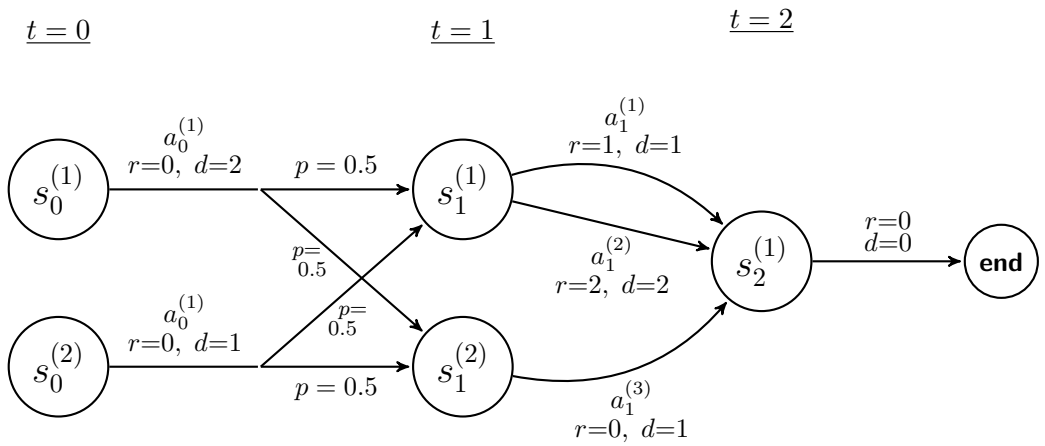

Figure 3.2: Finite horizon example.

We see that when the process starts at $s_0 = s_0^{(1)}$ and then proceeds to $s_1 = s_1^{(1)}$, choosing action $a_1 = a_1^{(2)}$ will lead to violation of the BC. Consider the Markov policy $\pi^M \in \Pi^{MD}$ where in state $s_1 = s_1^{(1)}$ action $a_1 = a_1^{(1)}$ is always chosen, such that $\pi_1^M(s_1^{(1)}) = a_1^{(1)}$. Clearly, the burstiness constraint will be obeyed under this policy. When the process starts at $s_0 = s_0^{(2)}$, the expected total reward is $\mathbb{E}^{s_0^{(2)},\pi^M}\left[\sum_{t\in\mathcal{T}} r_t\right] = 0.5$. Alternatively, consider the history-dependent policy $\pi^H \in \Pi^{HD}$ where in state $s_1 = s_1^{(1)}$, if the previous state was $s_0 = s_0^{(1)}$ then action $a_1 = a_1^{(1)}$ is chosen, and if the previous state was $s_0 = s_0^{(2)}$ then we choose action $a_1 = a_1^{(2)}$, such that $\pi_1^H(s_0^{(1)}, a_0^{(1)}, s_1^{(1)}) = a_1^{(1)}$ and $\pi_1^H(s_0^{(2)}, a_0^{(1)}, s_1^{(1)}) = a_1^{(2)}$. In this case, the burstiness constraint is obeyed always as well, and the expected total reward when $s_0 = s_0^{(2)}$ is $\mathbb{E}^{s_0^{(2)},\pi^H}\left[\sum_{t\in\mathcal{T}} r_t\right] = 1$. Hence, in this BCP, Markov policies are not sufficient to achieve optimality of the objective function while obeying the BC. A similar conclusion can be achieved in case of a deterministic MDP and stochastic cost function $d_t(\cdot,\cdot)$.

Chapter 4

Reformulation as a State-Constrained MDP

As previously mentioned in Section 3.4, the BCP can be posed as a state-constrained MDP, which can then be handled with standard MDP tools. The reformulation is done by restructuring the burstiness constraint into a simpler form and augmenting the system's state variable. Section 4.1 explains the motivation for these steps. The Reformulated BCP (RBCP) model is then summarized in Section 4.2. Section 4.3 establishes the correspondence between the BCP and its reformulation, the RBCP. We analyze the RBCP and discuss its properties in Section 4.4. We comment on other possible reformulations for the BCP in Section 4.5.

4.1 Burstiness Constraint Reformulation and State Augmentation

The rationale behind reformulation of the BCP as a state-constrained MDP is as follows. We first observe that the extensive set of inequalities in the burstiness constraint can be reduced to one inequailty per time point. At each inequality, the necessary past information may be epitomized in a single vector quantity, $y_t \in \mathbb{R}^K$. By appending y_t to the BCP's state variable, the burstiness constraint turns into a set of simple constraints on the augmented state variable, each associated with a single time point. In this form, the feasibility of a policy is easier to determine. In addition, we can narrow the optimization domain to Markov policies in the augmented state-space, thus simplifying the search for an optimal policy.

Lemma 4.1. *The event of obeying the burstiness constraint, denoted by $B_{\sigma,\rho}$ in Equa-*

tion (3.1)*, can be expressed by*

$$B_{\sigma,\rho} = \{\omega \in \Omega : \ z_t \leq \sigma, \ \forall t \in \mathcal{T}\}, \tag{4.1}$$

where

$$z_t \triangleq \max_{t_1 \in \{0,\ldots,t\}} \left\{ \sum_{t'=t_1}^{t} (d_{t'}(s_{t'}, a_{t'}) - \rho) \right\}, \quad \textit{for} \ \ t \in \mathcal{T}. \tag{4.2}$$

Furthermore, z_t *can be obtained recursively by*

$$z_0 = d_0(s_0, a_0) - \rho; \quad z_{t+1} = z_t^+ + d_{t+1}(s_{t+1}, a_{t+1}) - \rho, \ \forall t \in \mathcal{T}. \tag{4.3}$$

Proof. For simplicity, denote $d_t = d_t(s_t, a_t)$. Recall the definition of $B_{\sigma,\rho}$, and observe the equivalence of the following statements:

$$\begin{aligned}
&\sum_{t=t_1}^{t_2} d_t \leq \rho(t_2 - t_1 + 1) + \sigma, \quad \forall t_1, t_2 \in \mathcal{T} : \ t_1 \leq t_2 \\
\Leftrightarrow \ &\left\{ \sum_{t=t_1}^{t_2} (d_t - \rho) \leq \sigma, \quad \forall t_1 \in \{0, ..., t_2\} \right\}, \quad \forall t_2 \in \mathcal{T} \\
\Leftrightarrow \ &\max_{t_1 \in \{0,\ldots,t_2\}} \left\{ \sum_{t=t_1}^{t_2} (d_t - \rho) \right\} \leq \sigma, \quad \forall t_2 \in \mathcal{T},
\end{aligned}$$

thus proving Equation (4.1).

To prove the equivalence of Equations (4.2) and (4.3), observe that by Equation (4.2), indeed $z_0 = d_0 - \rho$, and for any $t \in \mathcal{T}$,

$$\begin{aligned}
z_{t+1} &= \max_{t_1 \in \{0,\ldots,t+1\}} \left\{ \sum_{t'=t_1}^{t+1} (d_{t'} - \rho) \right\} \\
&= \max_{t_1 \in \{0,\ldots,t+1\}} \left\{ \sum_{t'=t_1}^{t} (d_{t'} - \rho) \right\} + d_{t+1} - \rho \\
&= \max \left\{ \max_{t_1 \in \{0,\ldots,t\}} \left\{ \sum_{t'=t_1}^{t} (d_{t'} - \rho) \right\}, 0 \right\} + d_{t+1} - \rho \\
&= z_t^+ + d_{t+1} - \rho.
\end{aligned}$$

■

In Equation (4.1), the burstiness constraint is expressed as a set of inequalities, one per time point. However, since z_t depends on the current action, a_t, we cannot use it as a state variable. Instead, we define a similar term, y_t, which embodies the entire *past*

information that is relevant at time t.

Lemma 4.2. *The event of adherence to the burstiness constraint, denoted by $B_{\sigma,\rho}$ in Equation* (3.1)*, satisfies*

$$B_{\sigma,\rho} = \{\omega \in \Omega : \ y_t + d_t(s_t, a_t) - \rho \leq \sigma, \ \forall t \in \mathcal{T}\}, \tag{4.4}$$

where

$$y_0 \triangleq 0, \quad \textit{and}$$
$$y_t \triangleq \left(\max_{t_1 \in \{0,\ldots,t-1\}} \left\{ \sum_{t'=t_1}^{t-1} (d_{t'}(s_{t'}, a_{t'}) - \rho) \right\} \right)^+, \quad \textit{for} \ \ t \in \mathcal{T} \backslash \{0\}. \tag{4.5}$$

The term y_t can be obtained recursively by

$$y_0 = 0; \quad y_{t+1} = (y_t + d_t(s_t, a_t) - \rho)^+, \ \forall t \in \mathcal{T}. \tag{4.6}$$

Proof. To prove Equation (4.4), we use Equation (4.1) of Lemma 4.1 and show that $z_t = y_t + d_t - \rho$ as follows.

$$\begin{aligned} y_t + d_t - \rho &= \max \left\{ 0, \max_{t_1 \in \{0,\ldots,t-1\}} \left\{ \sum_{t'=t_1}^{t-1} (d_{t'} - \rho) \right\} \right\} + d_t - \rho \\ &= \max \left\{ d_t - \rho, \max_{t_1 \in \{0,\ldots,t-1\}} \left\{ \sum_{t'=t_1}^{t} (d_{t'} - \rho) \right\} \right\} \\ &= \max_{t_1 \in \{0,\ldots,t\}} \left\{ \sum_{t'=t_1}^{t} (d_{t'} - \rho) \right\} = z_t. \end{aligned} \tag{4.7}$$

For Equation (4.6), use Equations (4.3) and (4.7) to verify that $y_0 = z_0 - d_0 + \rho \equiv 0$, and

$$y_{t+1} = z_{t+1} - d_{t+1} + \rho = z_t^+ = (y_t + d_t - \rho)^+.$$

■

In Equation (4.5), y_t is defined as the maximum over all time points until t, of the accumulated deviation of previous immediate costs from ρ. Continuing the bucket of tokens analogy of the burstiness constraint, in section 2.3, y_t embodies the maximum possible "deficit" accumulated by time t (for brevity, we call y_t the *deficit* at time t). At each time point there are at most $\sigma - y_t$ "credit" points left from the past steps.

The selected action should satisfy $d_t(s_t, a_t) \leq \sigma - y_t + \rho$ at each step in order to fulfill the burstiness constraint. Equation (4.6) complies with this explanation of y_t: The process starts with zero initial deficit; At each time point, the new deficit depends on the previous deficit, y_t, and the new cost deviation, $d_t - \rho$.

Since each inequality in Equation (4.4) involves only the current state, action and deficit (respectively, s_t, a_t and y_t), by appending y_t to the state variable, $B_{\sigma,\rho}$ gets the form of a state constraint.

Towards this end, we augment the BCP's state variable to include y_t. Denote by $Y_t(\cdot)$ the state-sets of the new state variable, y_t. In explicit form, we can use any sets such that

$$\begin{aligned} &Y_0(s) \supseteq \{0\},\ \forall s \in S_0,\ \text{and} \\ &Y_{t+1}(s') \supseteq \big\{y' \in \mathbb{R}^K :\ y' = (y + d_t(s,a) - \rho)^+ \text{ where} \\ &\qquad\qquad s \in S_t,\ y \in Y_t(s),\ a \in A_t(s) \text{ and } p_t(s'|s,a) > 0\big\},\ \forall t \in \mathcal{T},\ s' \in S_{t+1}. \end{aligned} \tag{4.8}$$

Since S_t and $A_t(s)$ are finite, the minimal sets of possible y_t values are finite as well. However, these sets are problem-dependent and in general are hard to retrieve. On the other hand, we can always choose $Y_t(s) = [0, \infty)^K$, but this may complicate the calculations beyond necessary. We address these issues in the Finite Horizon Algorithms chapter (Chapter 5).

The reformulated MDP uses the same action space and reward structure as the BCP, respectively $\mathcal{A} = (A_t(s))_{t\in\mathcal{T},\ s\in S_t}$ and $\mathcal{R} = (r_t(\cdot,\cdot))_{t\in\mathcal{T}}$.

For stationarity of the reformulated BCP in case of a stationary infinite-horizon BCP, we need a state-space $Y(s)$ such that $Y(s) \supseteq \bigcup_{t\in\mathcal{T}} Y_t(s)$ for the additional state variable y_t. It is defined implicitly as any set such that

$$\begin{aligned} &Y(s) \supseteq \{0\},\ \forall s \in S,\ \text{and} \\ &Y(s') \supseteq \big\{y' \in \mathbb{R}^K :\ y' = (y + d(s,a) - \rho)^+ \text{ where} \\ &\qquad\qquad s \in S,\ y \in Y(s),\ a \in A(s) \text{ and } p(s'|s,a) > 0\big\},\ \forall s' \in S. \end{aligned} \tag{4.9}$$

In the case where $d(s,a),\ \rho$ and σ are all integer multiples of some basic quantity (e.g., are all inetgers), this is a discrete set. In the general case, $Y(s)$ is a continuous domain. The issue of selecting a suitable compact state-set for a stationary infinite horizon BCP is addressed in the Infinite Horizon Algorithms chapter (Chapter 6).

For the purpose of analysis, it will be useful to extend the burstiness constraint to the case of a nonzero initial deficit, by defining the *extended burstiness constraint*. Given an initial state $s \in S_0$ and $y \in \mathbb{R}^K$, and a control policy, adhrenece to the extended burstiness contraint is the following event:

$$B(y) \triangleq \{\omega \in \Omega : \quad y_t + d_t(s_t, a_t) - \rho \leq \sigma, \quad \forall t \in \mathcal{T} \text{ where} \\ y_0 = y \text{ and } y_{t+1} = (y_t + d_t(s_t, a_t) - \rho)^+, \ \forall t \in \mathcal{T}\}, \quad (4.10)$$

Evidently, $B(0) \equiv B_{\sigma,\rho}$.

4.2 State-Constrained MDP Formulation

4.2.1 RBCP Model and Notations

We summarize the reformulated BCP (RBCP) model herein. Given a BCP $\langle \mathcal{T}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}; \mathcal{D}, \sigma, \rho \rangle$ as formulated in Chapter 3, an RBCP is the 8-tuple of the form $\langle \mathcal{T}, \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}; \mathcal{D}, \sigma, \rho \rangle$, where $\langle \mathcal{T}, \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R} \rangle$ constitute a discrete-time MDP with finite state and action spaces, and:

- The time horizon, $\mathcal{T}$, is the same as in the BCP.
- The state-space is $\tilde{\mathcal{S}} \triangleq (\tilde{S}_t)_{t \in \mathcal{T}}$, a sequence of finite sets of available states for every time point, where $\tilde{S}_t \triangleq \{(s, y) : \ s \in S_t, \ y \in Y_t(s)\}$, and $Y_t(s)$ are finite sets, chosen according to Equation (4.8). The system's state at each time point is denoted by $\tilde{s}_t \in \tilde{S}_t$, and is of the form $\tilde{s}_t \triangleq (s_t, y_t)$ such that $s_t \in S_t$ and $y_t \in Y_t(s_t)$.
- The action space is the same as in the BCP, $\mathcal{A} = (A_t(s))_{t \in \mathcal{T}, \ s \in S_t}$. We denote by $a_t \in A_t(s_t)$ the action performed at time $t \in \mathcal{T}$ (in finite horizon BCPs, no action is performed on the last time point).
- The dynamics of the s_t component of the state variable is the same as in the BCP, stochastically governed by $p_t(\cdot|\cdot, \cdot)$. The dynamics of y_t is deterministically controlled by the current state and action, by $y_{t+1} = (y_t + d_t(s_t, a_t) - \rho)^+$. Hence, the RBCP's probability kernel is $\tilde{\mathcal{P}} \triangleq (\tilde{p}_t(\tilde{s}'|\tilde{s}, a))_{t \in \mathcal{T}}$ where $\tilde{s} = (s, y) \in \tilde{S}_t, \ a \in A_t(s), \ \tilde{s}' = (s', y') \in \tilde{S}_{t+1}$ and

$$\tilde{p}_t\left(\tilde{s}'|\tilde{s}, a\right) \triangleq p_t(s'|s, a) \cdot \mathbb{1}(y' = (y + d_t(s, a) - \rho)^+),$$

  such that $\tilde{p}_t(\cdot|\tilde{s}, a) \in \mathscr{P}(\tilde{S}_{t+1})$ and $\mathbb{P}\left(\tilde{s}_{t+1} = \tilde{s}'|\tilde{s}_t = \tilde{s}, \tilde{a}_t = a\right) = \tilde{p}_t\left(\tilde{s}'|\tilde{s}, a\right)$.

- The reward structure remains unchanged, $\mathcal{R} = (r_t(s,a))_{t\in\mathcal{T}}$ where $s \in S_t,\ a \in A_t(s)$ and $r_t(s,a) \in \mathbb{R}$ (in finite horizon BCPs, the reward in the last time point depends only on the current state).

In the case of a stationary infinite horizon BCP, the RBCP is stationary as well, omitting the temporal subscripts of the problem parameters. The resulting y-state-space is $Y(s) = \bigcup_{t\in\mathcal{T}} Y_t(s)$, and can be defined implicitly by Equation (4.9). The following notations are introduced as well:

$$\tilde{S}_t \equiv \tilde{S} \triangleq \{(s,y) :\ s \in S,\ y \in Y(s)\} \ \text{ and } \ \tilde{p}_t(\cdot|\cdot,\cdot) \equiv \tilde{p}(\cdot|\cdot,\cdot).$$

In the RBCP setting, we denote by $\tilde{H}_t$ the set of all possible t-histories:

$$\tilde{H}_t \triangleq \left\{(s_0, y_0, a_0, ..., s_t, y_t) :\ (s_{t'}, y_{t'}) \in \tilde{S}_{t'},\ a_{t'} \in A_{t'}(s_{t'}),\ \ \forall t' \le t\right\}, \ \text{ for } \ t \in \mathcal{T}.$$

The RBCP's sample space is $\tilde{\Omega} \triangleq \tilde{H}_N$ in the finite horizon case, and $\tilde{\Omega} \triangleq \tilde{H}_\infty$ in the infinite horizon case, where

$$\tilde{H}_\infty \triangleq \left\{(s_0, y_0, a_0, ...) :\ (s_t, y_t) \in \tilde{S}_t,\ a_t \in A_t(s_t),\ \forall t \in \{0, 1, ...\}\right\}.$$

Denote by $\tilde{\Pi}^{HR}$ and $\tilde{\Pi}^{MD}$ the sets of history-dependent, randomized control policies and Markov determinisitc control policies over RBCPs, respectively, and recall that $\tilde{\Pi}^{MD} \subset \tilde{\Pi}^{HR}$ (see Section 2.1.2 for definitions of the policy sets). Denote by $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\cdot)$ and $\tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}[\cdot]$ the probability measure and expectation operator, respectively, that are indcued when the RBCP starts with initial state $(s,y) \in \tilde{S}_0$ and uses the control policy $\tilde{\pi}$.

4.2.2 RBCP Feasibility

Given an initial RBCP state $(s,y) \in \tilde{S}_0$ and policy $\tilde{\pi}$, we define the event of obeying the extended burstiness constraint in the augmented state-space, $\tilde{B}$:

$$\tilde{B} \triangleq \left\{\tilde{\omega} \in \tilde{\Omega} :\ \ y_t + d_t(s_t, a_t) - \rho \le \sigma,\ \ \forall t \in \mathcal{T}\right\}. \tag{4.11}$$

A policy $\tilde{\pi}$ is said to be feasible for the initial state $(s,y) \in \tilde{S}_0$ if $\tilde{B}$ holds with probability 1 given $\tilde{\pi}$ and (s,y), or equivalently, $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1$. An initial state $(s,y) \in \tilde{S}_0$ is called feasible if there exists a feasible policy for it. Denote by $\tilde{S}_0^F$ the set

of feasible initial RBCP states,

$$\tilde{S}_0^F \triangleq \left\{(s,y) \in \tilde{S}_0 :\ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\right\}. \tag{4.12}$$

The problem of determining whether or not a feasible policy exists for any initial state of the RBCP is articulated via the following feasibility problem.

RBCF: Given an RBCP, find $\tilde{\Phi}(s,y)$ for any $(s,y) \in \tilde{S}_0$, where

$$\tilde{\Phi}(s,y) \triangleq \begin{cases} 1, & \text{if } \exists \tilde{\pi} \in \tilde{\Pi}^{HR} :\ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1; \\ 0, & \text{otherwise.} \end{cases}$$

We refer to $\tilde{\Phi}(s,y)$ as the RBCP's *feasibility indicator function.* Observe that

$$\tilde{S}_0^F = \left\{(s,y) \in \tilde{S}_0 :\ \tilde{\Phi}(s,y) = 1\right\}.$$

4.2.3 RBCP Optimization Problem

The RBCP's objective functions take the same form as the BCP's, with the expectation operator taken w.r.t. the control policy $\tilde{\pi}$ and initial state $(s,y) \in \tilde{S}_0$: In the finite horizon case,

$$\tilde{J}_N^{\tilde{\pi}}(s,y) \triangleq \tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}\left[\sum_{t=0}^{N-1} r_t(s_t,a_t) + r_N(s_N)\right],$$

and in the infinite horizon case,

$$\tilde{J}_\gamma^{\tilde{\pi}}(s,y) \triangleq \tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}\left[\sum_{t\in\mathcal{T}} \gamma^t r_t(s_t,a_t)\right], \quad \text{or} \quad \tilde{J}_{\text{ea}}^{\tilde{\pi}}(s,y) \triangleq \liminf_{N\to\infty}\left\{\tfrac{1}{N+1}\tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}\left[\sum_{t=0}^{N} r_t(s_t,a_t)\right]\right\},$$

where $\gamma \in (0,1)$.

The reformulated optimization problem seeks to optimize the objective $\tilde{J}_o^{\tilde{\pi}}(s,y)$ over all feasible history-dependent randomized RBCP control policies, and is articulated as follows.

RBCP: Given an RBCP and objective $o \in \{N, \gamma, \text{ea}\}$, find $\tilde{\pi}_o^{F^*} \in \tilde{\Pi}^{HR}$ such that

$$\tilde{J}_o^{\tilde{\pi}_o^{F^*}}(s,y) = \max_{\tilde{\pi}\in\tilde{\Pi}^{HR}} \{\tilde{J}_o^{\tilde{\pi}}(s,y) :\ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\},\ \forall (s,y) \in \tilde{S}_0^F. \tag{4.13}$$

Denote the problem's value by $\tilde{J}_o^{F^*}(s,y) \triangleq \tilde{J}_o^{\tilde{\pi}_o^{F^*}}(s,y)$.

Using the burstiness constraint's formulation as a state constraint, we can derive simple algorithms to solve **RBCF**. In addition, since the RBCP is a state-constrained MDP, we can revise the standard MDP algorithms to accomodate **RBCP**, yielding simple and efficient solution algorithms.

4.3 Correspondence Between BCP and RBCP

We argue that there exists a correspondence between BCP and RBCP policies, such that the feasibility and objective function of each BCP policy can be determined through the feasibility and objective function of a corresponding RBCP policy.

First, let us recall the form of history-dependent randomized policies. With slight abuse of notation, a history-dependent randomized BCP policy $\pi \in \Pi^{HR}$ is of the form $\pi = (\pi_t)_{t\in\mathcal{T}}$ such that $\pi_t(\cdot|h_t) \in \mathscr{P}(A_t(s_t)),\ \forall t \in \mathcal{T},\ h_t = (s_0, a_0, ..., s_t) \in H_t$. Similarly, a history-dependent randomized RBCP policy $\tilde{\pi} \in \tilde{\Pi}^{HR}$ is of the form $\tilde{\pi} = (\tilde{\pi}_t)_{t\in\mathcal{T}}$ such that $\tilde{\pi}_t(\cdot|\tilde{h}_t) \in \mathscr{P}(A_t(s_t)),\ \ \forall t \in \mathcal{T},\ \tilde{h}_t = (s_0, y_0, a_0, ..., s_t, y_t) \in \tilde{H}_t$.

Lemma 4.3. *Given an RBCP policy $\tilde{\pi} \in \tilde{\Pi}^{HR}$ and $y \in \mathbb{R}^K$, consider any corresponding BCP policy $\pi \in \Pi^{HR}$ for which*

$$\pi_t(a|h_t) = \tilde{\pi}_t(a|\tilde{h}_t), \quad \forall t \in \mathcal{T},\ h_t = (s_0, a_0, ..., s_t) \in H_t,\ a \in A_t(s_t), \tag{4.14}$$

where $\tilde{h}_t \in \tilde{H}_t$ is obtained from h_t by

$$\tilde{h}_t = (s_0, y_0, a_0, ..., s_t, y_t)\,,$$

such that $y_0 = y$ and $y_{t+1} = (y_t + d_t(s_t, a_t) - \rho)^+,\ \ \forall t \in \mathcal{T}$.

Then, for any $s \in S_0$,

$$\mathbb{P}^{\pi,s}(B(y)) = 1 \quad \Leftrightarrow \quad \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1, \tag{4.15}$$

and

$$J_o^{\pi}(s) = \tilde{J}_o^{\tilde{\pi}}(s, y), \tag{4.16}$$

where $o \in \{N, \gamma, ea\}$ is the objective's type.

Similarly, given a BCP policy $\pi \in \Pi^{HR}$ and $y \in \mathbb{R}^K$, consider any corresponding RBCP policy $\tilde{\pi} \in \tilde{\Pi}^{HR}$ for which

$$\begin{aligned}\tilde{\pi}_t(a|\tilde{h}_t) &= \pi_t(a|h_t),\\ &\forall t \in \mathcal{T},\ \ \tilde{h}_t = (s_0, y_0, a_0, ..., s_t, y_t) \in \{\tilde{H}_t :\ y_0 = y\},\ a \in A_t(s_t),\end{aligned} \tag{4.17}$$

where $h_t \in H_t$ is obtained from $\tilde{h}_t$ by $h_t = (s_0, a_0, ..., s_t)$. Then, for any $s \in S_0$, Equations (4.15) *and* (4.16) *are satisfied.*

Proof. Recall from the RBCP's state dynamics that y_t is a deterministic function of the previous state and action. Thus, given an initial y_0 and partial RBCP history $(s_t, a_t)_{t\in\mathcal{T}}$, the entire history $\tilde{h}_t = (\tilde{s}_t, a_t)_{t\in\mathcal{T}}$ can be reconstructed by using $\tilde{s}_t = (s_t, y_t)$ where $y_{t+1} = (y_t + d_t(s_t, a_t) - \rho)^+,\ \forall t \in \mathcal{T}$. Therefore, given some y, there is a one-to-one correspondence for every BCP history h_t with an RBCP history $\tilde{h}_t$ that begins with $y_0 = y$.

Next, observe that given a certain y, any pair of corresponding policies as defined in Equations (4.14) and (4.17) assign, at any time point, the same probability for a given action a, for every pair of such corresponding histories, h_t and $\tilde{h}_t$. In addition, recall that the dynamics of s_t is the same in BCP and RBCP, and in particular it is not affected by y_t. Therefore, for any initial state $s \in S_0$ and $y \in \mathbb{R}^K$, any pair of corresponding policies π and $\tilde{\pi}$ induce the same probability distribution on s_t and a_t, for any $t \in \mathcal{T}$:

$$\mathbb{P}^{\pi,s}(s_t, a_t) = \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(s_t, a_t).$$

We prove Equation (4.15) by observing, following the above discussion, that in particular, given an initial y, there exists a one-to-one correspondence for every burstiness-constrained BCP outcome $\omega \in B(y)$ with a burstiness-constrained RBCP outcome $\tilde{\omega} \in \tilde{B}$ at which $y_0 = y$.

Finally, consider the expected total reward objective, for example:

$$\begin{aligned}
\tilde{J}_N^{\tilde{\pi}}(s, y) &\triangleq \tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}\left[\sum_{t=0}^{N-1} r_t(s_t, a_t) + r_N(s_N)\right] \\
&= \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{N-1} r_t(s_t, a_t) + r_N(s_N)\right] \\
&\triangleq J_N^{\pi}(s),
\end{aligned}$$

thus proving Equation (4.16) for the expected total reward case. The proof for the expected discounted reward and expected long term average reward objectives is similar, relying on their form as sums of the expected immediate rewards.

■

With this relation between BCP and RBCP policies in mind, we can relate the solutions of the BCP and RBCP feasibility and optimization problems, as follows.

Lemma 4.4. *For any $s \in S_0$,*

$$\Phi(s) = \tilde{\Phi}(s, 0). \tag{4.18}$$

In addition, for any feasible initial BCP state, $s \in S_0^F$, and any objective type $o \in \{N, \gamma, ea\}$,

$$J_o^{F^*}(s) = \tilde{J}_o^{F^*}(s, 0), \tag{4.19}$$

and there exist optimizing policies $\pi_o^{F^*}$ and $\tilde{\pi}_o^{F^*}$ that are interrelated as described in Equations (4.14) *and* (4.17)*.*

Proof. Following Equation (4.15) of Lemma 4.3, and using the connection $B_{\sigma,\rho} = B(0)$, for any $s \in S_0$,

$$\exists \pi \in \Pi^{HR} : \ \mathbb{P}^{\pi,s}(B_{\rho,\sigma}) = 1 \quad \Leftrightarrow \quad \exists \tilde{\pi} \in \tilde{\Pi}^{HR} : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,0)}(\tilde{B}) = 1,$$

thus proving Equation (4.18).

Likewise, following Equations (4.15) and (4.16) of Lemma 4.3, given any $s \in S_0^F$ and $o \in \{N, \gamma, \mathrm{ea}\}$,

$$\begin{aligned} J_o^{F^*}(s) &\triangleq \max_{\pi \in \Pi^{HR}} \{J_o^{\pi}(s) : \ \mathbb{P}^{\pi,s}(B_{\rho,\sigma}) = 1\} \\ &= \max_{\tilde{\pi} \in \tilde{\Pi}^{HR}} \{\tilde{J}_o^{\tilde{\pi}}(s, 0) : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,0)}(\tilde{B}) = 1\} \\ &\triangleq \tilde{J}_o^{F^*}(s, 0), \end{aligned}$$

thus proving Equation (4.19).

■

Using the results of Lemma 4.4, we can solve **BCF** and **BCP** by solving **RBCF** and **RBCP**, respectively, for $y_0 = 0$.

4.4 Characterizing the RBCP

Several properties of the RBCP's feasibility and value functions can be obtained from its formulation. We first observe that the event $B(y)$ is monotone nonincreasing in y. Consequently, we show that $\tilde{\Phi}(s, y)$ has a threshold form in y, and that $\tilde{J}_o^{F^*}(s, y)$ is monotone nonincreasing in y. In addition, a simple condition for BCP feasibility can be derived. These results are shown in the following propositions.

Lemma 4.5. *The event $B(y)$, defined in Equation (4.10), is monotone nonincreasing in y, i.e., for any $y, y' \in \mathbb{R}^K$, if $y' \leq y$ then $B(y') \supseteq B(y)$.*

Proof. First, note that for any $x, y \in \mathbb{R}$, if $x \leq y$ then $x^+ \leq y^+$. For any $y, y' \in \mathbb{R}^K$, denote by $(y_t)_{t\in\mathcal{T}}$ the sequence generated by Equation (4.6) when $y_0 = y$, and by $(y'_t)_{t\in\mathcal{T}}$ the sequence generated by Equation (4.6) when $y'_0 = y'$. Observe that if $y' \leq y$ then $y'_t \leq y_t,\ \forall t \in \mathcal{T}$. Thus, given an initial state and policy, if some outcome $\omega \in \Omega$ belongs to $B(y)$, then it belongs to $B(y')$ as well: $\omega \in B(y) \Rightarrow \omega \in B(y')$. Therefore, $B(y) \subseteq B(y')$.

■

Lemma 4.6. *For any $s \in S_0$, let*

$$y^*(s) \triangleq \sup_{y\in\mathbb{R}^K} \{y :\ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\}. \tag{4.20}$$

Then,

a. $\tilde{\Phi}(s, y) = 1 \ \Leftrightarrow\ y \leq y^*(s), \quad \forall y \in Y_0(s)$.

b. $\tilde{J}_o^{F^*}(s, y)$ *is monotone nonincreasing in* y *(for any objective type* $o \in \{N, \gamma, ea\}$*).*

c. $\Phi(s) = 1 \ \Leftrightarrow\ y^*(s) \geq 0$.

Proof. To prove the right direction of item (a), recall that by definition of $y^*(s)$ in Equation (4.20), for any $(s, y) \in \tilde{S}_0$ such that $y \not\leq y^*(s)$, there does not exist $\tilde{\pi} \in \tilde{\Pi}^{HR}$ for which $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1$.

For the opposite direction, observe that by Equation (4.20), for any $s \in S_0$, there exists $\tilde{\pi} \in \tilde{\Pi}^{HR}$ such that $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y^*(s))}(\tilde{B}) = 1$. Using Equation (4.15) of Lemma 4.3 and Lemma 4.5, for any $s \in S_0$, $\tilde{\pi} \in \tilde{\Pi}^{HR}$ and $y, y' \in Y_0(s)$ such that $y' \leq y$, if $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1$ then $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y')}(\tilde{B}) = 1$. Therefore, for any $s \in S_0$ and $y \leq y^*(s)$ there exists $\tilde{\pi} \in \tilde{\Pi}^{HR}$ such that $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1$.

In addition, by Equation (4.15) of Lemma 4.3 and Lemma 4.5, the set of of feasible policies for $\tilde{B}$ with a certain initial y is monotone nonincreasing in y, i.e., for any $s \in S_0$ and $y, y' \in Y_0(s)$ such that $y' \leq y$,

$$\{\tilde{\pi} \in \tilde{\Pi}^{HR} :\ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\} \subseteq \{\tilde{\pi} \in \tilde{\Pi}^{HR} :\ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y')}(\tilde{B}) = 1\}.$$

Following item (a), for any $(s,y) \in \tilde{S}_0$ such that $y \leq y^*(s)$, there exists a feasible policy. Therefore, for any $s \in S_0$ and $y, y' \in Y_0(s)$ such that $y' \leq y \leq y^*(s)$,

$$\begin{aligned}
\tilde{J}_o^{F^*}(s,y) &\triangleq \max_{\tilde{\pi} \in \tilde{\Pi}^{HR}} \{\tilde{J}_o^{\tilde{\pi}}(s,y) : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\} \\
&\leq \max_{\tilde{\pi} \in \tilde{\Pi}^{HR}} \{\tilde{J}_o^{\tilde{\pi}}(s,y') : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y')}(\tilde{B}) = 1\} \\
&\triangleq \tilde{J}_o^{F^*}(s,y'),
\end{aligned}$$

where the first and third transitions are due to Lemma 4.3, thus proving item (b).

Finally, item (c) follows by applying item (a) above for $y = 0$ and using Equation (4.18) of Lemma 4.4.

■

The above properties will be used in the Finite and Infinite Horizon Algorithms chapters (Chapters 5 and 6, respectively) to determine the problem's feasibility, and to compute the problem's value and optimal policy.

4.5 Other Reformulations

The BCP reformulation introduced above uses an augmented state space and an extended burstiness constraint to express the BCP problem as a state-constrained MDP. Several other options for reformulation of the BCP as a state-constrained or unconstrained MDP are possible.

1. **Reward modification.** The reward function can be modified so as to express whether or not the burstiness constraint is adhered to at every time point, as follows:

$$\tilde{r}_t(s,y,a) = \begin{cases} r_t(s,a), & \text{if } y + d_t(s,a) - \rho \leq \sigma; \\ \phi, & \text{otherwise,} \end{cases}$$

where $t \in \mathcal{T},\ (s,y) \in \tilde{S}_t,\ a \in A_t(s)$, and ϕ is a non-numerical value with the following arithemetic rules:

$$\phi + x = \phi \quad \text{and} \quad \phi < x, \quad \forall x \in \mathbb{R}^K.$$

The objective functions in this formulations are created by substituting all of the terms in the BCP's objective function with their counterparts in the reformulation. For example, the reformulated total expected reward would be $\tilde{J}_N^{\tilde{\pi}}(s,y) =$

$\tilde{\mathbb{E}}^{\tilde{\pi},(s,y)}\left[\sum_{t\in\mathcal{T}}\tilde{r}_t(s_t,y_t,a_t)\right]$. Due to ϕ's arithmetic rules, $\tilde{J}_o^{\tilde{\pi}}(s,y)\neq\phi$ if and only if $\tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(B(y))=1$. Thus, the BCP feasibility and optimization problems can be combined into a single unconstrained optimization problem of maximizing $\tilde{J}_o^{\tilde{\pi}}(s,y)$, to which we can apply to it the standard MDP algorithms. We can also use $-\infty$ instead of the non-numerical value ϕ, if the reward function is known to be bounded.

2. **Constraint violation absorbing state.** This formulation adds an additional state, s_t^*, to the state-space at every time step. Reaching this state will denote that the burstiness constraint has been violated. We can then express the burstiness constraint by $s_t\neq s_t^*,\ \forall t\in\mathcal{T}$. The resulting state constraint can then be used in the modified reward scheme of item 1 above.

3. **Constraint violation state variable.** In this formulation, we further augment the state variable with a binary argument which denotes whether or not the constraint has been violated thus far. Its dynamics is deterministic,

$$z_{t+1}=\begin{cases}1, & \text{if } z_t=1 \text{ and } y_t+d_t(s_t,a_t)-\rho\leq\sigma;\\ 0, & \text{otherwise,}\end{cases}$$

where $z_0\equiv 1$. The resulting state constraint is then $z_t=1,\ \forall t\in\mathcal{T}$. This constraint formulation can then be used in the modified reward scheme of item 1 above.

Chapter 5

Finite Horizon Algorithms

The state-constrained formulation of the RBCP facilitates simple methods to determine its feasibility, and enables application of standard MDP tools to find the optimal RBCP values and policies. Using the connection between BCPs and RBCPs, we can then solve **BCF** and **BCP**. In this chapter we manifest these algorithms and characterize their solutions and complexity, for the finite horizon case.

In finite horizon RBCPs, the feasibility determination and objective optimization tasks are performed by backwards-induction type algorithms. We find the feasibility threshold fucntion, $y^*(s)$ (defined in Equation (4.20)), and solve **BCF**, by calculating backwards for every time point the maximum allowed deficit, y_t. We can then apply the standard MDP backward induction algorithm to the RBCP and solve **BCP**. Moreover, we can check which states are feasible and find the allowed actions at each time point while optimizing the objective function.

5.1 Feasibility Determination

5.1.1 Feasibility Threshold Function Equations

In order to find the feasible states in finite horizon RBCPs, we calculate backwards the maximum allowed deficit, y_t, at every time point, as seen in the next lemma. Towards this end, we make the following notations:

For any $t \in \mathcal{T}$ and $y \in \mathbb{R}^K$, $\tilde{B}_t(y)$ is the event of adherence to the extended burstiness constraint from time t onwards, when $y_t = y$, namely,

$$\tilde{B}_t(y) \triangleq \left\{\tilde{\omega} \in \tilde{\Omega} : \quad y_t = y; \quad y_{t'} + d_{t'}(s_{t'}, a_{t'}) - \rho \leq \sigma, \quad \forall t' \in \{t, ..., N\}\right\}.$$

For any $t \in \mathcal{T}$ and $s \in S_t$, $y_t^*(s)$ is the maximum deficit allowed at time t when the system is at state $s_t = s$, so that the extended burstiness constraint could be adhered to from this time point onward, namely,

$$y_t^*(s) \triangleq \sup_{y \in \mathbb{R}^K} \left\{ y : \ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1 \right\}.$$

Lemma 5.1. *Given a finite horizon RBCP,*

a. $y^*(s) = y_0^*(s), \ \forall s \in S_0$.

b. For any $t \in \mathcal{T}, \ s \in S_t$ *and* $y \in \mathbb{R}^K$,

$$\exists \tilde{\pi} \in \tilde{\Pi}^{HR} \ s.t. \ \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1 \ \Leftrightarrow \ y \leq y_t^*(s).$$

c. For any $s \in S_N$,

$$y_N^*(s) = \sigma - d_N(s) + \rho.$$

Also, for any $t \in \{0, ..., N-1\}, \ s \in S_t$, *and* $a \in A_t(s)$, *let*

$$f_t(s,a) \triangleq \min_{\substack{s' \in S_{t+1}: \\ p_t(s'|s,a) > 0}} \{y_{t+1}^*(s')\},$$

$$y_t(s,a) \triangleq \begin{cases} \min\{\sigma, f_t(s,a)\} - d_t(s,a) + \rho, & \text{if } f_t(s,a) \geq 0; \\ -\infty, & \text{otherwise.} \end{cases}$$

Then, for any $t \in \{0, ..., N-1\}$ *and* $s \in S_t$,

$$y_t^*(s) = \max_{a \in A_t(s)} y_t(s,a).$$

Proof. Item (a) follows trivially by the definition of $y^*(s)$ and observing that for any policy $\tilde{\pi}$ and initial state $(s,y) \in \tilde{S}_0$,

$$\tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_0(y) | s_0 = s) \equiv \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}).$$

The proof of item (b) is similar to that of Lemma 4.6(a). The right direction of item (c) follows from the definition of $y_t^*(s)$ as the maximal y for which a feasible policy exists for $\tilde{B}_t(y)$, given $s_t = s$. For the opposite direction, first observe that the event $\tilde{B}_t(y)$ is monotone nonincreasing in y for any $t \in \mathcal{T}$, using the same reasoning as in Lemma 4.5. Thus, for any $t \in \mathcal{T}, \ s \in S_t$ and $y \leq y_t^*(s)$, $\tilde{B}_t(y) \supseteq \tilde{B}_t(y_t^*(s))$. By

definition of $y_t^*(s)$, for any $t \in \mathcal{T}$ and $s \in S_t$ there exists a policy $\tilde{\pi} \in \tilde{\Pi}^{HR}$ for which $\tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y_t^*(s)) \mid s_t = s) = 1$. Therefore, for any $y \leq y_t^*(s)$, there also exists a policy $\tilde{\pi} \in \tilde{\Pi}^{HR}$ for which $\tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1$.

To prove item (c), first observe that by definition, for any $s \in S_N$, the maximum allowed deficit at time $t = N$ is

$$\begin{aligned} y_N^*(s) &\triangleq \sup_{y \in \mathbb{R}^K} \{y : \ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi}}(y + d_N(s_N) - \rho \leq \sigma \mid s_N = s) = 1\} \\ &= \sup_{y \in \mathbb{R}^K} \{y : \ y + d_N(s) - \rho \leq \sigma\} \\ &= \sigma - d_N(s) + \rho. \end{aligned}$$

Next, let $t \in \{0, ..., N-1\}$, $s \in S_t$ and $y \in \mathbb{R}^K$. Observe the following:

$$\begin{aligned} &\exists \tilde{\pi} \in \tilde{\Pi}^{HR} : \ \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1 \\ \Leftrightarrow \exists a \in A_t(s) \text{ s.t. } &\begin{cases} y + d_t(s,a) - \rho \leq \sigma \text{ and} \\ \forall (s', y') \in \tilde{S}_{t+1} \text{ s.t. } \tilde{p}_t((s', y')|(s, y), a) > 0, \\ \qquad \exists \tilde{\pi}' \in \tilde{\Pi}^{HR} : \ \tilde{\mathbb{P}}^{\tilde{\pi}'}(\tilde{B}_{t+1}(y') \mid s_{t+1} = s') = 1. \end{cases} \end{aligned} \tag{5.1}$$

Namely, existence of a policy under which $\tilde{B}_t(y)$ holds w.p. 1 when $s_t = s$, means that there exists an action a which satisfies one step of the burstiness constraint $(y + d_t(s,a) - \rho \leq \sigma)$, and that for any possible state transition (s', y'), there exists a policy $\tilde{\pi}'$ under which $\tilde{B}_{t+1}(y')$ holds w.p. 1 when $s_{t+1} = s'$.

Then recall that according to the RBCP state dynamics, for any $a \in A_t(s)$, and $(s', y') \in \tilde{S}_{t+1}$:

$$\tilde{p}_t\left((s', y')|(s, y), a\right) > 0 \quad \Leftrightarrow \quad p_t(s'|s, a) > 0 \ \text{ and } \ y' = (y + d_t(s,a) - \rho)^+.$$

Using this statement and item (b) above, if at time $t \in \{0, ..., N-1\}$ and state $(s, y) \in \tilde{S}_t$, action $a \in A_t(s)$ was performed, then existence of a feasible policy for any possible state transition $(s', y') \in \tilde{S}_{t+1}$ is equivalent to the following statements:

$$\begin{aligned} &\forall (s', y') \in \tilde{S}_{t+1} \ \text{ s.t. } \ \tilde{p}_t\left((s', y')|(s, y), a\right) > 0, \\ &\qquad \exists \tilde{\pi}' \in \tilde{\Pi}^{HR} : \ \tilde{\mathbb{P}}^{\tilde{\pi}'}\left(\tilde{B}_{t+1}(y') \mid s_{t+1} = s'\right) = 1 \\ \Leftrightarrow &\ \forall s' \in S_{t+1} \ \text{ s.t. } \ p_t(s'|s, a) > 0, \ \ (y + d_t(s,a) - \rho)^+ \leq y_{t+1}^*(s') \\ \Leftrightarrow &\ (y + d_t(s,a) - \rho)^+ \leq f_t(s,a). \end{aligned} \tag{5.2}$$

In addition, note that for any $a \in A_t(s)$,

$$\begin{cases} y + d_t(s,a) - \rho \leq \sigma \text{ and} \\ (y + d_t(s,a) - \rho)^+ \leq f_t(s,a) \end{cases} \quad \Leftrightarrow \quad y \leq y_t(s,a). \tag{5.3}$$

Finally, observe that

$$\exists a \in A_t(s) \text{ s.t. } y \leq y_t(s,a) \quad \Leftrightarrow \quad y \leq \max_{a \in A_t(s)} y_t(s,a). \tag{5.4}$$

Plugging Equations (5.2), (5.3), and (5.4) into Equation (5.1) and using again item (b) above, for any $t \in \{0, ..., N-1\}$ and $s \in S_t$,

$$y_t^*(s) = \max_{a \in A_t(s)} y_t(s,a).$$

■

5.1.2 Feasibility Determination Algorithm

Following Lemmas 4.4 and 5.1, we have the next backward-induction type algorithm to compute $y^*(s)$ and solve **BCF** in the finite horizon case. In fact, in order to solve **BCP**, we will need the entire set of functions $\{y_t^*(s)\}_{t \in \mathcal{T}}$, which we compute here as well.

Algorithm 1: Finite Horizon BCPs - Feasibility Computation.

1. For $s \in S_N$, set $y_N^*(s) = \sigma - d_N(s) + \rho$.
   If for all $s \in S_N$, $y_N^*(s) \ngeq 0$, then no feasible BCP policy exists;
   For any $t \in \{0, ..., N-1\}$ and $s \in S_t$, set $y_t^*(s) = -\infty$;
   For any $s \in S_0$, set $\Phi(s) = 0$;
   Set $S_0^F = \emptyset$ and return.

2. For $t = N-1, ..., 0$,
   For $s \in S_t$,
   For $a \in A_t(s)$,
   Set $f_t(s,a) = \min_{\substack{s' \in S_{t+1}: \\ p_t(s'|s,a) > 0}} \{y_{t+1}^*(s')\}$;
   Set $y_t(s,a) = \begin{cases} \min\{\sigma, f_t(s,a)\} - d_t(s,a) + \rho, & \text{if } f_t(s,a) \geq 0; \\ -\infty, & \text{otherwise.} \end{cases}$
   Set $y_t^*(s) = \max_{a \in A_t(s)} y_t(s,a)$.

If for all $s \in S_t,\ y_t^*(s) \not\geq 0$, then no feasible policy exists;
For any $t' \in \{0, ..., t\}$ and $s \in S_{t'}$, set $y_{t'}^*(s) = -\infty$;
For any $s \in S_0$, set $y^*(s) = -\infty$ and $\Phi(s) = 0$;
Set $S_0^F = \emptyset$ and return.

3. For any $s \in S_0$, the RBCP's feasibility threshold function is $y^*(s) = y_0^*(s)$, and the BCP's feasibility indicator function is $\Phi(s) = \begin{cases} 1, & \text{if } y^*(s) \geq 0; \\ 0, & \text{otherwise.} \end{cases}$
The set of feasible initial BCP states is $S_0^F = \{s \in S_0 :\ y^*(s) \geq 0\}$.

Note that the complexity of this algorithm is of the order of the total number of state-action pairs of the underlying BCP, namely $\mathcal{O}\left(N_{\mathcal{S}\times\mathcal{A}}\right)$, where

$$N_{\mathcal{S}\times\mathcal{A}} \triangleq \sum_{t=0}^{N-1} \Big| \{(s,a) :\ s \in S_t,\ a \in A_t(s)\} \Big| + \Big| S_N \Big|.$$

This is equal to the complexity of the backward induction algorithm when performed on the BCP, without state augmentation and disregarding the burstiness constraint.

5.2 Objective Function Optimization

5.2.1 RBCP Objective Function Equations

Towards calculating the RBCP's maximum total expected reward under burstiness constraints, we need to find the RBCP's feasible states and actions at every time point. We make the following definitions:

For any $t \in \mathcal{T}$, define the set of feasible RBCP states at time t:

$$\tilde{S}_t^F \triangleq \left\{ (s,y) \in \tilde{S}_t :\ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1 \right\}.$$

Also, for any $t \in \{0, ..., N-1\}$ and $(s,y) \in \tilde{S}_t^F$, denote the set of feasible RBCP actions at time t:

$$A_t^F(s,y) \triangleq \{a \in A_t(s) :\ y + d_t(s,a) - \rho \leq \sigma \text{ and } \forall s' \in S_{t+1} \text{ s.t. } p_t(s'|s,a) > 0,\ (y + d_t(s,a) - \rho)^+ \leq y_{t+1}^*(s')\}.$$

Denote by $\tilde{\Pi}^{MD,F} \subseteq \tilde{\Pi}^{MD}$ the set of Markov deterministic policies over the RBCP, which assign only actions from $A_t^F(s,y)$ to states $(s,y) \in \tilde{S}_t^F$, such that for any policy

$\tilde{\pi} \in \tilde{\Pi}^{MD,F}$, time point $t \in \{0, ..., N-1\}$, and state $(s, y) \in \tilde{S}_t$, if $(s, y) \in \tilde{S}_t^F$ then $\tilde{\pi}_t(s, y) \in A_t^F(s, y)$.

For any $t \in \mathcal{T},\ (s, y) \in \tilde{S}_t$ and $\tilde{\pi} \in \tilde{\Pi}^{HR}$, let

$$\tilde{v}_t^{\tilde{\pi}}(s, y) \triangleq \tilde{\mathbb{E}}^{\tilde{\pi}} \left[\sum_{t'=t}^{N-1} r_{t'}(s_{t'}, a_{t'}) + r_N(s_N) \mid s_t = s \right],$$

and

$$\tilde{v}_t^{F^*}(s, y) \triangleq \max_{\tilde{\pi} \in \tilde{\Pi}^{MD,F}} \left\{ \tilde{v}_t^{\tilde{\pi}}(s, y) \right\}.$$

The following lemma first states that if an RBCP state $(s, y) \in \tilde{S}_t$ is feasible, then the set of actions $A_t^F(s, y)$ is nonempty (i.e., there exist feasible actions for the given state). The lemma then gives conditions for an action $a \in A_t(s)$ to be feasible (i.e., $a \in A_t^F(s, y)$). We then give conditions for feasibility of RBCP Markov deterministic policies. Finally, we connect between the RBCP optimization problem and the above equations.

Lemma 5.2. *Given a finite horizon RBCP with a total expected reward objective,*

a. For any $t \in \{0, ..., N-1\}$ and $(s, y) \in \tilde{S}_t$,

$$(s, y) \in \tilde{S}_t^F \ \Leftrightarrow \ A_t^F(s, y) \neq \emptyset.$$

b. For any $t \in \{0, ..., N-1\},\ (s, y) \in \tilde{S}_t^F$ and $a \in A_t(s)$,

$$a \in A_t^F(s, y) \ \Leftrightarrow \ \begin{cases} y + d_t(s, a) - \rho \leq \sigma \ \text{and} \\ \forall (s', y') \in \tilde{S}_{t+1} \ \text{s.t.} \ \tilde{p}_t((s', y')|(s, y), a) > 0,\ (s', y') \in \tilde{S}_{t+1}^F. \end{cases}$$

c. For any $\tilde{\pi} \in \tilde{\Pi}^{MD}$,

$$\tilde{\pi} \in \tilde{\Pi}^{MD,F} \ \Leftrightarrow \ \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1,\ \forall t \in \mathcal{T},\ (s, y) \in \tilde{S}_t^F.$$

d. $\tilde{J}_N^{F^*}(s, y) = \tilde{v}_0^{F^*}(s, y),\ \forall (s, y) \in \tilde{S}_0^F$.

Proof. To prove item (a), observe the following equivalences, using Lemmas 5.1(b) and

5.1(c), for any $t \in \{0, ..., N-1\}$ and $(s, y) \in \tilde{S}_t$:

$$\begin{aligned}
(s,y) \in \tilde{S}_t^F &\Leftrightarrow \exists \tilde{\pi} \in \tilde{\Pi}^{HR} : \ \tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1 \\
&\Leftrightarrow y \leq y_t^*(s) \\
&\Leftrightarrow \exists a \in A_t(s) \text{ s.t. } \begin{cases} y + d_t(s,a) - \rho \leq \sigma \text{ and} \\ \forall s' \in S_{t+1} \text{ s.t. } p_t(s'|s,a) > 0, \ (y + d_t(s,a) - \rho)^+ \leq y_{t+1}^*(s') \end{cases} \\
&\Leftrightarrow A_t^F(s,y) \neq \emptyset.
\end{aligned}$$

Using Lemma 5.1(b) and the RBCP state dynamics, observe that for any $t \in \{0, ..., N-1\}$, $(s, y) \in \tilde{S}_t^F$ and $a \in A_t(s)$,

$$\begin{aligned}
a \in A_t^F(s,y) &\Leftrightarrow \begin{cases} y + d_t(s,a) - \rho \leq \sigma \text{ and} \\ \forall s' \in S_{t+1} \text{ s.t. } p_t(s'|s,a) > 0, \ (y + d_t(s,a) - \rho)^+ \leq y_{t+1}^*(s') \end{cases} \\
&\Leftrightarrow \begin{cases} y + d_t(s,a) - \rho \leq \sigma \text{ and} \\ \forall (s',y') \in \tilde{S}_{t+1} \text{ s.t. } \tilde{p}_t((s',y')|(s,y),a) > 0, \ (s',y') \in \tilde{S}_{t+1}^F, \end{cases}
\end{aligned}$$

thus proving item (b).

For proof of item (c), observe the following equivalences, using item (b) above, for any $\tilde{\pi} \in \tilde{\Pi}^{MD}$ and any $t \in \{0, ..., N-1\}$, $(s, y) \in \tilde{S}_t^F$:

$$\begin{aligned}
&\tilde{\pi} \in \tilde{\Pi}^{MD,F} \\
\Leftrightarrow\ &\tilde{\pi}_t(s,y) \in A_t^F(s,y) \\
\Leftrightarrow\ &\begin{cases} y + d_t(s, \tilde{\pi}_t(s,y)) - \rho \leq \sigma \text{ and} \\ \forall (s',y') \in \tilde{S}_{t+1} \text{ s.t. } \tilde{p}_t((s',y')|(s,y), \tilde{\pi}_t(s,y)) > 0, \quad (s',y') \in \tilde{S}_{t+1}^F \end{cases} \\
\Leftrightarrow\ &\tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_t(y) \mid s_t = s) = 1.
\end{aligned}$$

To prove item (d), observe that for any $(s, y) \in \tilde{S}_0^F$ and $\tilde{\pi} \in \tilde{\Pi}^{HR}$,

$$\tilde{v}_0^{\tilde{\pi}}(s,y) \equiv \tilde{J}_N^{\tilde{\pi}}(s,y),$$

and

$$\tilde{\mathbb{P}}^{\tilde{\pi}}(\tilde{B}_0(y)) \equiv \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}).$$

Then, using the dominance of Markov deterministic policies in the finite horizon

case and item (c) above, for any $(s,y) \in \tilde{S}_0^F$,

$$\begin{aligned}
\tilde{J}_N^{F^*}(s,y) &\triangleq \max_{\tilde{\pi}\in\tilde{\Pi}^{HR}} \{\tilde{J}_N^{\tilde{\pi}}(s,y) : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\} \\
&= \max_{\tilde{\pi}\in\tilde{\Pi}^{MD}} \{\tilde{J}_N^{\tilde{\pi}}(s,y) : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1\} \\
&= \max_{\tilde{\pi}\in\tilde{\Pi}^{MD,F}} \{\tilde{J}_N^{\tilde{\pi}}(s,y)\} \\
&\triangleq \tilde{v}_0^{F^*}(s,y).
\end{aligned}$$

■

Using Lemma 5.2(d), we can compute $\tilde{J}_N^{F^*}(s,y)$ with the following backwards-induction equations.

Lemma 5.3. *Given a finite horizon RBCP with a total expected reward objetcive, for any* $(s,y) \in \tilde{S}_N^F$,

$$\tilde{v}_N^{F^*}(s,y) = r_N(s).$$

Also, for any $t \in \{0,...,N-1\}$, $(s,y) \in \tilde{S}_t^F$ *and* $a \in A_t^F(s,y)$, *let*

$$\tilde{Q}_t(s,y,a) \triangleq r_t(s,a) \ + \sum_{\substack{s'\in S_{t+1}: \\ p_t(s'|s,a)>0}} p_t(s'|s,a) \cdot \tilde{v}_{t+1}^* \left(s', (y + d_t(s,a) - \rho)^+\right).$$

Then, for any $t \in \{0,...,N-1\}$ *and* $(s,y) \in \tilde{S}_t^F$,

$$\tilde{v}_t^{F^*}(s,y) = \max_{a\in A_t^F(s,y)} \tilde{Q}_t(s,y,a).$$

Proof. First observe that by definition, for any $(s,y) \in \tilde{S}_N^F$, $y + d_N(s) - \rho \leq \sigma$. Thus, for any $(s,y) \in \tilde{S}_N^F$,

$$\begin{aligned}
\tilde{v}_N^{F^*}(s,y) &= \max_{\tilde{\pi}\in\tilde{\Pi}^{MD,F}} \left\{\tilde{\mathbb{E}}^{\tilde{\pi}}\left[r_N(s)\right] \text{ s.t. } \tilde{\mathbb{P}}^{\tilde{\pi}}(y + d_N(s) - \rho \leq \sigma) = 1\right\} \\
&\equiv r_N(s).
\end{aligned}$$

Then, observe that for any $t \in \{0,...,N-1\}$, $(s,y) \in \tilde{S}_t$ and $\tilde{\pi} \in \tilde{\Pi}^{MD}$,

$$\tilde{v}_t^{\tilde{\pi}}(s,y) = r_t(s, \tilde{\pi}_t(s,y)) + \sum_{(s',y')\in\tilde{S}_{t+1}} \tilde{p}_t\big((s',y')\big|(s,y), \tilde{\pi}_t(s,y)\big) \cdot \tilde{v}_{t+1}^{\tilde{\pi}}(s',y').$$

Using Lemma 5.2(b), for any $t \in \{0, ..., N-1\}$ and $(s, y) \in \tilde{S}_t^F$,

$$\begin{aligned}
\tilde{v}_t^{F^*}(s,y) &= \max_{a \in A_t^F(s,y)} \left\{ r_t(s,a) + \sum_{(s',y') \in \tilde{S}_{t+1}} \tilde{p}_t\big((s',y')\big|(s,y),a\big) \cdot \max_{\tilde{\pi}' \in \tilde{\Pi}^{MD,F}} \tilde{v}_{t+1}^{\tilde{\pi}'}(s',y') \right\} \\
&= \max_{a \in A_t^F(s,y)} \left\{ r_t(s,a) + \sum_{\substack{(s',y') \in \tilde{S}_{t+1}: \\ \tilde{p}_t((s',y')|(s,y),a) > 0}} \tilde{p}_t\big((s',y')\big|(s,y),a\big) \cdot \tilde{v}_{t+1}^{F^*}(s',y') \right\} \\
&= \max_{a \in A_t^F(s,y)} \left\{ r_t(s,a) + \sum_{\substack{s' \in S_{t+1}: \\ p_t(s'|s,a) > 0}} p_t(s'|s,a) \cdot \tilde{v}_{t+1}^{F^*}\left(s', (y + d_t(s,a) - \rho)^+\right) \right\} \\
&\triangleq \tilde{Q}_t(s,y,a).
\end{aligned}$$

■

5.2.2 Selection of State-Sets

In order to implement the equations of Lemma 5.2 to solve **BCP**, we need to find finite and discrete sets of feasible states, $\{\tilde{S}_t^F\}_{t \in \mathcal{T}}$. Using Lemma 5.1(b), we have

$$\tilde{S}_t^F = \Big\{(s,y) \in \tilde{S}_t : \ y \le y_t^*(s)\Big\} = \Big\{(s,y) : \ s \in S_t, \ y \in Y_t(s), \ y \le y_t^*(s)\Big\}.$$

Recall from Equation (4.8) that the sets $Y_t(s)$ can be chosen arbitrarily so long as

$$\begin{aligned}
&Y_0(s) \supseteq \{0\}, \ \forall s \in S_0, \\
&Y_{t+1}(s') \supseteq \Big\{y' \in \mathbb{R}^K : \ y' = (y + d_t(s,a) - \rho)^+ \text{ where} \\
&\qquad\qquad s \in S_t, \ y \in Y_t(s), \ a \in A_t(s) \text{ and } p_t(s'|s,a) > 0\Big\}, \\
&\qquad\qquad\qquad\qquad \forall t \in \mathcal{T}, \ s' \in S_{t+1}.
\end{aligned}$$

We first observe that $Y_t(s)$ can be bounded: Since we are only interested in $y_0 = 0$ and by definition $y_t \ge 0, \ \forall t \in \{1, ..., N\}$, we can bound $Y_t(s) \subseteq \mathbb{R}_+^K$. Thus, we can constrain $\tilde{S}_t^F \subseteq \Big\{(s,y) : \ s \in S_t, \ y \in [0, y_t^*(s)]\Big\}$.

In the case where $d_t(\cdot,\cdot)$, ρ and σ are all integer multiples of some basic quantity $q \in \mathbb{R}^K$ (e.g., are natural numbers), so will be y_t, thus we may consider y to belong to the set of all integer multiples of q within $[0, y_t^*(s)]$. This is a simple, finite and discrete set, even if larger than the true set of possible y_t values. In the general case where this property cannot be established, we can alternatively use a suitable uniform discretization of the range $[0, y_t^*(s)]$, and quantize/interpolate values to/from this set.

5.2.3 Backward Induction Algorithm

With the equations of Lemma 5.3, we have the next algorithm to solve **BCP** in the finite horizon case with an expected total reward objective. Note that this algorithm uses the set of functions $\{y_t^*(s)\}_{t\in\mathcal{T}}$, which can be computed via Algorithm 1.

Algorithm 2: Finite Horizon BCPs with an Expected Total Reward Objective - Backwards Induction.

0. State sets preparation:
   Using $\{y_t^*(s)\}_{t\in\mathcal{T}}$, select suitable $\{\tilde{S}_t^F\}_{t\in\mathcal{T}}$ sets as described in Section 5.2.2.

1. For $(s,y)\in\tilde{S}_N^F$, set $\tilde{v}_N^{F^*}(s,y)=r_N(s)$.

2. For $t=N-1,...,0$,
   For $(s,y)\in\tilde{S}_t^F$,
   Set

$$A_t^F(s,y)=\Big\{a\in A_t(s):\ y+d_t(s,a)-\rho\le\sigma \text{ and } \forall s'\in S_{t+1} \text{ s.t. } p_t(s'|s,a)>0,\ (y+d_t(s,a)-\rho)^+\le y_{t+1}^*(s')\Big\}.$$

   For $a\in A_t^F(s,y)$, set

$$\tilde{Q}_t(s,y,a)=r_t(s,a)+\sum_{\substack{s'\in S_{t+1}:\\ p_t(s'|s,a)>0}} p_t(s'|s,a)\cdot\tilde{v}_{t+1}^{F^*}\left(s',(y+d_t(s,a)-\rho)^+\right).$$

   Set $\tilde{\pi}_t^{F^*}(s,y)\in\operatorname{argmax}_{a\in A_t^F(s,y)}\tilde{Q}_t(s,y,a)$;
   Set $\tilde{v}_t^{F^*}(s,y)=\max_{a\in A_t^F(s,y)}\tilde{Q}_t(s,y,a)$.

3. The RBCP policy $\tilde{\pi}_N^{F^*}=(\tilde{\pi}_t^{F^*})_{t=0}^{N-1}\in\tilde{\Pi}^{MD,F}$ optimizes **RBCP**, and $\tilde{J}_N^{F^*}(s,y)=\tilde{v}_0^{F^*}(s,y),\ \forall(s,y)\in\tilde{S}_0^F$ is the optimal **RBCP** value function.
   An optimal **BCP** policy is $\pi_N^{F^*}=(\pi_t^{F^*})_{t=0}^{N-1}\in\Pi^{HD}$, which is retrieved from $(\tilde{\pi}_t^{F^*})_{t=0}^{N-1}$ by

$$\pi_t^{F^*}(s_0,a_0,...,s_t)=\begin{cases}\tilde{\pi}_t^{F^*}(s_t,y_t), & \text{if } (s_t,y_t)\in\tilde{S}_t^F;\\ \text{arbitrarily chosen from } A_t(s_t), & \text{otherwise,}\end{cases}$$

   for any $t\in\{0,...,N-1\}$ and $(s_0,a_0,...,s_t)\in H_t$, where

$$y_0=0,\quad y_{t+1}=(y_t+d_t(s_t,a_t)-\rho)^+,\ \forall t\in\mathcal{T}.$$

   The optimal **BCP** value is $J_N^{F^*}(s)=\tilde{v}_0^{F^*}(s,0),\ \forall s\in S_0^F$.

Note that the optimal policy for the finite horizon BCP is Markov and deterministic in the augmented state-space. In the original state-space, the optimal policy is history-dependent and deterministic. Moreover, several history sequences might have probability 0 under the optimal policy, leading to obsolete entries in $\pi_N^{F^*}$. Obviously, it is more efficient to store the RBCP optimal policy and then calculate the deficit term, y_t, on the fly in order to determine the action to be performed at every time point.

The algorithm's complexity is $\mathcal{O}\left(N_{\tilde{\mathcal{S}}^F \times \mathcal{A}^F}\right)$, where $N_{\tilde{\mathcal{S}}^F \times \mathcal{A}^F}$ is the number of feasible RBCP state-action pairs:

$$N_{\tilde{\mathcal{S}}^F \times \mathcal{A}^F} = \sum_{t=0}^{N-1} \left| \left\{ (s, y, a) : \ (s, y) \in \tilde{S}_t^F, \ a \in A_t^F(s, y) \right\} \right| + \left| \tilde{S}_N^F \right|.$$

Thus, the complexity depends on the level of discretization of the deficit term, and on the number of feasible states and actions at every time point.

We note here that the above algorithm can use the fact that $\tilde{v}_t^{F^*}(s, y), \ A_t^F(s, y)$, $\tilde{\pi}_t^{F^*}(s, y)$ and $\tilde{Q}_t(s, y, a)$ are piecewise constant in y. The number of discontinuities in $\tilde{v}_t^{F^*}(s, y), \ \tilde{\pi}_t^{F^*}(s, y)$ and $A_t^F(s, y)$ is at most $\prod_{t'=t}^{N-1} \sum_{\substack{s \in S_{t'}: \\ y_{t'}^*(s) \geq 0}} |A_{t'}(s)|$, thus the number of constant-valued intervals in these functions is finite. Consequently, rather than storing the functions' values for any $y \in Y_t(s)$, we can simply store the values and boundaries of each such interval, and look up the required values when necessary.

5.3 Objective Function Optimization Without Feasibility Information

The optimal RBCP objective function can also be found without precomputing the feasibility threshold functions $\{y_t^*(s)\}_{t \in \mathcal{T}}$, by combining Algorithms 1 and 2. This is done by appropriately extending the RBCP's functions to nonfeasible states and actions, as shown herein.

5.3.1 Extended RBCP Objective Function Equations

Towards computing the RBCP's optimal value, we define ϕ, a nonnumerical value with the arithmetic rules

$$\phi \cdot 0 = 0, \quad \phi < x \ \text{ and } \ \phi + x = \phi, \quad \forall x \in \mathbb{R}.$$

We also define the following: For any $(s, y) \in \tilde{S}_N$, let

$$\tilde{v}_N^*(s, y) \triangleq \begin{cases} r_N(s), & \text{if } y + d_N(s) - \rho \leq \sigma; \\ \phi, & \text{otherwise.} \end{cases}$$

Also, for any $t \in \{0, ..., N-1\}$, $(s, y) \in \tilde{S}_t$ and $a \in A_t(s)$, let

$$\tilde{Q}_t(s, y, a) \triangleq \begin{cases} r_t(s, a) + \sum_{s' \in S_{t+1}} p_t(s'|s, a) \cdot \tilde{v}_{t+1}^* \left(s', (y + d_t(s, a) - \rho)^+\right), & \text{if } y + d_t(s, a) - \rho \leq \sigma; \\ \phi, & \text{otherwise,} \end{cases}$$

$$\tilde{v}_t^*(s, y) \triangleq \max_{a \in A_t(s)} \tilde{Q}_t(s, y, a),$$

$$\tilde{\pi}_t^*(s, y) \in \operatorname*{argmax}_{a \in A_t(s)} \tilde{Q}_t(s, y, a). \tag{5.5}$$

Lemma 5.4. *Given a finite horizon RBCP with a total expected reward objective,*

a. For any $t \in \mathcal{T}$ and $(s, y) \in \tilde{S}_t$,

$$\tilde{v}_t^*(s, y) = \begin{cases} \tilde{v}_t^{F^*}(s, y) & \text{if } (s, y) \in \tilde{S}_t^F; \\ \phi & \text{otherwise.} \end{cases} \tag{5.6}$$

b. For any $(s, y) \in \tilde{S}_0^F$, $\tilde{J}_N^{F^*}(s, y) = \tilde{v}_0^*(s, y)$.

Proof. We prove item (a) by induction. First, recall that by definition, for any $(s, y) \in \tilde{S}_N$,

$$y + d_N(s) - \rho \leq \sigma \;\Leftrightarrow\; (s, y) \in \tilde{S}_N^F.$$

Thus,

$$\tilde{v}_N^*(s, y) = \begin{cases} \tilde{v}_N^{F^*}(s, y), & \text{if } (s, y) \in \tilde{S}_N^F; \\ \phi, & \text{otherwise,} \end{cases}$$

therefore Equation (5.6) holds for $t = N$.

Next, for any $t \in \{0, ..., N-1\}$, assume that Equation (5.6) holds for $t+1$. Then, using the definition of $\tilde{S}_{t+1}^F$ and Lemma 5.1(b), the following statements are equivalent, for any $(s', y') \in \tilde{S}_{t+1}$:

$$\begin{aligned} \tilde{v}_{t+1}^*(s', y') \neq \phi &\Leftrightarrow (s', y') \in \tilde{S}_{t+1}^F \\ &\Leftrightarrow y' \leq y_{t+1}^*(s'). \end{aligned}$$

Therefore, using ϕ's arithmetic rules and the above equivalence, for any $(s, y) \in \tilde{S}_t$ and $a \in A_t(s)$,

$$\begin{aligned}
\tilde{Q}_t(s, y, a) \neq \phi \quad &\Leftrightarrow \quad \begin{cases} y + d_t(s, a) - \rho \leq \sigma \ \text{ and} \\ \tilde{v}^*_{t+1}(s', (y + d_t(s, a) - \rho)^+) \neq \phi, \ \forall s' \in S_{t+1} \text{ s.t. } p_t(s'|s, a) > 0. \end{cases} \\
&\Leftrightarrow \quad \begin{cases} y + d_t(s, a) - \rho \leq \sigma \ \text{ and} \\ (y + d_t(s, a) - \rho)^+) \leq y^*_{t+1}(s'), \ \forall s' \in S_{t+1} \text{ s.t. } p_t(s'|s, a) > 0. \end{cases} \\
&\Leftrightarrow \quad a \in A^F_t(s, y).
\end{aligned}$$

Therefore, for any $(s, y) \in \tilde{S}_t$ and $a \in A_t(s)$,

$$\tilde{Q}_t(s, y, a) = \begin{cases} r_t(s, a) + \sum\limits_{\substack{s' \in S_{t+1}: \\ p_t(s'|s,a) > 0}} p_t(s'|s, a) \cdot \tilde{v}^{F^*}_{t+1}\left(s', (y + d_t(s, a) - \rho)^+\right), & \text{if } a \in A^F_t(s, y); \\ \phi, & \text{otherwise.} \end{cases}$$

Then observe that for any $(s, y) \in \tilde{S}_t$,

$$\begin{aligned}
\tilde{v}^*_t(s, y) \neq \phi \ &\Leftrightarrow \ \exists a \in A_t(s) : \ \tilde{Q}_t(s, y, a) \neq \phi \\
&\Leftrightarrow \ A^F_t(s, y) \neq \emptyset \\
&\Leftrightarrow \ (s, y) \in \tilde{S}^F_t.
\end{aligned}$$

On the other hand, if for some $(s, y) \in \tilde{S}_t$, $\tilde{v}^*_t(s, y) \neq \phi$, then

$$\begin{aligned}
\tilde{v}^*_t(s, y) &= \max_{a \in A^F_t(s,y)} \left\{ r_t(s, a) + \sum_{\substack{s' \in S_{t+1}: \\ p_t(s'|s,a) > 0}} p_t(s'|s, a) \cdot \tilde{v}^{F^*}_{t+1}\left(s', (y + d_t(s, a) - \rho)^+\right) \right\} \\
&\triangleq \tilde{v}^{F^*}_t(s, y).
\end{aligned}$$

Thus, for any $(s, y) \in \tilde{S}_t$,

$$\tilde{v}^*_t(s, y) = \begin{cases} \tilde{v}^{F^*}_t(s, y), & \text{if } (s, y) \in \tilde{S}^F_t, \\ \phi, & \text{otherwise,} \end{cases}$$

which proves Equation (5.6) for any t.

Item (b) then follows by applying Equation (5.6) for $t = 0$.

■

5.3.2 Selection of State-Sets

In order to implement Equations 5.5 to solve **BCP**, we need to find finite and discrete state-sets, $\{\tilde{S}_t\}_{t\in\mathcal{T}}$. Recall that $\tilde{S}_t \triangleq \{(s,y):\ s\in S_t,\ y\in Y_t(s)\}$, and that by Equation (4.8), the sets $Y_t(s)$ can be chosen arbitrarily so long as

$$\begin{aligned} &Y_0(s) \supseteq \{0\},\ \forall s\in S_0,\\ &Y_{t+1}(s') \supseteq \Big\{y'\in\mathbb{R}^K:\ y' = (y + d_t(s,a) - \rho)^+ \text{ where}\\ &\qquad\qquad s\in S_t,\ y\in Y_t(s),\ a\in A_t(s) \text{ and } p_t(s'|s,a) > 0\Big\},\\ &\qquad\qquad\qquad\qquad \forall t\in\mathcal{T},\ s'\in S_{t+1}. \end{aligned}$$

In the general case, finding the minimal sets is a cumbersome task. Instead, we would like to find simple sets that satisfy the above. We first observe that these sets can be bounded. A lower bound is found in the same way as in Section 5.2.2: Since we are only interested in $y_0 = 0$ and by definition $y_t \geq 0,\ \forall t\in\{1,...,N\}$, we can bound $Y_t(s)\subseteq\mathbb{R}^K_+$. For an upper bound, we cannot use $y^*_t(s)$ (which limits us to feasible y's only). In stead, observe from Equations (5.5) that values of $\tilde{v}^*_t(s,y)$ for any $y\not\leqslant\sigma$ are never used: at each time point, in order to calculate $\tilde{v}^*_t(s,y)$, we only need values of $\tilde{v}^*_{t+1}(s',y')$ for $y'\leq\sigma$. This is because $y_{t+1}\not\leqslant\sigma$ means that $y_t + d_t(s_t,a_t) - \rho\not\leqslant\sigma$, thus the burstiness constraint is violated and we can immediately assign ϕ to the value function of the policy in question. Consequently, we can constrain $Y_t(s)\subseteq[0,\sigma]$.

As done in Section 5.2.2, in the case where $d_t(\cdot,\cdot),\ \rho$ and σ are all integer multiples of some basic quantity $q\in\mathbb{R}^K$ (e.g., are natural numbers), so will be y_t. Thus, we may consider $Y_t(s)$ to be the set of all integer multiples of q within $[0,\sigma]$, which is a simple, finite and discrete set, even if larger than the true set of possible y_t values. In the general case where this property cannot be established, we can alternatively use a suitable uniform discretization of the range $[0,\sigma]$.

5.3.3 Backward Induction Algorithm Without Precomputing Feasibility

Using Lemma 5.4, we obtain the following algorithm to solve **BCF** and **BCP** simultaneously for the finite horizon case with an expected total reward objective.

Algorithm 3: Finite Horizon BCPs with an Expected Total Reward Objective - Backwards Induction, Without Feasibility Precomputation.

0. State sets preparation:

   Select suitable $\{\tilde{S}_t\}_{t\in\mathcal{T}}$ sets as described in Section 5.3.2.

1. For $(s,y)\in\tilde{S}_N$, set $\tilde{v}_N^*(s,y)=\begin{cases} r_N(s), & \text{if } y+d_N(s)-\rho\leq\sigma; \\ \phi, & \text{otherwise.}\end{cases}$

2. For $t=N-1,...,0$,

   For $(s,y)\in\tilde{S}_t$,

   For $a\in A_t(s)$, set

$$\tilde{Q}_t(s,y,a)=\begin{cases} r_t(s,a)+\sum\limits_{s'\in S_{t+1}} p_t(s'|s,a)\cdot\tilde{v}_{t+1}^*(s',z^+), & \text{if } z\leq\sigma; \\ \phi, & \text{otherwise,}\end{cases}$$

   where $z=y+d_t(s,a)-\rho$.

   Set $\tilde{\pi}_t^*(s,y)\in\text{argmax}_{a\in A_t(s)}\,\tilde{Q}_t(s,y,a)$;

   Set $\tilde{v}_t^*(s,y)=\max_{a\in A_t(s)}\tilde{Q}_t(s,y,a)$.

3. For any $s\in S_0$, the BCP's feasibility indicator function is $\Phi(s)=\begin{cases}1, & \text{if } \tilde{v}_0^*(s,0)\neq\phi; \\ 0, & \text{otherwise.}\end{cases}$

   The set of feasible initial BCP states is $S_0^F=\{s\in S_0:\ \tilde{v}_0^*(s,0)\neq\phi\}$.

   An optimal **BCP** policy is $\pi_N^{F^*}=(\pi_t^{F^*})_{t=0}^{N-1}\in\Pi^{HD}$, which is retrieved from $\{\tilde{\pi}_t^*\}_{t=0}^{N-1}$ by

$$\pi_t^{F^*}(s_0,a_0,...,s_t)=\tilde{\pi}_t^*(s_t,y_t),$$

   for any $t\in\{0,...,N-1\}$ and $(s_0,a_0,...,s_t)\in H_t$, where

$$y_0=0,\quad y_{t+1}=(y_t+d_t(s_t,a_t)-\rho)^+,\ \forall t\in\mathcal{T}.$$

   The optimal **BCP** value is $J_N^{F^*}(s)=\tilde{v}_0^*(s,0),\ \forall s\in S_0^F$.

This algorithm's complexity is $\mathcal{O}\left(N_{\tilde{S}\times\tilde{A}}\right)$, where $N_{\tilde{S}\times\tilde{A}}$ is the total number of possible RBCP state-action pairs:

$$N_{\tilde{S}\times\tilde{A}}=\sum_{t=0}^{N-1}\left|\left\{(s,y,a):\ (s,y)\in\tilde{S}_t,\ a\in A_t(s)\right\}\right|+\left|\tilde{S}_N\right|.$$

This quantity depends on the level of discretization of the deficit term and on the total number of states and actions at every time point. This is bigger than the complexity of

Algorithm 2, since this algorithm goes through all states and actions, disregarding their feasibility, whereas Algorithm 2 considers only the feasible states and actions.

This algorithm too can use the fact that $\tilde{v}_t^*(s,y)$, $\tilde{\pi}_t^*(s,y)$ and $\tilde{Q}_t(s,y,a)$ are piecewise constant in y. As in Algorithm 2, the number of discontinuities in $\tilde{v}_t^*(s,y)$ and $\tilde{\pi}_t^*(s,y)$ is at most $\prod_{t'=t}^{N-1} \sum_{s \in S_{t'}} |A_{t'}(s)|$, thus the number of constant-valued intervals in these functions is finite. Consequently, rather than storing the functions' values for any $y \in Y_t(s)$, we can simply store the values and boundaries of each such interval, and look up the required values when necessary.

Chapter 6

Infinite Horizon Algorithms

Efficient algorithms for solving the RBCP feasibility and optimization problems can be found for the stationary infinite horizon case as well. We can then solve **BCF** and **BCP**, using the connection between BCPs and RBCPs. In this chapter we lay out the algorithms for the infinite horizon case, and characterize their solutions and complexity.

In infinite horizon RBCPs, we start with writing a fixed-point equation for the feasibility threshold function, $y^*(s)$ (defined in Equation (4.20)). In order to solve this equation, we characterize its solutions and find an algorithm to compute $y^*(s)$, thus solving **BCF**. We can then determine which states and policies of the RBCP are feasible, and apply the standard MDP results to the RBCP. For example, in case of an expected discounted reward, we can apply Bellman's equations (see Section 2.1.7.1) to the RBCP. Consequently, we write value iteration, policy iteration and linear programming algorithms to solve **BCP**.

6.1 Feasibility Determination

6.1.1 Feasibility Threshold Function Equations

In order to find the feasibility threshold function, $y^*(s)$, and solve **BCF** in the stationary infinite horizon case, we use a technique similar to the one used by Blackwell in [Bla67]. We first observe that $y^*(s)$ has a specific structure: as shown in the next lemma, $y^*(s)$ satifies $y^*(s) = \mathcal{F}\{y^*\}(s),\ \forall s \in S$, where $\mathcal{F}$ is the following *one-step feasibility* operator,

which operates on functions of the form $y(\cdot): S \to \mathbb{R}^K$:

$$\begin{aligned}
\mathcal{F}\{y\}(s) &\triangleq \max_{a \in A(s)} y(s,a), \quad \text{where} \\
y(s,a) &\triangleq \begin{cases} \min\{\sigma, f_y(s,a)\} - d(s,a) + \rho, & \text{if } f_y(s,a) \geq 0; \\ -\infty, & \text{otherwise,} \end{cases} \\
f_y(s,a) &\triangleq \min_{\substack{s' \in S: \\ p(s'|s,a)>0}} y(s'). \qquad (6.1)
\end{aligned}$$

However, $y^*(s)$ is not the only function to satisfy the above equation. Denote by $\mathcal{Y}$ the set of all fixed points of $\mathcal{F}$, i.e.,

$$\mathcal{Y} \triangleq \left\{ y(\cdot): S \to \mathbb{R}^K \ \text{ s.t. } \ y(s) = \mathcal{F}\{y\}(s), \ \forall s \in S \right\}. \qquad (6.2)$$

We observe that $y^*(s)$ is the maximal function in $\mathcal{Y}$, element-wise. In addition, we observe that the one-step feasibility operator defined above, $\mathcal{F}$, is monotone. These properties will help us find the feasibility threshold function, $y^*(s)$, and to solve **BCF**. These results are established in the following lemma.

Lemma 6.1. *Given an infinite horizon RBCP,*

a. $y^*(s)$ *is a fixed point of* $\mathcal{F}$.

b. Any function $y(\cdot) \in \mathcal{Y}$ *satisfies* $y(s) \leq y^*(s), \ \forall s \in S$.

c. For any two functions $x(\cdot), y(\cdot): S \to \mathbb{R}^K$, *if* $x(s) \leq y(s), \ \forall s \in S$, *then* $\mathcal{F}\{x\}(s) \leq \mathcal{F}\{y\}(s), \ \forall s \in S$.

Proof.

To prove item (a), recall that for any $(s,y) \in \tilde{S}$,

$$\begin{aligned}
\exists \tilde{\pi} \in \tilde{\Pi}^{HR}: \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1 \ &\Leftrightarrow \tilde{\Phi}(s,y) = 1 \\
&\Leftrightarrow y \leq y^*(s).
\end{aligned}$$

Also recall that according to the RBCP's state dynamics, for any $(s,y) \in \tilde{S}$, $a \in A(s)$, and $(s',y') \in \tilde{S}$,

$$\tilde{p}\left((s',y')|(s,y),a\right) > 0 \quad \Leftrightarrow \quad p(s'|s,a) > 0 \ \text{ and } \ y' = (y + d(s,a) - \rho)^+.$$

Then observe the following equivalences, for any $(s,y) \in \tilde{S}$, using Lemma 4.6(a):

$$
\begin{aligned}
& y \le y^*(s) \\
\Leftrightarrow\ & \tilde{\Phi}(s,y) = 1 \\
\Leftrightarrow\ & \exists \tilde{\pi} \in \tilde{\Pi}^{HR} :\ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1 \\
\Leftrightarrow\ & \exists a \in A(s) \text{ s.t. } \begin{cases} y + d(s,a) - \rho \le \sigma \text{ and} \\ \forall (s',y') \in \tilde{S} \text{ s.t. } \tilde{p}((s',y')|(s,y),a) > 0,\ \exists \tilde{\pi}' \in \tilde{\Pi}^{HR} :\ \tilde{\mathbb{P}}^{\tilde{\pi}',(s',y')}(\tilde{B}) = 1 \end{cases} \\
\Leftrightarrow\ & \exists a \in A(s) \text{ s.t. } \begin{cases} y + d(s,a) - \rho \le \sigma \text{ and} \\ \forall s' \in S \text{ s.t. } p(s'|s,a) > 0,\ (y + d(s,a) - \rho)^+ \le y^*(s') \end{cases}
\end{aligned} \tag{6.3}
$$

Then, let

$$
\begin{aligned}
f_{y^*}(s,a) &\triangleq \min_{\substack{s' \in S: \\ p(s'|s,a) > 0}} y^*(s'), \\
y^*(s,a) &\triangleq \begin{cases} \min\{\sigma, f_{y^*}(s,a)\} - d(s,a) + \rho, & \text{if } f_{y^*}(s,a) \ge 0; \\ -\infty, & \text{otherwise.} \end{cases}
\end{aligned}
$$

Following Equation (6.3) above, observe the following equivalences, for any $(s,y) \in \tilde{S}$:

$$
\begin{aligned}
& y \le y^*(s) \\
\Leftrightarrow\ & \exists a \in A(s) :\ y + d(s,a) - \rho \text{ and } (y + d(s,a) - \rho)^+ \le f_{y^*}(s,a) \\
\Leftrightarrow\ & \exists a \in A(s) :\ y \le y^*(s,a) \\
\Leftrightarrow\ & y \le \max_{a \in A(s)} y^*(s,a).
\end{aligned}
$$

Therefore, $y^*(s) = \max_{a \in A(s)} y^*(s,a) \equiv \mathcal{F}\{y^*\}(s),\ \forall s \in S$.

For proof of item (b), let $y(\cdot) \in \mathcal{Y}$. For any state $s \in S$ at which $y(s) = -\infty$, trivially $y(s) \le y^*(s)$. For any state $s \in S$ at which $y(s) \ne -\infty$, and for any $y \le y(s)$, there exists an action $a \in A(s)$ for which $y \le y(s,a) \ne -\infty$. Therefore, for such (s,y) pairs,

$$
\exists a \in A(s) :\ \begin{cases} y \le \sigma - d(s,a) + \rho & \text{and} \\ y \le f_y(s,a) - d(s,a) + \rho & \text{and} \\ f_y(s,a) \ge 0, \end{cases}
$$

or equivalently,

$$\exists a \in A(s): \begin{cases} y + d(s,a) - \rho \leq \sigma \quad \text{and} \\ \forall s' \in S: \ p(s'|s,a) > 0, \quad (y + d(s,a) - \rho)^+ \leq y(s'). \end{cases}$$

Telescopically expanding this statement yields that

$$\exists a \in A(s): \begin{cases} y + d(s,a) - \rho \leq \sigma \quad \text{and} \\ \forall s' \in S: \ p(s'|s,a) > 0, \quad \exists a' \in A(s'): \\ \qquad \begin{cases} (y + d(s,a) - \rho)^+ + d(s',a') - \rho \leq \sigma \quad \text{and} \\ \forall s'' \in S: \ p(s''|s',a') > 0, \quad \big((y + d(s,a) - \rho)^+ + d(s',a') - \rho\big)^+ \leq y(s''). \end{cases} \end{cases}$$

Further expanding the above expression infinitely many time points ahead would yield that for any such (s, y) pair there exists a *policy* for which the extended burstiness constraint, $B(y)$, is obeyed with probability 1. Hence, these (s, y) pairs are feasible initial states for the RBCP. Since this is true for any $y \leq y(s)$, and by definition of $y^*(s)$ as the maximum feasible initial deficit (y) for state s, we conclude that $y(s) \leq y^*(s)$ also when $y(s) \neq -\infty$, thus, indeed $y^*(\cdot) \in \mathcal{Y}$.

Consequently, $y^*(s) = \max_{y \in \mathcal{Y}} y(s), \ \forall s \in S$.

For proof of (c), let $x(\cdot), y(\cdot) : S \to \mathbb{R}^K$ such that $x(s) \leq y(s), \ \forall s \in S$. Then,

$$f_x(s,a) = \min_{\substack{s' \in S: \\ p(s'|s,a) > 0}} x(s') \ \leq \ \min_{\substack{s' \in S: \\ p(s'|s,a) > 0}} y(s') \ = \ f_y(s,a).$$

Therefore, $f_x(s,a) \geq 0 \ \Rightarrow \ f_y(s,a) \geq 0$. Consequently,

$$\begin{aligned} x(s,a) &= \begin{cases} \min\{f_x(s,a), \sigma\} - d(s,a) + \rho, & \text{if } f_x(s,a) \geq 0; \\ -\infty, & \text{otherwise,} \end{cases} \\ &\leq \begin{cases} \min\{f_y(s,a), \sigma\} - d(s,a) + \rho, & \text{if } f_y(s,a) \geq 0; \\ -\infty, & \text{otherwise,} \end{cases} \\ &= y(s,a); \end{aligned}$$

and finally

$$\mathcal{F}\{x\}(s) = \max_{a \in A(s)} x(s,a) \ \leq \ \max_{a \in A(s)} y(s,a) \ = \ \mathcal{F}\{y\}(s).$$

■

The results of Lemma 6.1 aren't enough for finding $y^*(s)$: we still need to find the highest-valued function in $\mathcal{Y}$. In the following lemma, we construct this function iteratively. We begin with an upper bound for $y^*(s)$, and make it tighter as we progress. Eventually, we receive a function which belongs to $\mathcal{Y}$, thus it must be $y^*(s)$. The proof is limited to the case where $d(\cdot,\cdot)$, ρ and σ are all integer multiples of some basic quantity $q \in \mathbb{R}^K$.

Lemma 6.2. *Assume an infinite horizon RBCP for which $d(\cdot,\cdot)$, ρ and σ are all integer multiples of some basic quantity $q \in \mathbb{R}^K$. For any $s \in S,\ n \in \mathbb{N}$, let*

$$y_1(s) \triangleq \max_{a \in A(s)} \{\sigma - d(s,a) + \rho\},$$
$$y_{n+1}(s) \triangleq \mathcal{F}\{y_n\}(s).$$

Then,

a. *For any $s \in S$, the sequence $\{y_n(s)\}_{n\in\mathbb{N}}$ is monotone nonincreasing in n.*

b. *There exists $T \in \mathbb{N}$ for which $y_T(s) = y_{T+1}(s),\ \forall s \in S$.*

c. *For any $n \in \mathbb{N}$ and $s \in S$, $y_n(s) \geq y^*(s)$.*

d. *For any $s \in S$, $y_T(s) = y^*(s)$.*

Proof. The proof of item (a) is made by induction. First, observe that by definition, $y_n(s,a) \leq \sigma - d(s,a) + \rho$ for any $n \in \mathbb{N},\ s \in S$ and $a \in A(s)$. Therefore, for any $s \in S$,

$$\begin{aligned} y_2(s) &\triangleq \max_{a \in A(s)} y_2(s,a) \\ &\leq \max_{a \in A(s)} \{\sigma - d(s,a) + \rho\} \\ &\triangleq y_1(s). \end{aligned}$$

Next, for any $n \in \mathbb{N}$, assume $y_{n+1}(s) \leq y_n(s),\ \forall s \in S$. Then, by Lemma 6.1(c), for any $s \in S$,

$$y_{n+2}(s) \triangleq \mathcal{F}\{y_{n+1}\}(s) \leq \mathcal{F}\{y_n\}(s) \triangleq y_{n+1}(s).$$

Before proving item (b), we make the following notation: given two K-dimensional vectors, $x, y \in \mathbb{R}^K$, the expression $x \leq^s y$ denotes that there is at least one element $i \in \{1,...,K\}$ for which the inequality holds strictly, such that $x_i < y_i$.

We also denote the following set of actions, for any $n \in \mathbb{N}$ and $s \in S$:

$$A^{(n)}(s) \triangleq \operatorname*{argmax}_{a \in A(s)} y_n(s,a).$$

Observe the equivalence of the following statements, for any $n \in \mathbb{N}$ and $s \in S$:

$$\begin{aligned}
& y_n(s) \neq y_{n+1}(s) \\
\Leftrightarrow\ & y_n(s) \geq^s y_{n+1}(s) \\
\Leftrightarrow\ & \forall a \in A^{(n)}(s),\ y_n(s,a) \geq^s y_{n+1}(s,a) \\
\Leftrightarrow\ & \forall a \in A^{(n)}(s),\ f_{y_n}(s,a) \geq^s f_{y_{n+1}}(s,a), \qquad (6.4)
\end{aligned}$$

where the first equivalence follows from item (a) above, and the latter two equivalences follow by definition of $y_n(s,a)$ and $f_{y_n}(s,a)$.

We prove item (b) by contradiction. If there does not exist a time point $T \in \mathbb{N}$ at which $y_T(s) = y_{T+1}(s),\ \forall s \in S$, then for any $n \in \mathbb{N}$ there exists some $s \in S$ at which $y_n(s) \neq y_{n+1}(s)$. Since the number of possible states is finite, this means that there exists at least one state $s \in S$ for which for any time point $n \in \mathbb{N}$, there exists a later time point, $n' \geq n$, at which $y^*_{n'}(s) \neq y^*_{n'+1}(s)$. Using Equation (6.4), this means that for some $s \in S$, $\forall n \in \mathbb{N}\ \exists n' \geq n:\ \forall a \in A^{(n')}(s),\ f_{y_{n'}}(s,a) \geq^s f_{y_{n'+1}}(s,a)$.

Recall that the lemma assumes that $d(\cdot,\cdot)$, ρ and σ are all integer multiples of some basic quantity $q \in \mathbb{R}^K$ (e.g., are natural numbers). Thus, any decrease in $f_{y_n}(s,a)$ is by at least $q \geq^s 0$. Since the number of possible actions is finite, if the values of $f_{y_{n'}}(s,a),\ a \in A^{(n')}(s)$ gradually decrease by at least q, then eventually, for some $n' \in \mathbb{N}$, $f_{y_{n'}}(s,a) \not\geq 0,\ \forall a \in A^{(n')}(s)$. For this state and time point, we get that $y_{n'}(s,a) = -\infty,\ \forall a \in A^{(n')}(s)$, and therefore $y^*_{n'}(s) = -\infty$. However, since $y^*_n(s)$ is monotonic nondecreasing in n, this means that for any $n'' \geq n'$, $y^*_{n''}(s) = -\infty$, which contradicts the fact that for this state, for any time point $n \in \mathbb{N}$, there exists a later time point, $n' \geq n$, at which $y^*_{n'}(s) \neq y^*_{n'+1}(s)$. We see that indeed at some time point $n = T \in \mathbb{N}$, $y^*_n(s)$ stops changing for all states, which proves item (b) in the integer multiple case.[1]

The proof of item (c) is made by induction, with similarity to that of item (a). Since by Lemma 6.1(a), $y^*(s) = \mathcal{F}\{y^*\}(s),\ \forall s \in S$, we conclude that $y^*(s) \leq \max_{a \in A(s)}\{\sigma - d(s,a) + \rho\} \triangleq y_1(s)$, for any $s \in S$. Next, assume that for some $n \in \mathbb{N}$, $y^*(s) \leq y_n(s),\ \forall s \in S$. Then,

$$y^*(s) = \mathcal{F}\{y^*\}(s) \leq \mathcal{F}\{y_n\}(s) \triangleq y_{n+1}(s),\ \forall s \in S,$$

which proves item (c).

[1] The same result can be shown for the general case where $d(\cdot,\cdot)$, ρ and σ are not all integer multiples of some basic quantity, using the fact that any decrease in $f_{y_n}(\cdot,\cdot)$ is by at least $(\min_{s \in S,\ a \in A(s):\ d(s,a) \geq^s \rho}\{d(s,a) - \rho\})$.

Finally, by applying $y_{T+1}(s) = y_T(s)$ into the definition of $y_{T+1}(s)$ we get that $y_T(s) = \mathcal{F}\{y_T\}(s),\ \forall s \in S$. Thus, $y_T(\cdot) \in \mathcal{Y}$, and therefore by Lemma 6.1(b), $y_T(s) \leq y^*(s),\ \forall s \in S$. Combining this result with that of item (c) above, we conclude that $y_T(s) = y^*(s),\ \forall s \in S$, proving (d).

■

6.1.2 Feasibility Determination Algorithm

As consequence of Lemmas 4.4 and 6.2, we obtain the following algorithm to compute $y^*(s)$ and to solve **BCF** in the stationary infinite horizon case.

Algorithm 4: Infinite Horizon BCPs - Feasibility Computation.

1. Let $y_1(s) = \max\limits_{a \in A(s)} \{\sigma - d(s,a) + \rho\},\ \forall s \in S$.

2. For $n = 1, 2, ...,$

   For $s \in S$,

   For $a \in A(s)$,

   Set $f_{y_n}(s,a) = \min\limits_{\substack{s' \in S: \\ p(s'|s,a) > 0}} y_n(s')$;

   Set $y_n(s,a) = \begin{cases} \min\{f_{y_n}(s,a), \sigma\} - d(s,a) + \rho, & \text{if } f_{y_n}(s,a) \geq 0; \\ -\infty, & \text{otherwise.} \end{cases}$

   Set $y_{n+1}(s) = \max_{a \in A(s)} y_n(s,a)$.

   If $y_{n+1}(s) = y_n(s),\ \forall s \in S$, then continue to step 3.

3. For any $s \in S$, the RBCP's feasibility threshold function is $y^*(s) = y_n(s)$, and the BCP's feasibility indicator function is $\Phi(s) = \begin{cases} 1, & \text{if } y^*(s) \geq 0; \\ 0, & \text{otherwise.} \end{cases}$

   The set of feasible initial BCP states is $S^F = \{s \in S :\ y^*(s) \geq 0\}$.

6.2 Objective Function Optimization

6.2.1 RBCP Objective Function Equations

In order to apply MDP algorithms to the stationary infinite horizon RBCP, we need equations of the form of (2.1), (2.2) and (2.3) for RBCPs. Note that the mentioned

equations cannot be applied to the RBCP optimization problem in their current form, since the RBCP constrains the feasible policies for optimization. Instead, we need to identify the feasible actions for each RBCP state, and the feasible RBCP policies. We establish suitable equations in this subsection.

In the following lemma we identify the set of feasible policies and find an RBCP equivalent to Equation (2.1). Towards this end, we make the following notations:

For any state $(s,y) \in \tilde{S}$, denote the (possibly empty) set of actions

$$A^F(s,y) \triangleq \{a \in A(s) : \ y + d(s,a) - \rho \leq \sigma \text{ and } \forall s' \in S \text{ s.t. } p(s'|s,a) > 0, \ (y + d(s,a) - \rho)^+ \leq y^*(s')\}. \quad (6.5)$$

Also denote by $\tilde{\Pi}^{HR,F} \subseteq \tilde{\Pi}^{HR}$ the space of general (history-dependent, randomized) policies over the RBCP, which assign only actions from $A^F(s,y)$ to feasible states, such that if $\tilde{\pi} \in \tilde{\Pi}^{HR,F}$, then for any $t \in \mathcal{T}$, if $(s_t, y_t) \in \tilde{S}^F$ then $a_t \in A^F(s_t, y_t)$.

Lemma 6.3. *Given an infinite horizon RBCP and objective function type $o \in \{\gamma, ea\}$,*

a. For any $(s,y) \in \tilde{S}$,

$$(s,y) \in \tilde{S}^F \ \Leftrightarrow \ A^F(s,y) \neq \emptyset.$$

b. For any $(s,y) \in \tilde{S}^F$ and $a \in A(s)$,

$$a \in A^F(s,y) \ \Leftrightarrow \ \begin{cases} y + d(s,a) - \rho \leq \sigma \text{ and} \\ \forall (s',y') \in \tilde{S} \text{ s.t. } \tilde{p}\big((s',y')|(s,y),a\big) > 0, \ (s',y') \in \tilde{S}^F. \end{cases}$$

c. For any feasible RBCP state, $(s,y) \in \tilde{S}^F$,

$$\tilde{J}_o^{F^*}(s,y) = \max_{\tilde{\pi} \in \tilde{\Pi}^{HR,F}} \tilde{J}_o^{\tilde{\pi}}(s,y). \quad (6.6)$$

Proof. To prove item (a) of the lemma, observe the following equivalences for any $(s,y) \in \tilde{S}$, using Lemmas 4.6(a) and 6.1(a).

$$\begin{aligned} (s,y) \in \tilde{S}^F &\Leftrightarrow \tilde{\Phi}(s,y) = 1 \\ &\Leftrightarrow y \leq y^*(s) \\ &\Leftrightarrow \ \exists a \in A(s) : \begin{cases} y + d(s,a) - \rho \leq \sigma \text{ and} \\ \forall s' \in S \text{ s.t. } p(s'|s,a) > 0, \ (y + d(s,a) - \rho)^+ \leq y^*(s') \end{cases} \\ &\Leftrightarrow A^F(s,y) \neq \emptyset. \end{aligned}$$

For proof of item (b), recall the following equivalence, for any $(s,y) \in \tilde{S}$, $a \in A(s)$, and $(s',y') \in \tilde{S}$:

$$\tilde{p}\left((s',y')|(s,y),a\right) > 0 \quad \Leftrightarrow \quad p(s'|s,a) > 0 \ \text{ and } \ y' = (y + d(s,a) - \rho)^+.$$

Also recall, using using Lemma 4.6(a), that for any $(s,y) \in \tilde{S}$,

$$\begin{aligned} y \leq y^*(s) \ &\Leftrightarrow \ \tilde{\Phi}(s,y) = 1 \\ &\Leftrightarrow \ (s,y) \in \tilde{S}^F. \end{aligned}$$

Then observe that for any $(s,y) \in \tilde{S}^F$ and $a \in A(s)$,

$$\begin{aligned} a \in A^F(s,y) \ &\Leftrightarrow \ \begin{cases} y + d(s,a) - \rho \leq \sigma \text{ and} \\ \forall s' \in S \text{ s.t. } p(s'|s,a) > 0, \ (y + d(s,a) - \rho)^+ \leq y^*(s') \end{cases} \\ &\Leftrightarrow \ \begin{cases} y + d(s,a) - \rho \leq \sigma \text{ and} \\ \forall (s',y') \in \tilde{S} : \text{ s.t. } \tilde{p}((s',y')|(s,y),a) > 0, \ (s',y') \in \tilde{S}^F. \end{cases} \end{aligned}$$

To prove item (c), recall that when $\tilde{\pi} \in \tilde{\Pi}^{HR,F}$, actions are chosen only from $A^F(s_t,y_t)$ at any time point t and state $(s_t,y_t) \in \tilde{S}$. Using item (b) above, observe that under $\tilde{\Pi}^{HR,F}$, for any feasible initial state $(s,y) \in \tilde{S}^F$, at any time point $t \in \mathcal{T}$, $(s_t,y_t) \in \tilde{S}^F$ and $y_t + d(s_t,a_t) - \rho \leq \sigma$ with probability 1. Thus,

$$\tilde{\pi} \in \tilde{\Pi}^{HR,F} \ \Leftrightarrow \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}) = 1, \ \forall (s,y) \in \tilde{S}^F.$$

Using the definition of $\tilde{J}_o^{F^*}(\cdot)$,

$$\begin{aligned} \tilde{J}_o^{F^*}(s,y) &\triangleq \max_{\tilde{\pi} \in \tilde{\Pi}^{HR}} \{\tilde{J}_o^{\tilde{\pi}}(s,y) : \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(B) = 1\} \\ &= \max_{\tilde{\pi} \in \tilde{\Pi}^{HR,F}} \tilde{J}_o^{\tilde{\pi}}(s,y), \quad \forall (s,y) \in \tilde{S}^F. \end{aligned}$$

■

We see that for any feasible state $(s,y) \in \tilde{S}^F$, the action set $A^F(s,y)$ contains all possible feasible actions, such that if the system is in a feasible state $(s,y) \in \tilde{S}^F$ and action $a \in A^F(s,y)$ is performed, then for any possible state history that may occur, the burstiness constraint could be obeyed. The set of policies $\tilde{\Pi}^{HR,F}$ comprises all of the control policies under which the burstiness constraint is obeyed, for all feaisble states.

Thus, the stationary infinite horizon MDP $\left\langle \mathcal{T}, \tilde{S}^F, \left(A^F(s,y)\right)_{(s,y) \in \tilde{S}^F}, \tilde{\mathcal{P}}, \mathcal{R} \right\rangle$ constitutes an unconstrained MDP whose optimal value is the same as RBCP's, for any initial

state, and we can apply to it the relevant standard MDP algorithms such as value iteration, policy iteration and linear programming.

Towards this end, we denote by $\tilde{\Pi}^{SR,F}$ and $\tilde{\Pi}^{SD,F}$ the spaces of stationary randomized policies and stationary deterministic policies, which assign only actions from $A^F(s,y)$ to feasible states, such that for any $\tilde{\pi} \in \tilde{\Pi}^{SR,F}$, $\tilde{\pi}(\cdot|s,y) \in \mathscr{P}(A^F(s,y)),\ \forall (s,y) \in \tilde{S}^F$, and for any $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$, $\tilde{\pi}(s,y) \in A^F(s,y),\ \forall (s,y) \in \tilde{S}^F$.

In order to apply the value iteration and policy iteration algorithms to infinite horizon RBCPs with an expected discounted reward, we need Bellman-type equations, similar to (2.2) and (2.3). Although Equation (2.2) does apply for RBCPs, for computational reasons it would be favorable to write such an equation for the feasible states only. Thus, we make the following useful notations:

For any policy $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$, denote by $\tilde{J}_\gamma^{\tilde{\pi},F}(\cdot) : \tilde{S}^F \to \mathbb{R}$ the value function of $\tilde{\pi}$, for the feasible states only, such that

$$\tilde{J}_\gamma^{\tilde{\pi},F}(s,y) \triangleq \tilde{J}_\gamma^{\tilde{\pi}}(s,y),\ \forall s \in \tilde{S}^F.$$

Denote by $\tilde{V}^F$ the space of real-valued functions on $\tilde{S}^F$, $\tilde{V}^F \triangleq \{\tilde{v} : \tilde{S}^F \to \mathbb{R}\}$. When $\tilde{S}^F$ is finite, $\tilde{V}^F$ is equivalent to the space of $|\tilde{S}^F|$-vectors on $\mathbb{R}$, i.e., $\mathbb{R}^{|\tilde{S}^F|}$. Hence, we will interchangeably refer to any $\tilde{v} \in \tilde{V}^F$ as a function on $\tilde{S}^F$ and as a vector in $\mathbb{R}^{|\tilde{S}^F|}$. In particular, observe that $\tilde{J}_\gamma^{\tilde{\pi},F} \in \tilde{V}^F$.

$\tilde{P}^{\tilde{\pi},F} \in [0,1]^{|\tilde{S}^F| \times |\tilde{S}^F|}$ and $\tilde{r}^{\tilde{\pi},F} \in \mathbb{R}^{|\tilde{S}^F|}$ are, respectively, the state transition matrix and immediate reward vector induced when using a policy $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$, such that

$$\begin{aligned}\tilde{P}^{\tilde{\pi},F}_{(s,y),(s',y')} &\triangleq \tilde{p}\big((s',y')|(s,y),\tilde{\pi}(s,y)\big) \\ &= p(s'|s,\tilde{\pi}(s,y)) \cdot \mathbb{1}\big(y' = (y + d(s,\tilde{\pi}(s,y)) - \rho)^+\big)\end{aligned}$$

and

$$\tilde{r}^{\tilde{\pi},F}_{(s,y)} \triangleq r(s,\tilde{\pi}(s,y)).$$

Lemma 6.4. *Given an infinite horizon RBCP with an expected discounted reward objective,*

a. For any $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$,

$$\tilde{J}_\gamma^{\tilde{\pi},F} = \tilde{r}^{\tilde{\pi},F} + \gamma \tilde{P}^{\tilde{\pi},F} \tilde{J}_\gamma^{\tilde{\pi},F}. \tag{6.7}$$

b. For any feasible RBCP state, $(s,y) \in \tilde{S}^F$,

$$\tilde{J}_\gamma^{F^*}(s,y) = \max_{a \in A^F(s,y)} \left\{ r(s,a) + \gamma \sum_{\substack{s' \in S: \\ p(s'|s,a)>0}} p(s'|s,a) \cdot \tilde{J}_\gamma^{F^*}\big(s', (y + d(s,a) - \rho)^+\big) \right\}. \tag{6.8}$$

Proof. Recall from Lemma 6.3(a) that if $(s,y) \in \tilde{S}^F$ then $A^F(s,y)$ is nonempty, and that by definition of $\tilde{\Pi}^{SD,F}$, any policy $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$ assigns only actions from $A^F(s,y)$ to feasible states $(s,y) \in \tilde{S}^F$. Then recall that by definition of $A^F(s,y)$, for any $(s,y) \in \tilde{S}^F$, $a \in A^F(s,y)$ and $(s',y') \in \tilde{S}$, if $\tilde{p}((s',y')|(s,y),a) > 0$, then $(s',y') \in \tilde{S}^F$.

Thus, for any $\tilde{\pi} \in \tilde{\Pi}^{SD,F}$ and $(s,y) \in \tilde{S}^F$,

$$\begin{aligned}
\tilde{J}_\gamma^{\tilde{\pi},F}(s,y) &\triangleq \tilde{\mathbb{E}}^{\tilde{\pi},(s,y)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \\
&= r(s, \tilde{\pi}(s,y)) + \gamma \sum_{(s',y') \in \tilde{S}} \tilde{p}\big((s',y')|(s,y), \tilde{\pi}(s,y)\big) \cdot \tilde{J}_\gamma^{\tilde{\pi}}(s',y') \\
&= r(s, \tilde{\pi}(s,y)) + \gamma \sum_{(s',y') \in \tilde{S}^F} \tilde{p}\big((s',y')|(s,y), \tilde{\pi}(s,y)\big) \cdot \tilde{J}_\gamma^{\tilde{\pi},F}(s',y') \\
&= \left(r^{\tilde{\pi},F} + \gamma \tilde{P}^{\tilde{\pi},F} \cdot J_\gamma^{\tilde{\pi},F} \right)_{(s,y)}.
\end{aligned}$$

Using the same arguments, for any $(s,y) \in \tilde{S}^F$,

$$\begin{aligned}
\tilde{J}_\gamma^{F^*}(s,y) &= \max_{\tilde{\pi} \in \tilde{\Pi}^{SD,F}} \tilde{J}_\gamma^{\tilde{\pi},F}(s,y) \\
&= \max_{\tilde{\pi} \in \tilde{\Pi}^{SD,F}} \left\{ \tilde{\mathbb{E}}^{\tilde{\pi},(s,y)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \right\} \\
&= \max_{a \in A^F(s,y)} \max_{\tilde{\pi}' \in \tilde{\Pi}^{SD,F}} \left\{ r(s,a) + \gamma \sum_{(s',y') \in \tilde{S}} \tilde{p}\big((s',y')|(s,y),a\big) \cdot \tilde{J}_\gamma^{\tilde{\pi}'}(s',y') \right\} \\
&= \max_{a \in A^F(s,y)} \max_{\tilde{\pi}' \in \tilde{\Pi}^{SD,F}} \left\{ r(s,a) + \gamma \sum_{(s',y') \in \tilde{S}^F} \tilde{p}\big((s',y')|(s,y),a\big) \cdot \tilde{J}_\gamma^{\tilde{\pi}',F}(s',y') \right\} \\
&= \max_{a \in A^F(s,y)} \left\{ r(s,a) + \gamma \sum_{(s',y') \in \tilde{S}^F} \tilde{p}\big((s',y')|(s,y),a\big) \cdot \tilde{J}_\gamma^{F^*}(s',y') \right\} \\
&= \max_{a \in A^F(s,y)} \left\{ r(s,a) + \gamma \sum_{\substack{s' \in S: \\ p(s'|s,a)>0}} p(s'|s,a) \cdot \tilde{J}_\gamma^{F^*}\big(s', (y + d(s,a) - \rho)^+\big) \right\}.
\end{aligned}$$

■

Equations (6.7) and (6.8) are the RBCP equivalents to Equations (2.2) and (2.3). With these in mind, we can apply the value iteration, policy iteration and linear programming algorithms to solve **RBCP** for the stationary infinite horizon case with an expected discounted reward objective.

6.2.2 Selection of State-Set

Towards applying the standard MDP algorithms to RBCPs in the infinite horizon case, we need to find a discrete set of feasible states, $\tilde{S}^F$. Using Lemma 4.6(a), we have $\tilde{S}^F = \left\{(s,y) \in \tilde{S} :\ y \leq y^*(s)\right\} = \{(s,y) :\ s \in S,\ y \in Y(s),\ y \leq y^*(s)\}$. Recall from Equation (4.9) that the sets $Y(s)$ can be chosen arbitrarily so long as

$$Y(s') \supseteq \big\{y' \in \mathbb{R}^K :\ y' = (y + d(s,a) - \rho)^+ \text{ where } s \in S,\ y \in Y(s),\ a \in A(s) \text{ and } p(s'|s,a) > 0\big\},\ \forall s' \in S.$$

First, note that $Y(s)$ can be bounded, as in the finite horizon case. Since we start with $y_0 = 0$, and by definition, $y_t \geq 0,\ \forall t \in \mathcal{T} \setminus 0$, we can bound $Y(s) \subseteq \mathbb{R}^K_+$. Consequently, we can constrain $\tilde{S}^F \subseteq \{(s,y) :\ s \in S,\ y \in [0, y^*(s)]\}$.

In the case where $d(\cdot,\cdot),\ \rho$ and σ are all integer multiples of some basic quantity $q \in \mathbb{R}^K$ (e.g., are natural numbers), so will be y_t, thus we may consider y to belong to the set of all integer multiples of q within $[0, y^*(s)]$. This is a simple, finite and discrete set, even if larger than the true set of possible feasible y_t values. In the general case where this property cannot be established, we should use a suitable uniform discretization of the range $[0, y^*(s)]$, and quantize/interpolate values to/from this set.

6.2.3 Value Iteration Algorithm

By applying the value iteration algorithm (see Section 2.1.7.2) to Equation (6.8), we obtain the following algorithm to solve **BCP** in the stationary infinite horizon case with an expected discounted reward objetcive.

Algorithm 5: Infinite Horizon BCPs with an Expected Discounted Reward Objective - Value Iteration.

0. State and action sets preparation:
   Denote $S^F = \{s \in S :\ y^*(s) \geq 0\}$.
   Select a suitable $\tilde{S}^F$ set as described in Section 6.2.2.

For any $(s,y) \in \tilde{S}^F$, set

$$A^F(s,y) = \{a \in A(s) :\ y + d(s,a) - \rho \leq \sigma \text{ and } \forall s' \in S :\ p(s'|s,a) > 0,\ (y + d(s,a) - \rho)^+ \leq y^*(s')\}.$$

1. Select some $\tilde{v}^0(s,y),\ \forall (s,y) \in \tilde{S}^F$, and specify $\epsilon > 0$.

2. For $n = 1, 2, ...,$

   For $(s,y) \in \tilde{S}^F$,

   For $a \in A^F(s,y)$, set

   $$\tilde{Q}^n(s,y,a) = r(s,a) + \gamma \sum_{\substack{s' \in S: \\ p(s'|s,a)>0}} p(s'|s,a) \cdot \tilde{v}^{n-1}\left(s', (y + d(s,a) - \rho)^+\right);$$

   Set $\tilde{v}^n(s,y) = \max\limits_{a \in A^F(s,y)} \tilde{Q}^n(s,y,a)$.

   If $\max\limits_{s \in S^F} \{\tilde{v}^n(s,0) - \tilde{v}^{n-1}(s,0)\} < \epsilon(1-\gamma)/2\gamma$, continue to step 3.

3. The RBCP policy $\tilde{\pi}_\epsilon^{F^*} \in \tilde{\Pi}^{SD,F}$, which is defined by

   $$\tilde{\pi}_\epsilon^{F^*}(s,y) \in \operatorname*{argmax}_{a \in A^F(s,y)} \tilde{Q}^n(s,y,a),\ \ \forall (s,y) \in \tilde{S}^F,$$

   is ϵ-optimal for **RBCP**, and $\tilde{J}_\epsilon^{F^*}(s,y) = \tilde{v}^n(s,y),\ \forall (s,y) \in \tilde{S}^F$ is an ϵ-optimal approximation of **RBCP**'s value function.

   The BCP policy $\pi_\epsilon^{F^*} = (\pi_t^{F^*})_{t=0}^\infty \in \Pi^{HD}$, which is defined by

   $$\pi_t^{F^*}(s_0, a_0, ..., s_t) \in \tilde{\pi}_\epsilon^{F^*}(s_t, y_t),$$

   for any $t \in \mathcal{T}$ and $(s_0, a_0, ..., s_t) \in H_t$, where

   $$y_0 = 0,\ \ y_{t+1} = (y_t + d(s_t, a_t) - \rho)^+,\ \ \forall t \in \mathcal{T},$$

   is ϵ-optimal for **BCP**. $J_\gamma^{\pi_\epsilon^{F^*}}(s) = \tilde{v}^n(s,0),\ \forall s \in S^F$ is an ϵ-optimal approximation of **BCP**'s value function, i.e., $\|J_\gamma^{\pi_\epsilon^{F^*}} - J_\gamma^{F^*}\|_\infty \leq \epsilon$.

Note that the optimal policy for the stationary infinite horizon BCP is stationary and deterministic in both the original and the augmented state-spaces.

6.2.4 Policy Iteration Algorithm

Using Equation (6.7), we can apply the policy iteration algorithm (see Section 2.1.7.3) to the stationary infinite horizon RBCP with an expected discounted reward objective, as follows.

Algorithm 6: Infinite Horizon BCPs with an Expected Discounted Reward Objective - Policy Iteration.

0. State and action sets preparation:
   Denote $S^F = \{s \in S :\ y^*(s) \geq 0\}$.
   Select a suitable $\tilde{S}^F$ set as described in Section 6.2.2.
   For any $(s,y) \in \tilde{S}^F$, set

$$A^F(s,y) = \{a \in A(s) :\ y + d(s,a) - \rho \leq \sigma \text{ and } \forall s' \in S :\ p(s'|s,a) > 0,\ (y + d(s,a) - \rho)^+ \leq y^*(s')\}.$$

1. Select some $\tilde{\pi}^0 \in \tilde{\Pi}^{SD,F}$.

2. For $n = 0, 1, ...,$
   Policy evaluation: Solve $(I - \gamma \tilde{P}^{\tilde{\pi}^n,F})\tilde{v}^n = \tilde{r}^{\tilde{\pi}^n,F}$ for $\tilde{v}^n \in \mathbb{R}^{|\tilde{S}^F|}$.
   Policy improvement: For $(s,y) \in \tilde{S}^F$, choose

$$\tilde{\pi}^{n+1}(s,y) \in \underset{a \in A^F(s,y)}{\operatorname{argmax}} \left\{ r(s,a) + \gamma \sum_{\substack{s' \in S: \\ p(s'|s,a)>0}} p(s'|s,a) \cdot \tilde{v}^n\big(s', (y + d(s,a) - \rho)^+\big) \right\},$$

   setting $\tilde{\pi}^{n+1}(s,y) = \tilde{\pi}^n(s,y)$ if possible.
   If $\tilde{\pi}^{n+1} = \tilde{\pi}^n$, continue to step 3.

3. The RBCP policy $\tilde{\pi}_\gamma^{F^*} = \tilde{\pi}^n \in \tilde{\Pi}^{SD,F}$ optimizes **RBCP**, and the optimal **RBCP** value is $\tilde{J}_\gamma^{F^*}(s,y) = \tilde{v}^n(s,y),\ \forall (s,y) \in \tilde{S}^F$.
   The BCP policy $\pi_\gamma^{F^*} = (\pi_t^{F^*})_{t=0}^\infty \in \Pi^{HD}$, which is defined by

$$\pi_t^{F^*}(s_0, a_0, ..., s_t) = \tilde{\pi}^n(s_t, y_t),$$

   for any $t \in \mathcal{T}$ and $(s_0, a_0, ..., s_t) \in H_t$, where

$$y_0 = 0,\quad y_{t+1} = (y_t + d(s_t, a_t) - \rho)^+,\quad \forall t \in \mathcal{T},$$

   optimizes **BCP**. The optimal **BCP** value is $J_\gamma^{F^*}(s) = \tilde{v}^n(s,0),\ \forall s \in S^F$.

6.2.5 Linear Programming Algorithm

Following the above derivations, we can apply the linear programming algorithm (see Section 2.1.7.4) to the RBCP as follows.

Algorithm 7: Infinite Horizon BCPs with an Expected Discounted Reward Objective - Linear Programming.

0. State and action sets preparation:
   Denote $S^F = \{s \in S :\ y^*(s) \geq 0\}$.
   Select a suitable $\tilde{S}^F$ set as described in Section 6.2.2.
   For any $(s, y) \in \tilde{S}^F$, set

$$A^F(s,y) = \{a \in A(s) :\ y + d(s,a) - \rho \leq \sigma \text{ and } \forall s' \in S :\ p(s'|s,a) > 0,\ (y + d(s,a) - \rho)^+ \leq y^*(s')\}.$$

1. Solve the following linear program:

$$\tilde{x}^* = \underset{\tilde{x}}{\operatorname{argmax}} \sum_{(s,y)\in\tilde{S}^F} \sum_{a\in A^F(s,y)} r(s,a) \cdot \tilde{x}(s,y,a),$$

   subject to

$$\sum_{a'\in A^F(s',y')} \tilde{x}(s',y',a') - \gamma \sum_{\substack{(s,y)\in\tilde{S}^F,\ a\in A^F(s,y):\\ y'=(y+d(s,a)-\rho)^+}} p(s'|s,a)\cdot\tilde{x}(s,y,a) = \tilde{\alpha}(s',y'),\quad \forall (s',y') \in \tilde{S}^F,$$

$$\tilde{x}(s,y,a) \geq 0,\quad \forall (s,y) \in \tilde{S}^F,\ a \in A^F(s,y).$$

2. The RBCP policy $\tilde{\pi}_\gamma^{F^*} \in \tilde{\Pi}^{SR}$, which is defined by

$$\tilde{\pi}_\gamma^{F^*}(a|s,y) = \begin{cases} \dfrac{\tilde{x}^*(s,y,a)}{\sum_{a'\in A^F(s,y)} \tilde{x}^*(s,y,a')}, & \text{if } a \in A^F(s,y); \\ 0 & \text{otherwise,} \end{cases}$$

   for any $(s, y) \in \tilde{S}^F$ and $a \in A(s)$, optimizes **RBCP**, and the optimal **RBCP** value is $\tilde{J}_\gamma^{F^*} = \tilde{J}_\gamma^{\tilde{\pi}_\gamma^{F^*}}$.

The BCP policy $\pi_\gamma^{F^*} = (\pi_t^{F^*})_{t=0}^{\infty} \in \Pi^{HR}$, which is defined by

$$\pi_t^{F^*}(a|s_0, a_0, ..., s_t) = \begin{cases} \dfrac{\tilde{x}^*(s_t, y_t, a)}{\sum_{a' \in A^F(s_t, y_t)} \tilde{x}^*(s_t, y_t, a')}, & \text{if } a \in A^F(s_t, y_t); \\ 0 & \text{otherwise,} \end{cases}$$

for any $t \in \mathcal{T}$, $(s_0, a_0, ..., s_t) \in H_t$, and $a \in A(s_t)$, where

$$y_0 = 0, \quad y_{t+1} = (y_t + d(s_t, a_t) - \rho)^+, \quad \forall t \in \mathcal{T},$$

optimizes **BCP**, and the optimal **BCP** value is $J_\gamma^{F^*}(s) = \tilde{J}_\gamma^{F^*}(s, 0), \ \forall s \in S$.

We see that in the infinite horizon case, the optimal policy is stationary in the extended state space, but history-dependent in the original BCP state space. Thus, in order to determine the action to be performed at every time point, we must calculate the deficit term, y_t, on the fly.

As in the finite horizon case, the algorithms' complexities depend heavily on the level of discretization of the deficit term, which determines the size of the RBCP state space.

Chapter 7

Examples

In this chapter we demonstrate the application of some of our algorithms to BCP examples, and explore the burstiness constraint's impact on the BCP's solution.

7.1 Setup

Consider the system described in the Introduction (Chapter 1). A queue is sending jobs to a job-processing server. The queue's state is described by the number of jobs in the queue at every time point, s_t. Our queue can hold $\bar{S}$ jobs at most, such that the state space is $S = \{0, ..., \bar{S}\}$. We control the queue by sending out a number of jobs to the server at each time point, a_t, so that the set of available actions for state $s \in S$ is $A(s) = \{0, ..., s\}$. The number of incoming jobs at the queue is denoted by x_t, and is Poisson-distributed, i.e., $x_t \sim \text{Pois}(\lambda)$ with some $\lambda > 0$. Thus, the queue's state transition equation is $s_{t+1} = \min\{s_t + x_t - a_t, \bar{S}\}$. The system is illustrated in Figure 7.1.

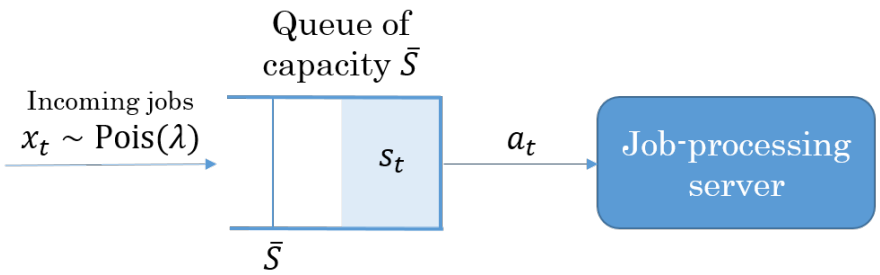

Figure 7.1: Job queue for a server

We would like to maximize the expected discounted number of jobs being processed by the server, resulting with a reward function of $r(s, a) = a$. In addition, a cost

of $d(s,a)$ is induced at every time point. The cost sequence is required to be (σ,ρ) burstiness-constrained. The resulting optimization problem is

$$\max_{\pi\in\Pi^{HR}} \quad \mathbb{E}^{\pi,s}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]$$
$$\text{s.t.} \quad \sum_{t=t_1}^{t_2} d(s_t,a_t) \le \rho(t_2-t_1+1)+\sigma, \quad \forall t_1,t_2\in\{0,1,...\}:\ t_1\le t_2, \quad \text{w.p. } 1.$$

In the following analysis, we will examine the problem's feasibility and its optimal value and policy, depending on the cost function $d(s,a)$ and the burstiness coefficients, σ and ρ. We use a maximal queue capacity of $\bar{S}=3$, and $\lambda=1$ as the event rate of incoming jobs at the queue. For optimal value and policy calculation, we use a discount factor of $\gamma=0.2$, and an error threshold of $\epsilon=10^{-5}$ for the value iteration algorithm.

7.2 Example 1

In our first example, a cost is incurred per every job sent to the server, yielding a cost function of $d(s,a)=a$.

7.2.1 Feasibility

First, let us examine the problem's feasibility per different values of the burstiness coefficients, σ and ρ. Using our Feasibility Determination Algorithm (Algorithm 4), we calculate the feasibility threshold function of each state. The results are presented in Table 7.1.

		$y^*(s)$			
σ	ρ	$s=0$	$s=1$	$s=2$	$s=3$
0	0	0	0	0	0
0	2	2	2	2	2
0	3	3	3	3	3
3	0	3	3	3	3
3	1	4	4	4	4

Table 7.1: Dependence of the feasibility threshold function on the burstiness coefficients when $d(s,a)=a$

Note that when the queue only pays for the jobs sent, i.e., $d(s,a) = a$, the problem is feasible regardless of the burstiness coefficient values. This is because at any time point and any state, there exists at least one feasible action, $a = 0$, which translates to not sending any jobs at all. Thus, we can always make sure that the cost doesn't exceed burstiness constraints. This is seen in the above reuslts, where $y^*(s) \geq 0$ for any $\sigma, \rho \geq 0$. This result complies with our sufficient condition for feasibility from Claim 3.2.

Increasing the burstiness coefficient values leads to an increase in the feasibility threshold function, $y^*(s)$, expressing the increase in the maximum allowed deficit when the system can allow more spending.

However, as will be seen in the next subsection, although different values of (σ, ρ) can achieve the same $y^*(s)$ values, they exhibit different BCP optimal values.

7.2.2 Optimal Value and Policy

Next, we examine the burstiness constraint's influence on the system's optimal value and policy. Using our Extended Value Iteration Algorithm (Algorithm 5), we calculate the queue's maximum feasible expected discounted reward, and the policy which leads to this value. For comparison, we also calculate the optimal value and policy of the unconstrained problem. The results are presented in Table 7.2.

Recall that the optimal RBCP policy, $\tilde{\pi}^{F^*}$, is defined on the augmented state space, $\tilde{S} = \{(s,y) : \ s \in S, \ y \in Y(s)\}$. First, note the dependence of the y state-space, $Y(s)$, on $y^*(s)$: $Y(s) = \{0, ..., y^*(s)\}, \ \forall s \in S$. Thus, in $(\sigma, \rho) = (0,0)$, where $y^*(s) = 0, \ \forall s \in S$, the only feasible y value is 0, whereas when $y^*(s) = 3, \ \forall s \in S$, the y state space is $Y(s) = \{0,1,2,3\}, \ \forall s \in S$.

When $(\sigma, \rho) = (0,0)$, the queue can't afford any spending on sent jobs. The only feasible policy is not sending jobs to the server at all, $\tilde{\pi}^{F^*}(s,0) = 0, \ \forall s \in S$, regardless of the number of pending jobs. Consequently, no reward will be accrued, leading to an optimal value of 0.

However, when $(\sigma, \rho) = (0,3)$ the problem's optimal value is equal to that of the unconstrained problem. This is due to the the deficit variable's state dynamics: Recall that the system starts with $y_0 = 0$ and that $y_{t+1} = (y_t + d(s_t, a_t) - \rho)^+, \ \forall t > 0$. Since the cost is $d(s,a) = a$, it is always $\bar{S}$ at most. When $\rho \geq \bar{S}$, this means that $y_t = 0$ at any time point, under any policy. In addition, whenever $\rho \geq \bar{S}$ and the deficit is $y_t = 0$, we can allow to send all jobs in queue to the server, $\tilde{\pi}^{F^*}(s,0) = s, \ \forall s \in S$, as would be done in the unconstrained problem to maximize the value.

An interesting result is seen when $(\sigma, \rho) = (3,0)$: Although this case has the same

	$J_\gamma^{F^*}(s)$				$\tilde{\pi}_\gamma^{F^*}(s,y)$
	$s=0$	$s=1$	$s=2$	$s=3$	
MDP	0.24	1.24	2.24	3.24	s: 0 1 2 3 $\pi_\gamma^*(s)$: 0 1 2 3
BCP, $\sigma=0,\rho=0$	0	0	0	0	s \ y: 0 1 2 3 0: 0 0 0 0
BCP, $\sigma=0,\rho=3$	0.24	1.24	2.24	3.24	s \ y: 0 1 2 3 0: 0 1 2 3 1: 0 1 2 2 2: 0 1 1 1 3: 0 0 0 0
BCP, $\sigma=3,\rho=0$	0.23	1.2	2.14	3	s \ y: 0 1 2 3 0: 0 1 2 3 1: 0 1 2 2 2: 0 1 1 1 3: 0 0 0 0

Table 7.2: Dependence of the optimal value and policy on the burstiness coefficients when $d(s,a)=a$

$y^*(s)$ values **and the same optimal policy** as the case $(\sigma,\rho)=(0,3)$, it leads to a **lower optimal value**. This is due to ρ's effect on y_t's state dynamics. Since $\rho<\bar{S}$ in this case, states with y values higher than 0 are visited with nonzero probability under any policy. At such states, we cannot allow to send all jobs in queue to the server, leading to a lower expected discounted reward.

7.3 Example 2

In the next example, a cost is incurred for both sending jobs to the server and holding jobs in queue, i.e., $d(s,a)=s+a$.

7.3.1 Feasibility

Table 7.3 presents the feasibility threshold function for different values of the burstiness coefficients.

In this case, when $(\sigma,\rho)=(0,2)$, there does not exist a feasible policy for the queue, since $y^*(s)<0$ for all states. This is because for state $s=3$, there does not exist a feasible action $a\in A(s=3)=\{0,...,3\}$ for which one step of the burstiness constraint is obeyed, i.e., $d(s=3,a)\leq\rho=2$. Since in our queue example, every pair of states is

		$y^*(s)$			
σ	ρ	$s=0$	$s=1$	$s=2$	$s=3$
0	2	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	3	3	2	1	0
10	2	$-\infty$	$-\infty$	$-\infty$	$-\infty$
1	3	4	3	2	1

Table 7.3: Dependence of the feasibility threshold function on the burstiness coefficients when $d(s,a) = s + a$

communicating for any given policy (that is, every state is accessible from any state given any possible policy), all of the queue's states are infeasible as well.

By increasing ρ to 3, the above problem is solved: performing action $a = 0$ in state $s = 3$ preserves obedience of the burstiness constraint and does not increase the system's deficit. We see that all values of $y^*(s)$ are non-negative when $(\sigma, \rho) = (0, 3)$.

Alternatively, increasing σ to 10 does not make the problem feasible (in fact, when $\rho = 2$, no value of σ would lead to feasibility). This is because under any policy, the initial "credit" represented by σ will eventually be spent with nonzero probability. For example, visiting state $s = 3$ for 11 times in a row would lead to violation of the burstiness constraint, and this history sequence has a nonzero probability under any possible policy. These results demonstrate our necessary condition for feasibility in Claim 3.1.

However, when $(\sigma, \rho) = (1, 3)$, the feasibility threshold function does change, expressing the increase in the maximum allowed deficit in this case.

Compare the above results with those of Table 7.1, and observe the difference between $y^*(s)$ values in both cases for the same (σ, ρ) values. In particular, observe that when the cost was independent of the state s, i.e., $d(s,a) = a$, $y^*(s)$ had the same values for different states. In the current example, where $d(s,a) = s + a$, breaking the symmetry of the cost between states leads to dependence of $y^*(s)$ on the state s.

We remark that situations at which some states are feasible and others aren't arise in BCPs that are not fully-communicating (i.e., when there exist policies under which some states are inaccessible from other states).

7.3.2 Optimal Value and Policy

Table 7.4 presents the optimal value and policy for different values of the burstiness coefficients.

In this case as well, we can see the dependence of the y state-space on $y^*(s)$: $Y(s) = \{0, ..., y^*(s)\},\ \forall s \in S$. Since $y^*(s)$ has different values for different states, not

	$J_\gamma^{F^*}(s)$				$\tilde{\pi}_\gamma^{F^*}(s,y)$				
	$s=0$	$s=1$	$s=2$	$s=3$					
MDP	0.24	1.24	2.24	3.24	s	0	1	2	3
					$\pi_\gamma^*(s)$	0	1	2	3
BCP, $\sigma=0, \rho=3$	0.14	1.14	1.17	0	y \ s	0	1	2	3
					0	0	1	1	0
					1	0	1	0	-
					2	0	0	-	-
					3	0	-	-	-
BCP, $\sigma=1, \rho=3$	0.2	1.2	2.15	1.09	y \ s	0	1	2	3
					0	0	1	2	1
					1	0	1	1	0
					2	0	1	0	-
					3	0	0	-	-
					4	0	-	-	-

Table 7.4: Dependence of the optimal value and policy on the burstiness coefficients when $d(s,a) = s + a$

all deficit values (y) are feasible at all states.

Compare the above results with those of Table 7.2: We see that the change in cost function also leads to a different optimal policy and values, for the same (σ, ρ) values. Also observe that increasing the burstiness coefficient values allows more spending, leading to higher optimal values, albeit lower than those of the unconstrained problem.

Chapter 8

Summary, Conclusion and Future Work

The burstiness constraint models a limit on a process's long-term average cost, except for a prescribed amount of deviations. In this research we developed a framework for handling burstiness-constrained MDPs (BCPs).

We start with reformulating the burstiness constraint and posing it as a simple set if inequalities, one per every time point, by formulating the maximum deficit that has been acquired by any time point, y_t.

We then handle the BCP feasibility problem and propose dynamic programming-type algorithms to solve it. This is done by characterizing the maximum allowed deficit at every time point, $y_t^*(s)$. In the finite horizon case, we write a backwards induction algorithm, starting from the final time point and recursively progressing to the BCP's initial states. In infinite horizon BCPs, we find $y^*(s)$ by solving a fixed-point equation. We prove that our solution algorithm is finite when the cost function $d(s,a)$ and the burstiness coefficients σ, ρ are all integer multiples of the same quantity. However, this can be proved for the general case as well.

Next, we propose a scheme for solving BCP optimization problems, by constructing an unconstrained MDP, as follows. Since y_t embodies the entire past information necessary at every time point, the feasibility of every action depends only on the state s_t and deficit y_t at a given time point. By augmenting the BCP's state-space to include y_t as part of the state variable, the burstiness constraint turns into a state constraint. The optimal policies are then Markov in the augmented state-space. Using only feasible states and actions in the reformulated MDP yields an unconstrained MDP with an augmented state-space. This MDP can be optimized with common MDP algorithms such as backwards induction, value iteration, policy iteration and linear programming formulation. We write these algorithms for the reformulated BCP, and relate their

solutions to the optimal BCP value and policy.

Examining the optimal BCP policy and value for various examples demonstrates several effects:

1. Feasibility of a BCP depends on the state transition dynamics, the cost function, and the burstiness coefficients. If for any state s there exists an action a for which the cost is zero, i.e., $d(s,a) = 0$, then the problem is feasible for any value of $\sigma, \rho \geq 0$. However, if a BCP is infeasible for certain values of σ, ρ, increasing the burstiness coefficients may turn the problem feasible.

2. In feasible BCPs, increasing the burstiness coefficients gives the system more flexibility to use actions with a higher cost ($d(s,a)$) while still obeying the burstiness constraint, in order to increase the objective function, yielding a higher optimal value.

3. A policy in the extended state-space yields different values of the objective for different values of ρ. This is because ρ affects the dynamics of the extended state-space. consequently, Two sets of burstiness coefficients may share the same optimal policy but lead to different optimal values.

We conclude this work by pointing out several directions for future research on this topic.

1. An interesting relationship to consider is that between the feasibilty threshold function, $y^*(s)$, and the burstiness coefficient σ. Recall from its definition in Equation (4.20) that $y^*(s)$ is calculated for a certain value of σ. Hence, in the following discussion we will denote it by $y^*_\sigma(s)$. Clearly, when σ increases we can allow more burstiness and the maximal allowed initial deficit can be higher, leading to a larger $y^*_\sigma(s)$. We suggest the following connection between the two, for a given value of ρ. First, we denote by $\tilde{B}_\sigma$ the event of the RBCP obeying the extended burstiness constraint, given $\sigma \in \mathbb{R}^K_+$:

$$\tilde{B}_\sigma \triangleq \left\{ \tilde{\omega} \in \tilde{\Omega} : \ y_t + d_t(s_t, a_t) - \rho \leq \sigma, \ \ \forall t \in \mathcal{T} \right\}.$$

   We then define $\sigma^*(s,y)$ as the minimal σ for which there exists a feasible policy, when $s_0 = s \in S_0$ and $y_0 = y \in \mathbb{R}^K$:

$$\sigma^*(s,y) \triangleq \inf \left\{ \sigma \in \mathbb{R}^K_+ : \ \ \exists \tilde{\pi} \in \tilde{\Pi}^{HR} \ \text{ s.t. } \ \tilde{\mathbb{P}}^{\tilde{\pi},(s,y)}(\tilde{B}_\sigma) = 1 \right\}.$$

We also define the minimal σ for which there exists a feasible BCP policy for any BCP initial state:

$$\sigma^* \triangleq \inf\left\{\sigma \in \mathbb{R}_+^K : \quad \exists \pi \in \Pi^{HR} \text{ s.t. } \mathbb{P}^{\pi,s}(B_\sigma) = 1, \quad \forall s \in S_0\right\}.$$

With these definitions, we make the following arguments:

a. For any $\sigma \in \mathbb{R}_+^K$, the corresponding feasibility threshold function $y_\sigma^*(s)$ can be found using the following equation:

$$y_\sigma^*(s) = \sigma - \sigma^*(s, 0), \quad \forall s \in S_0.$$

b. The minimal σ for feasibility of the BCP at any initial state is

$$\sigma^* = \max_{s \in S_0} \sigma^*(s, 0).$$

c. For a stationary infinite horizon BCP, $\sigma^*(s, y)$ satisfies the following equation:

$$\sigma^*(s, y) = \min_{a \in A(s)} \max\left\{ y + d(s, a) - \rho, \max_{\substack{s' \in S: \\ p(s'|s,a) > 0}} \sigma^*(s', (y + d(s, a) - \rho)^+) \right\}.$$

This function can be found using a method similar to the one from Section 6.1, of successive evaluations of a suitable operator until convergence to a fixed point. For a finite horizon BCP, we can write a similar dynamic programming-type set of equations, and calculate $\sigma^*(s, y)$ via backwards induction.

2. A similar analysis can be performed on the burstiness coefficient ρ, where we would like to know what the minimal ρ for BCP feasibility at any initial state is. In addition, we would like to study the connection between ρ and the BCP's expected average cost, e.g., $\liminf_{N\to\infty} \left\{ \frac{1}{N+1} \mathbb{E}^{\pi,s} \left[\sum_{t=0}^{N} d(s_t, a_t) \right] \right\}$ in a stationary infinite horizon BCP.

3. In this work we assumed discrete-time systems with finite state and action spaces. We relied on these assumptions in the feasibility determination algorithms and in extending standard finite MDP algorithms to our BCPs. A natural sequel to this work is to extend our framework to continuous-time systems and to countable state and action spaces.

4. A different MDP framework where burstiness constraints can be invloved is that of optimal control in a burstiness-constrained environment. . Whereas BCPs attempt to optimize our policy while obeying burstiness constraints, we can also consider a

game-like setting where our goal is to optimize our policy when playing against an agent which is known to be burstiness-constrained. Such a game can model, for example, an input process to a queue which is known to be burstiness-constrained. This knowledge about the oponent can be used to maximize our profit.

Bibliography

[Alt99] Eitan Altman. *Constrained Markov Decision Processes*. CRC Press, 1999.

[Bla67] David Blackwell. Positive Dynamic Programming. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 415–418. University of California Press, 1967.

[CAK12] Yanpei Chen, Sara Alspaugh, and Randy Katz. Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads. *Proceedings of the VLDB Endowment*, 5(12):1802–1813, August 2012.

[Cru91] Rene L. Cruz. A Calculus for Network Delay. I. Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, 1991.

[ECA15] Mahmoud El Chamie and Behcet Açikmeşe. Finite-Horizon Markov Decision Processes with State Constraints. *arXiv:1507.01585 [cs, math]*, July 2015.

[EL90] A. E. Eckberg and D. M. Lucantoni. A Traffic/Performance Analysis of the Bandwidth Management Throughput-Burstiness Filter. In *Proceedings of the 29th IEEE Conference on Decision and Control*, volume 4, pages 2118–2123, 1990.

[FKR95] Jerzy A. Filar, Dmitry Krass, and Keith W. Ross. Percentile Performance Criteria for Limiting Average Markov Decision Processes. *IEEE Transactions on Automatic Control*, 40(1):2–10, January 1995.

[HK84] A. Hordijk and L. C. M. Kallenberg. Constrained Undiscounted Stochastic Dynamic Programming. *Mathematics of Operations Research*, 9(2):276–289, May 1984.

[KA95] Takis Konstantopoulos and Venkat Anantharam. Optimal Flow Control Schemes that Regulate the Burstiness of Traffic. *IEEE/ACM Transactions on Networking*, 3(4):423–432, 1995.

[LV91] S. Low and P. Varaiya. A Simple Theory of Traffic and Resource Allocation in ATM. In *Global Telecommunications Conference, 1991*, volume 3, pages 1633–1637, December 1991.

[MOSM90] M. Murata, Y. Oie, T. Suda, and H. Miyahara. Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks. *IEEE Journal on Selected Areas in Communications*, 8(3):447–458, April 1990.

[Piu06] A. B. Piunovskiy. Dynamic Programming in Constrained Markov Decision Processes. *Control and Cybernetics*, 35(3):645, 2006.

[Put14] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[Ros89] Keith W. Ross. Randomized and Past-Dependent Policies for Markov Decision Processes with Multiple Constraints. *Operations Research*, 37(3):474, June 1989.

[RV89] Keith W. Ross and Ravi Varadarajan. Markov Decision Processes with Sample Path Constraints: The Communicating Case. *Operations Research*, 37(5):780, October 1989.

[RV91] Keith W. Ross and Ravi Varadarajan. Multichain Markov Decision Processes with a Sample Path Constraint: A Decomposition Approach. *Mathematics of Operations Research*, 16(1):195, February 1991.

[StaS99] David Starobinski and Moshe Sidi. Stochastically Bounded Burstiness for Communication Networks. In *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, pages 36–42, 1999.

[Tur86] Jonathan Turner. New Directions in Communications (or Which Way to the Information Age?). *IEEE Communications Magazine*, 24(10):8–15, 1986.

[TZL11] Jianzhe Tai, Juemin Zhang, Jun Li, Walid Meleis, and Ningfang Mi. ArA: Adaptive Resource Allocation for Cloud Computing Environments under Bursty Workloads. In *Proceedings of the 30th International IEEE Conference on Performance Computing and Communications Conference (IPCCC)*, pages 1–8, 2011.

[Val01] S. Valaee. A Recursive Estimator of Worst-Case Burstiness. *IEEE/ACM Transactions on Networking*, 9(2):211–222, April 2001.

[Whi88] D. J. White. Mean, Variance, and Probabilistic Criteria in Finite Markov Decision Processes: a Review. *Journal of Optimization Theory and Applications*, 56(1):1–29, 1988.

המדיניות הנתונה.

עקב מבנה זה של אילוץ הפרציות, המדיניויות הקבילות אינן מרקוביות כי אם תלויות היסטוריה, ככלל. כתוצאה מכך, בדיקת קיומן של מדיניויות קבילות הינה תהליך מסורבל ולא־יעיל, במתכונתו הנוכחית של האילוץ. כמו כן, לא ניתן לפתור את בעיית האופטימיזציה באמצעות הכלים המקובלים לפתרון תה"מ, כגון ניסוח בצורת בעיית תכנות דינמי או בעיית תכנות לינארי.

אנו מציעים אלגוריתמים לקביעת קיומן של מדיניויות קבילות לבעיה, ולמציאת מדיניות קבילה אופטימלית, בתנאי אופק זמן סופי ואינסופי.

בשלב הראשון, אנו מנסחים מחדש את אילוץ הפרציות כך שיכלול אי־שוויון אחד בלבד לכל נקודת זמן. עבור אופק זמן סופי, האלגוריתם לבדיקת קיומן של מדיניויות קבילות מבצע אינדוקציה לאחור: בכל צעד זמן, אנו בודקים אם ניתן לקיים את האילוץ מנקודת זמן זו ועד סוף התהליך. בהיעדר אפשרות כזו עבור אופק זמן אינסופי, אנו מאפיינים את אילוץ הפרציות באמצעות משוואת נקודת שבת, ומוצאים אלגוריתם לפתרון המשוואה ולבדיקת קיומן של מדיניויות קבילות לבעיה.

לאחר מכן, אנו מרחיבים את מרחב המצב של התהליך, כך שמשתנה המצב כולל איבר נוסף המגלם בתוכו את המידע ההיסטורי הנחוץ לקביעת קיום האילוץ בכל נקודת זמן. במרחב המצב המורחב, מדיניויות המקיימות את האילוץ הן מרקוביות, ולכן בעלות מבנה פשוט יותר. לאחר שזיהינו את הפעולות המותרות בכל מצב מורחב, מתקבל תהליך החלטה מרקובי עם מרחב מצב מורחב וללא אילוצים. על תהליך זה ניתן להחיל את האלגוריתמים המקובלים לפתרון בעיות מסוג זה. לפיכך, אנו מנסחים אלגוריתמים למציאת מדיניות קבילה אופטימלית עבור מערכות באופק זמן סופי עם פונקציית רווח של תוחלת התגמול המצטבר (expected total reward), ועבור מערכות סטציונריות באופק זמן אינסופי עם פונקציית רווח של תוחלת התגמול המהוון (expected discounted reward).

לבסוף, אנו בוחנים את השפעתם של פונקציית העלות וקבועי הפרציות σ, ρ על הדוגמה הבאה: אנו שולטים בתור של עבודות, אשר שולח עבודות לביצוע בשרת. מצב המערכת הוא מספר העבודות בתור, ובכל רגע נתון אנו מחליטים כמה עבודות לשלוח לשרת. ברצוננו למצוא מדיניות בקרה שתתן מקסימום לתוחלת סכום העבודות שנשלחות לשרת (לאחר היוון), תחת אילוץ פרציות על פונקציית עלות כלשהי של מספר העבודות בתור ומספר העבודות שנשלחו לביצוע בכל רגע נתון.

כאשר העלות מוטלת רק על העבודות הנשלחות לביצוע, קיימת תמיד מדיניות קבילה. זאת כיוון שבכל מצב קיימת פעולה שעבורה העלות היא 0 - אי־שליחת עבודות לתור. לעומת זאת, כאשר העלות מוטלת הן על העבודות הנמצאות בתור והן על העבודות הנשלחות לביצוע, לא תמיד קיימת מדיניות קבילה. כתלות בקבועי הפרציות, ישנה פחות או יותר גמישות בהגדלת מספר העבודות הנשלחות לביצוע תוך קיום האילוץ, וכתוצאה מכך משתנה הערך האופטימלי של פונקציית הרווח המתקבלת. ניתן לראות כי יכולה להתקבל אותה מדיניות אופטימלית עבור ערכים שונים לקבועי הפרציות, אולם ערך פונקציית הרווח עבור כל מדיניות יהיה שונה, עקב השפעת קבועי הפרציות על דינמיקת התהליך במרחב המצב המורחב.

תקציר

פרציות (burstiness) של תהליך דינמי הינה התנהגות המאופיינת בשינויים פתאומיים בעוצמת התהליך או בתכיפותו. תהליכים פרציים מתקבלים, בין היתר, ברשתות תקשורת, מערכות אחסון קבצים ומערכות מחשוב־ענן. אותות כגון אלה גורמים לחוסר־איזון של העומס המוטל על משאבי הרשת ולהצטברות עומס חריג ברשת, המביאים לביצועים ירודים של מערכת התקשורת או המחשוב.

תהליכי החלטה מרקוביים (Markov Decision Processes) הינם מודל מקובל עבור מערכות קבלת החלטות רב־שלביות בסביבה אקראית. במערכות כאלה, בכל צעד זמן מתקבל תגמול (reward) כתלות במצב המערכת והפעולה המתבצעת באותו שלב, $r_t(s_t, a_t)$, ונדרש למצוא מדיניות בקרה המניבה מקסימום לפונקציית רווח כלשהי של התגמולים המתקבלים. קיימים מגוון אלגוריתמים לחישוב יעיל של מדיניות בקרה אופטימלית לבעיות אלה.

בעבודה זו אנו מתמקדים בתהליכי החלטה מרקוביים עם אילוצי פרציות. בתהליכים אלה, נוסף על התגמול המיידי המתקבל בכל צעד זמן, ישנה גם עלות (cost) מיידית, התלויה במצב המערכת והפעולה המתבצעת באותו שלב, $d_t(s_t, a_t)$. אנו מחפשים מדיניות אופטימלית לפונקציית הרווח של התהליך, תחת אילוץ על פרציות העלות המיידית המתקבלת תחת מדיניות זו. מודל הפרציות שאנו משתמשים בו דורש כי כל סכום של עלויות עוקבות יהיה חסום על־ידי גודל יחסי למספר העלויות העוקבות הנסכמות, בתוספת קבוע. בניסוח מתמטי, סדרת העלויות $(d_t)_{t\in\mathcal{T}}$ עומדת באילוץ הפרציות עם קבועי הפרציות $\sigma, \rho \geq 0$ אם היא מקיימת את התנאי הבא:

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

על סדרת העלויות $(d_t(s_t, a_t))_{t\in\mathcal{T}}$ לקיים תנאי זה בהסתברות 1 תחת המדיניות הנבחרת. להלן נכנה כל מדיניות שעבורה סדרת העלויות $(d_t(s_t, a_t))_{t\in\mathcal{T}}$ עומדת באילוץ פרציות זה בהסתברות 1, בתואר "קבילה" (feasible).

אנו רואים כי האילוץ כולל כ־$N^2/2$ אי־שוויונות עבור מערכת באופק זמן סופי (כאשר N הוא מספר צעדי הזמן), ואינסוף אי־שוויונות עבור מערכת באופק זמן אינסופי. כל אי־שוויון כזה כולל ערכי עלות מנקודות זמן שונות, כך שלא ניתן לבודד את האילוצים המוטלים על העלות בכל נקודת זמן בנפרד. על כל אי־השוויונות להתקיים, לכל מופע אפשרי של סדרת המצבים והפעולות, $(s_0, a_0, s_1, ...)$, תחת

המחקר בוצע בהנחייתו של פרופסור נחום שימקין, בפקולטה להנדסת חשמל.

תודתי נתונה לטכניון על תמיכתו הכלכלית הנדיבה בהשתלמותי.

תהליכי החלטה מרקוביים
עם אילוצי פרציות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים בהנדסת חשמל

מיכל גולן

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
כסלו התשע"ז חיפה דצמבר 2016

תהליכי החלטה מרקוביים
עם אילוצי פרציות

מיכל גולן