The Dissertation Committee for Jiaxiao Zheng
certifies that this is the approved version of the following dissertation:

Resource Sharing in Network Slicing and Human-Machine Interactions

Committee:

Gustavo de Veciana, Supervisor

François Baccelli

Sanjay Shakkottai

Constantine Caramanis

Thibaud Taillefumier

Resource Sharing in Network Slicing and Human-Machine Interactions

by

Jiaxiao Zheng

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2019

Dedicated to all who devote themselves in this great discovery.

Acknowledgments

After went through such an unforgettable journey, I would like to thank many people for their most selfless support. I could not imagine myself finally making it without them. First I would like to thank my advisor, Prof. Gustavo de Veciana, not only for his continuous assisting and pushing me to progress but also for setting a great example of researcher and instructor.

I would like to thank my dissertation committee members: Prof. François Baccelli, Prof. Constantine Caramanis, Prof. Sanjay Shakkottai, and Prof. Thibaud Taillefumier, for their advise and help in finishing this dissertation. I really appreciate their time and valuable comments on this thesis. Furthermore, Prof. Baccelli's fundamental work on Palm probability offers me a critical perspective to approach the problem described in this dissertation. Also, I want to thank several other professors for providing high-quality courses which are helpful in both my graduate research works and future career. I wish to thank my two mentors at Google, Dr. Bingjun Xiao and George Karagoulis, for their help and guidance during my intern at Google.

I would like to thank my fellow colleagues and labmates, including but not limited to Dr. Zheng Lu, Dr. Virag Shah, Dr. Yuhuan Du, Dr. Yicong Wang, Jean Abou Rahal, Jianhan Song and so on, for those insightful discussions and funny moments we had.

I also would like to thank my parents, for their wholeheartedly supporting me during and even prior to my education. Finally, I wish to express my most sincere gratitude and appreciation to my wife, Wenjun Xu, for her keeping me company during my good and bad days, and also her listening and supporting when I need it the most.

Resource Sharing in Network Slicing and Human-Machine Interactions

Publication No. _____

Jiaxiao Zheng, Ph.D.
The University of Texas at Austin, 2019

Supervisor: Gustavo de Veciana

In this thesis we explore two novel resource allocation models. The first addresses challenges associated with dynamic sharing of network resources by multiple tenants/services via network slicing. The second focuses on a data-driven approach to the optimization of resource allocation in interactive human-machine processes.

In our first thrust we investigate how to allocate shared storage, computation, and/or connectivity resources distributed amongst multiple tenants/virtual service providers which have dynamic loads. It is expected that next generation of wireless network will be shared by an increasing number of data-intensive mobile applications (e.g., autonomous cars, IoT, interactive 360° video streaming), and tenants/service providers. A key functional requirement for such infrastructure is enabling efficient sharing of heterogeneous resource among tenants/service providers supporting spatially varying and dynamic user demands, both from the point of view of enabling the deployment

and performance management to diverse service providers and/or tenants, as well as means to increase utilization and reduce CAPEX/OPEX associated with deploying possible new infrastructures.

To that end, we propose a novel dynamic resource sharing policy, namely, Share Constrained Proportional Fair (SCPF), which allocates a predefined 'share' of a pool of (distributed) resources to each slice. We provide a characterization of the achievable performance gains over General Processor Sharing (GPS), and Static Slicing (SS), i.e., fixed allocation of resources to slices. We also characterize the associated share dimensioning problem, asking when a particular set of load profiles and QoS requirements are feasible, as well as what should be an appropriate pricing strategy. We further consider possible slice-based admission control scheme where slices engage in an underlying game to maximize their carried loads subject to performance requirements.

In order to accommodate settings where one would wish to provision different types of resources which are coupled through user demands, we generalize SCPF to a more general resource allocation criterion, namely, Share Constrained Slicing (SCS), which extends traditional $\alpha-$fairness criterion, by striking a balance among inter- and intra-slice fairness vs. overall efficiency. We show that SCS has several desirable properties including slice-level *protection, envyfreeness,* and *load-driven elasticity.* In practice, mobile users' dynamics could make the cost of implementing SCS high, so we also study the feasibility of using a dynamically weighted max-min fair policy as a surrogate resource allocation scheme. For a setting with stochastic loads and elastic user

requirements, we model the user dynamics under SCS as a queuing network and establish the stability condition. Finally, and perhaps surprisingly, we show via extensive simulation that while SCS (and/or the surrogate weighted max-min allocation) provides inter-slice protection, they can also achieve improved job delay and/or perceived throughput, as compared to other weighted max-min based allocation schemes whose intra-slice weight allocation is not share-constrained, e.g., traditional max-min and/or discriminatory processor sharing.

In our second thrust we study how to optimize resource allocation in the context of human-machine interactions. Examples of such processes could include systems aimed at assisting humans in interactive learning, workload allocation, or web-search advertising. We devise an innovative framework to enable the optimization of a reward over an interactive process in a data-driven manner. This is a challenging problem for several reasons: (1) humans' behavior is not easily modeled and may reflect biases, memory and be sensitive to sequencing, all of which should/could be inferred from data; (2) because these interactions are typically sequential and transient, inferring such complex models for human behavior is difficult; (3) furthermore, in order to collect data on human-machine interactions one must choose a machine policy which in turn may bias inferences on human behavior. In this thesis we approach the problem of jointly estimating human behavior and optimizing machine policies via Alternating Entropy-Reward Ascent (AREA) algorithm. We characterize AREA in terms of its space and time complexity and convergence.

We also provide an initial validation based on synthetic data generated by an established noisy nonlinear model for human decision-making.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Resource sharing and allocation among network tenants and users are a critical element of the design and optimization of network performance. In this thesis we study two novel models of resource sharing and allocation. One is pertinent to the case where customers' distributions are dynamically changing across the network and are supported by multiple network tenants (i.e., over-the-top service providers or applications), which are sharing different types of resources distributed at the network 'edge'. The other is pertinent to the cases where customers and systems are involved in an interactive process, and the objective is to improve the associated utility.

1.1 Network Slicing

In our first research thrust we consider with network slicing where multiple tenants share wireless infrastructure via "slices" of resources customized to specific mobile services needs, e.g., mobile broadband, media, OTT service providers, and machine-type communications. Customization of network slices may include the allocation of (virtualized) resources (communication/computation), per-slice policies, performance monitoring and manage-

ment, security, accounting, etc. The ability to deploy service-specific slices is viewed, not only as means to meet the diverse and sometimes stringent demands of emerging services, e.g., vehicular communications, augmented reality, but also as an approach for infrastructure providers to reduce their costs while developing revenue streams. The ideal end goal in this setting would be to reproduce the success of cloud compute/storage providers in the context of providing virtualized wireless connectivity. Resource allocation in this context is, however, more challenging than for traditional cloud computing. Indeed, rather than drawing on a centralized pool of resources, a network slice requires allocations across a distributed pool of resources, e.g., base stations.

The sharing of spectrum and/or network infrastructure is viewed as one way of reducing capital/operational costs and is already being considered by standardization bodies, see [1, 2], which have specified architectural and technical requirements, but left the sharing criteria and algorithmic issues open. By aggregating network slices' traffic on shared resources, it is expected that operators could realize substantial savings, that might justify/enable new shared investments in next generation technologies including 5G, mmWave and massive MIMO. In this thesis we propose a novel sharing scheme, namely, Share Constrained Proportionally Fair (SCPF), which adjusts resource allocations according to the dynamic customer loads that each network operator has across the network resources and a pre-allocated share of the pool. The approach is studied from various perspectives, and is shown to outperform traditional approaches such as Static Slicing, and Generalized Processor Sharing

in a wide range of conditions.

As the wireless communication technology evolves, next generation networks also seek to support a variety of data-intensive services and applications which share different types of resources. Coupling of provisioning across heterogeneous resources raises new challenges to our resource sharing scheme by limiting its flexibility in aligning resource allocations to the spatial variations of each resource type. To accommodate this, we propose a more general slice-based resource allocation criterion, namely, Share Constrained Slicing (SCS), which extends traditional $\alpha-$fairness criterion to a setting where slices are assigned shares of resource pools. We demonstrate that SCS succeeds in providing *slice level protection, envyfreeness*, and *load-driven elasticity*, and it also allows a low-complexity surrogate allocation policy based on (dynamically weighted) max-min fair. SCS, and/or the surrogate max-min fair, is shown to outperform prevailing sharing criterion, e.g., Dominant Resource Fair [3] and Discriminatory Processor Sharing [4, 5] in many settings.

Part I is devoted to exploring resource allocation in a network slicing context and introduce a variety of new results in this area. Part of the results are available in [6], [7], [8] and [9].

1.2 Human-Machine Interactions

In our second thrust we turn from optimizing dynamic sharing process in networking system to the optimization of interactive processes associated with human-machine interactions. For example, in a sequential web search

setting, the placement and timing of advertisements and/or information may attract customers' attention and can potentially influence their decisions, i.e., impact the convergence process to a decision; in an educational context, the sequencing of learning material and/or problems may significantly affect students' experience and the final learning outcomes. Because the nature of human decision making/learning process may be not available to the system, it is desirable to allow the system to try different allocation schemes iteratively and make adjustments based on the observed outcomes. The adjustment of the system, together with customers' responses, form an interactive process, where the machine can influence the human over time based on his/her responses to date. The optimization of such process is challenging in many ways. First, humans' decision-making processes are complex, biased and typically transient, i.e., humans' preferences might change over time. Thus the customers' decisions might be highly dependent on previous interactions. Also, the sequential nature of the interactive process naturally leads to exponential complexity. In this thesis, we use a data-driven approach to solve such problem. We propose an alternating maximization algorithm, which combines an estimation phase based on the maximum entropy principle for interactive processes, and an optimization phase, which modifies the machine behavior/resource allocation given the estimated model for human behavior/responses. Its performance, complexity, and convergence properties are discussed.

Part II is devoted to presenting this framework and introducing our results in this area. Part of the results are available in [10], and is under

preparation to submit to Allerton Conference on Communication, Control, and Computing.

Part I

Resource Sharing in Network Slicing

Chapter 2

Network Slicing with Parallel Resources

2.1 Introduction

2.1.1 Background and Motivation

Next generation wireless systems are expected to embrace SDN/NFV technologies towards realizing slices of shared wireless infrastructure which are customized for specific mobile services, e.g., mobile broadband, media, OTT service providers, and machine-type communications. Customization of network slices may include allocation of (virtualized) resources (communication/computation), per-slice policies, performance monitoring and management, security, accounting, etc. The ability to deploy service specific slices is viewed, not only as means to meet the diverse and sometimes stringent demands of emerging services, e.g., vehicular, augmented reality, but also as

This chapter was partially included in the following two papers. J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs, Statistical multiplexing and traffic shaping games for network slicing, in proceeding of *WiOpt'17*, May 2017, and J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs, Statistical multiplexing and traffic shaping games for network slicing (extended), *IEEE/ACM Trans. on Networking*, Dec. 2018. The author was responsible for the major part of developing those analytic results, conducting simulation-based evaluation, and writing the paper.

an approach for infrastructure providers to reduce costs while developing new revenue streams. Resource allocation virtualization in this context is more challenging than for traditional cloud computing. Indeed, rather than drawing on a centralized pool of resources, a network slice requires allocations across a distributed pool of resources, e.g., base stations. The challenge is thus to promote efficient *statistical multiplexing* amongst slices over pools of shared resources.

Network slices can be used to enable the sharing of network resources amongst competing (possibly virtual) operators. Indeed, the sharing of spectrum and infrastructure is viewed as one way of reducing capital/operational costs and is already being considered by standardization bodies, see [1, 2], which have specified architectural and technical requirements, but left the sharing criteria and algorithmic issues open. By aggregating their traffic onto shared resources, it is expected that operators could realize substantial savings, which might justify/enable new shared investments in next generation technologies including 5G, mmWave and massive MIMO.

The focus of this chapter is on resource sharing amongst slices supporting stochastic (mobile) loads. A natural approach to sharing is complete partitioning (see, e.g., [11]), which we refer to as *static slicing*, whereby resources are statically partitioned and allocated to slices, according to a service level agreement, irrespective of slices' instantaneous loads. This offers each slice a guaranteed allocation at each base station, and protection from each other's traffic, but, as we will see, poor efficiency. Other approaches include

full sharing (where all slices are served on a FCFS basis without resource reservation), generalized processor sharing [12], which pre-assigns a share to each slice, and allocate resource at each base station proportionally to the shares among the slices which have active users. Instead, we advocate an alternative approach wherein each slice is pre-assigned a *fixed* share of the pool of resources, and re-distributes its share equally amongst its active customers. In turn, each base station allocates resources to customers in proportion to their shares. We refer to this sharing model as Share Constrained Proportionally Fair (SCPF) resource allocation. By contrast with static slicing, SCPF is *dynamic* (since its resource allocations depend on the network state) but constrained by the network slices' pre-assigned shares (which provides a degree of protection amongst slices).

2.1.2 Related Work

There is an enormous amount of related work on network resource sharing in the engineering, computer science and economics communities. The standard framework used in the design and analysis of communication networks is utility maximization (see e.g., [13] and references therein) which has led to the design of several transport and scheduling mechanisms and criteria, e.g., the often considered proportional fair criterion. The SCPF mechanism, described above, should be viewed as a Fisher market where agents (slices), which are share (budget) constrained, bid on network resources, see, e.g., [14], and for applications [15, 16, 17]. The choice to re-distribute a slice's share

(budget) equally amongst its users, can be viewed as a network mandated policy, but also emerges naturally as the social optimal, market and Nash equilibrium when slices exhibit (price taking) strategic behavior in optimizing their own utility, see [18].

The novelty of our work lies in considering slice based sharing, under stochastic loads and in particular studying the expected performance resulting from such SCPF-based resource allocations among coupled slices. Other researchers who have considered performance of stochastic networks, e.g., [19, 20], and others, have studied networks where customers are allocated resources (along routes) based on maximizing a sum of customers utilities. These works focus on network stability for 'elastic' customers, e.g., file transfers. Subsequently [21, 22] extended this line of work, to the evaluation of mean file delays, but only under balanced fair resource allocations (as a proxy for proportional fairness). Our focus here is on SCPF-based sharing amongst slices with stochastic loads and on 'inelastic' or 'rate-adaptive' customers, e.g., video, voice, and more generally customers on properly provisioned networks, whose activity on the network can be assumed to be independent of their resource allocations.

Finally there is much ongoing work on developing the network slicing concept, see e.g., [23, 24] and references therein, including development of approaches to network virtualization in RAN architectures, e.g, [25, 26, 27], and SDN-based implementation, e.g., [28]. This chapter focuses on devising good slice-based resource sharing criteria to be incorporated into such architectures.

2.1.3 Contributions

This thesis makes several contributions centering on a simple and practical resource sharing mechanism: SCPF. First, this thesis considers user performance (bit transmission delay) on slices supporting stochastic loads. In particular it develops expressions for (i) the mean performance seen by a typical user on a network slice; and (ii) the achievable performance gains versus Static Slicing (SS) and Generalized Processor Sharing (GPS). We show that when a slice's load is more 'imbalanced' than, and/or 'orthogonal' to, the aggregate network load, one will see higher performance gains. The analysis of this thesis provides an insightful picture of the 'geometry' of statistical multiplexing for SCPF-based network slicing. Second, under SCPF, traditional network dimensioning translates to a coupled share dimensioning problem, which addresses whether there exist feasible share allocations given slices' expected loads and performance requirements. We provide a solution to robust share dimensioning for SCPF-based network slicing. We further develop some understanding regarding how one should price shared resources amongst slices with heterogeneous traffic profiles.

Third, we consider decentralized per-slice performance management under SCPF sharing. In particular, we consider admission control aimed at maximizing a slice's carried load subject to a performance constraint. When slices unilaterally optimize their admission control policies, the coupling of their decisions can be viewed as a 'traffic shaping' game, which is shown to have a Nash equilibrium. For a high load regime we explicitly characterize the equi-

librium and the associated gains in carried load for SCPF versus static slicing. Finally, we present detailed simulations for a shared distributed infrastructure supporting slices with mobility patterns different than that assumed in the theoretical analysis and more practical SINR model. The results match our analysis well, which further supports our conclusions on gains in both performance and carried loads of SCPF sharing.

2.2 Model and Performance Analysis

2.2.1 Network Slices, Resources and Mobile Service Traffic

We consider a collection of base stations (sectors) $\mathcal{B}$ shared by a set of network slices $\mathcal{V}$, with cardinalities B and V respectively. For example, $\mathcal{V}$ might denote slices supporting different services or (virtual) mobile operators, etc.

We envisage each slice v providing a mobile service in the region served by the base stations $\mathcal{B}$ –generalizations to subsets of base stations are natural. Each slice supports a stochastic load of users (devices/customers) with an associated mobility/handoff policy. In particular, we assume that exogenous arrivals to slice v at base station b follow a Poisson process with intensity γ_b^v and let $\boldsymbol{\gamma}^v$ denote the (column) vector of arrival intensities at each base station associated with slice v, i.e. $\boldsymbol{\gamma}^v = (\gamma_b^v : b \in \mathcal{B})$. Each slice v customer at base station b has an independent sojourn time with mean μ_b^v after which it is randomly routed to another base station or exits the system. As explained below we assume that such mobility patterns do not depend on the resources allocated to users. We let $\boldsymbol{Q}^v = (q_{i,j}^v : i, j \in \mathcal{B})$ denote a slice-dependent routing matrix where $q_{i,j}^v$ is the probability a slice v customer moves from base station i to j and $1 - \sum_{j \in \mathcal{B}} q_{i,j}^v$ is the probability it exits the system.

This model induces an overall traffic intensity for slice v across base stations satisfying flow conservation equations: for all $b \in \mathcal{B}$ we have $\kappa_b^v = \gamma_b^v + \sum_{a \in \mathcal{B}} \kappa_a^v q_{a,b}^v$, where κ_b^v is the traffic intensity of slice v on base station b. Accounting for users' sojourn times, the mean offered load of slice v on

base station b is $\rho_b^v = \kappa_b^v \mu_b^v$, and $\boldsymbol{\rho}^v := (\rho_b^v : b \in \mathcal{B})$ captures its system load distribution. Letting $\boldsymbol{\mu}^v = (\mu_b^v : b \in \mathcal{B})$, the flow conservation equations can be rewritten in matrix form as:

$$\boldsymbol{\rho}^v = \text{diag}(\boldsymbol{\mu}^v)(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)^{-1}\boldsymbol{\gamma}^v. \tag{2.1}$$

If $\boldsymbol{Q}^v$ is irreducible, $\boldsymbol{I} - (\boldsymbol{Q}^v)^T$ is irreducibly diagonally dominant thus always invertible. Otherwise, we can always find a permutation matrix of $\mathcal{B}$, say $\boldsymbol{P}$ to make:

$$\boldsymbol{P}^T(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{B}_{1,2} & \dots \\ & \ddots & \vdots \\ & & \boldsymbol{A}_K \end{bmatrix},$$

where K is the number of irreducible classes. Moreover, at least one base station of each irreducible class has a nonzero exiting probability, thus $\boldsymbol{A}_K$ must be invertible. Then the invertibility of $\boldsymbol{I} - (\boldsymbol{Q}^v)^T$ follows.

This model corresponds to a multi-class network of $M/GI/\infty$ queues (base stations), where each slice corresponds to a class of customers, see, e.g.,[29]. Such networks are known to have a *product-form stationary* distribution, i.e., the numbers of customers on slice v at base station b, denoted by N_b^v, are mutually independent and $N_b^v \sim \text{Poisson}(\rho_b^v)$. Since the sum of independent Poisson random variables is again Poisson, the total number of customers on slice v is such that $N^v = \sum_{b\in\mathcal{B}} N_b^v \sim \text{Poisson}(\rho^v)$ where $\rho^v := \sum_{b\in\mathcal{B}} \rho_b^v$.

Our network model for the numbers of customers and mobility across base stations, assumes that customer sojourn/activity/mobility are independent of the network state and of the resources a customer is allocated. This

is reasonable for properly engineered slices where the performance a customer sees does not impact its activity, e.g., *inelastic* or *rate-insensitive* applications seeing acceptable performance. This covers a wide range of applications including voice, video streaming, IoT monitoring, real-time control, and even, to some degree, elastic web browsing sessions where users are peak rate constrained and this constraint typically dictates their performance.

There are several natural generalizations to this model including class-based routing and user sessions (e.g. web browsing) which are not always active at the base stations they visit, see, e.g., [29].

2.2.1.1 Network Slice Resource Sharing

In the sequel we consider a setting where the resources allocated to a slice's customers depend on the overall network state, i.e., number of customers each slice has on each base station, corresponding to the stochastic process described in Section 2.2.1. Let us consider a *snapshot* of the system's state and let $\mathcal{U}_b^v, \mathcal{U}_b, \mathcal{U}^v$, and $\mathcal{U}$ denote sets of active customers on slice v at base station b, at base station b, on slice v, and on the overall network, respectively. Thus, the cardinalities of these sets correspond to a realization of the system 'state', i.e., $|\mathcal{U}_b^v| = n_b^v$ and $|\mathcal{U}^v| = n^v$, where in a stationary regime n^v and n_b^v are realizations of Poisson random variables N^v and N_b^v, respectively.

Each base station b is modeled as a finite resource shared by its associated users $\mathcal{U}_b$. A customer $u \in \mathcal{U}_b$ can be allocated a fraction $f_u \in [0, 1]$ of that resource, e.g., of resource blocks in a given LTE frame, or allocated the

resource for a fraction of time, where $\sum_{u \in \mathcal{U}_b} f_u = 1$. We shall neglect quantization effects. The transmission rate to customer u, denoted by r_u, is then given by $r_u = f_u c_u$ where c_u denotes the current peak rate for that user. To model customer heterogeneity across slices/base stations we shall assume c_u for a typical customer on slice v at base station b is an independent realization of a random variable, denoted by C_b^v, whose distribution may depend on the slice, since slices may support different types of customer devices (e.g., car connectivity vs. mobile phone) and depend on the base station, since typical slice v users may have different spatial distributions with respect to base station b or see different levels of interference.

Below we consider three resource allocation schemes; the first two are used as benchmarks, while the third is the one under study in this chapter. For all we assume each slice is allocated a 'share' of the network resources $s^v, v \in \mathcal{V}$ such that $s^v > 0$ and $\sum_{v \in \mathcal{V}} s^v = 1$.

Definition 2.2.1. Static Slicing (SS): Under SS, slice v is allocated a fixed fraction s^v of each base station b's resources, and each customer $u \in \mathcal{U}_b^v$ gets an equal share, i.e., $1/n_b^v$, of the slice v's resources at base station b. Thus the users transmission rate r_u^{SS} is given by $r_u^{SS} = \frac{s^v}{n_b^v} c_u$.

Definition 2.2.2. Generalized Processor Sharing (GPS): [12] Under GPS, each active slice v at base station b such that $n_b^v > 0$ is allocated a fraction of the base station b's resources proportionally to its share s^v. Thus

a user $u \in \mathcal{U}_b^v$ sees a transmission rate r_u^{GPS} given by

$$r_u^{GPS} = \frac{s^v}{n_b^v \sum_{v' \in \mathcal{V}} s^{v'} \mathbf{1}_{\{n_b^{v'} > 0\}}} c_u. \tag{2.2}$$

Definition 2.2.3. Share Constrained Proportionally Fair (SCPF): Under SCPF each slice re-distributes its share of the overall network resources equally amongst its active customers, which thus get a weight (sub-share) $w_u = \frac{s^v}{n_v}$ for $u \in \mathcal{U}^v, \forall v \in \mathcal{V}$. In turn, each base station allocates resources to customers in proportion to their weights. So a user $u \in \mathcal{U}_b^v$ gets a transmission rate r_u^{SCPF} given by

$$r_u^{SCPF} = \frac{w_u}{\sum_{u' \in \mathcal{U}_b} w_{u'}} c_u = \frac{\frac{s^v}{n^v}}{\sum_{v' \in \mathcal{V}} \frac{n_b^{v'} s^{v'}}{n^{v'}}} c_u. \tag{2.3}$$

A simple example illustrating the differences among three schemes is as follows. Suppose there are two base stations, i.e., $\mathcal{B} = \{b_1, b_2\}$, and two slices $\mathcal{V} = \{1, 2\}$ each with an equal share of the network resource. Consider a snapshot of the system where Users u_1, u_2 are on Slice 1 and u_3, u_4 are on Slice 2. Also, u_1, u_2, and u_3 are at base station b_1 and u_4 is at base station b_2. Let us assume for simplicity that $c_u = 1, \forall u \in \mathcal{U}$. In this case, under SS at b_1 the two users on Slice 1 need to share $\frac{1}{2}$ of the resource while u_3 on Slice 2 is allocated the other $\frac{1}{2}$, while at b_2, half of the resource is wasted due to the absence of active users on Slice 1. By contrast, GPS utilizes all resources at b_2 by allocating all of them to u_4, and it makes the same allocation as SS at b_1. Under SCPF, because each user is allocated the same weight $\frac{1}{4}$, at b_1, three users are allocated the same rate $\frac{1}{3}$ and at b_2 all bandwidth is given to

u_4. This example shows how SCPF achieves better network-wide fairness than GPS and SS, while ensuring that resources are not wasted.

Indeed, under SCPF the *overall* fraction of resources slice v is allocated at base station b is proportional to $\frac{n_b^v}{n^v}s^v$, i.e., its *share* times its *relative* number of users at the base station. This provides a degree of *elasticity* to variations in the slice's spatial loads. However, if a slice has a large number of customers, its customers' weights are proportionally decreased, which protects other slices from such overloads. Note that SCPF requires minimal information exchanges among base stations and is straightforward to implement, e.g., using SDN-like framework. In addition, as mentioned in Section 2.1, SCPF resource allocations are *socially optimal* for certain types of budget-constrained Fisher Markets.

Theorem 2.2.1. *When base stations allocate resources to their associated users proportionally to users' weights (sub-shares), equal weight (sub-share) assignments are socially optimal in the sense that they maximize the following surrogate network utility, which is a weighted sum of logs of user perceived transmission rates.*

$$\begin{aligned} \max_{\boldsymbol{w}\succeq\mathbf{0}} \quad & \sum_{v\in\mathcal{V}} \frac{s^v}{|\mathcal{U}^v|} \sum_{u\in\mathcal{U}^v} \log(r_u) \\ \textit{such that:} \quad & s^v = \sum_{u\in\mathcal{U}^v} w_u \quad \forall v \in \mathcal{V} \\ & r_u = \frac{w_u}{\sum_{u'\in\mathcal{U}_b} w_{u'}} c_u \quad \forall u \in \mathcal{U}. \end{aligned} \tag{2.4}$$

Alternatively, the above optimization (and thus resource allocation) can

be recognized as the Eisenberg Gale program characterizing the equilibrium of a Fisher market and thus the allocations reached under a variety of response dynamics, see e.g., [30].

Proof. A $\boldsymbol{w}$ induces a probability distribution at resource b as:

$$\boldsymbol{p}_b(\boldsymbol{w}) = \left(p_u(\boldsymbol{w}) \triangleq \frac{w_u}{\sum_{u' \in \mathcal{U}_b} w_{u'}} : u \in \mathcal{U}_b \right),$$

Suppose the equal weight allocation is $\boldsymbol{w}^* = \left(w_u^* = \frac{s^v}{|\mathcal{U}^v|} : u \in \mathcal{U} \right)$, where $u \in \mathcal{U}^v$. Let us define the objective function of Problem (2.4) as $U(\boldsymbol{w})$. For an arbitrary $\boldsymbol{w}$ we have:

$$\begin{aligned}
U(\boldsymbol{w}^*) - U(\boldsymbol{w}) &= \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{U}^v} \frac{s^v}{|\mathcal{U}^v|} \left(\log \left(\frac{w_u^* c_u}{\sum_{u' \in \mathcal{U}_b} w_{u'}^*} \right) - \log \left(\frac{w_u c_u}{\sum_{u' \in \mathcal{U}_b} w_{u'}} \right) \right) \\
&= \sum_{b \in \mathcal{B}} \left(\sum_{u' \in \mathcal{U}_b} w_{u'}^* \right) \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{U}_b^v} p_u(\boldsymbol{w}^*) \log \left(\frac{p_u(\boldsymbol{w}^*)}{p_u(\boldsymbol{w})} \right) \\
&= \sum_{b \in \mathcal{B}} \left(\sum_{u' \in \mathcal{U}_b} w_{u'}^* \right) D\left(\boldsymbol{p}_b(\boldsymbol{w}^*) || \boldsymbol{p}_b(\boldsymbol{w}) \right) \\
&\geq 0,
\end{aligned}$$

where $D\left(\boldsymbol{p}_b(\boldsymbol{w}^*) || \boldsymbol{p}_b(\boldsymbol{w}) \right)$ denotes the Kullback-Leibler divergence between $\boldsymbol{p}_b(\boldsymbol{w}^*)$ and $\boldsymbol{p}_b(\boldsymbol{w})$. The second equality holds true by repartitioning users by the base stations they are associated with. The last inequality comes from the nonnegativity of the Kullback-Leibler divergence. Thus $\boldsymbol{w}^*$ has a higher utility than any other arbitrary weight allocation thus is optimal. □

2.2.2 Performance Evaluation

In this section we study the expected performance seen by a slice's typical customer. Given our focus on inelastic/rate adaptive traffic and tractability, we choose our customer performance metric as the reciprocal transmission rate, referred to as the *Bit Transmission Delay (BTD)*, see, e.g.,[31]. This corresponds to the time taken to transmit a 'bit', so lower BTDs indicate higher rates and thus better performances. BTD is a high-level metric capturing the instantaneous QoS perceived by a user, e.g., short packet transmission delays are roughly proportional to the BTD. By guaranteeing a good BTD we can guarantee that the user perceived QoS is acceptable all the time, instead of in an average sense. Alternatively, the negative of the BTD can be viewed as a concave utility function of the rate, which in the literature (see, e.g., [32]) was referred to as the *potential delay* utility. Concave utility functions tend to favor allocations that exhibit reduced variability in a stochastic setting. Given the stochastic loads on the network, we shall evaluate the average BTD seen by a typical (i.e., randomly selected) customer on a slice, i.e., averaged over the stationary distribution of the network state and transmission capacity seen by typical users, e.g., C_b^v, at each base station. Such averages naturally place higher weights on congested base stations, where a slice may have more users, best reflecting the overall performance customers will see.

2.2.2.1 Analysis of BTD Performance

Consider a *typical* customer on slice v and let $\mathbb{E}^v$ denote the expectation of the system state as seen by such a customer, i.e., under the Palm distribution [33].

For SCPF, we let R^v be a random variable denoting the rate of a typical customer on slice v, and R_b^v that of such customer on slice v at base station b. Similarly, let $R^{v,SS}$, $R_b^{v,SS}$, $R^{v,GPS}$, and $R_b^{v,GPS}$ denote these quantities under SS and GPS, respectively. Thus, under SCPF the average BTD for a typical slice v customer is given by $\mathbb{E}^v[\frac{1}{R^v}]$. The next result characterizes the mean BTD under SCPF, SS, and GPS under our traffic model. We introduce some further notation in Table 2.1.

We use $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_{\boldsymbol{M}} := \boldsymbol{x}_1^T \boldsymbol{M} \boldsymbol{x}_2$ to denote the weighted inner product of vectors, where $\boldsymbol{M}$ is a diagonal matrix. Also, we use $\|\boldsymbol{x}\|_{\boldsymbol{M}} := \sqrt{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}$ to denote the weighted norm of a vector, where $\boldsymbol{M}$ is a diagonal matrix. In both cases, when $\boldsymbol{M}$ is the identity matrix $\boldsymbol{I}$ we simply omit it. In addition, $\|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_1$ denote the L2-norm and L1-norm of $\boldsymbol{x}$, respectively.

Theorem 2.2.2. *For network slicing based on SCPF, the mean BTD for a typical customer on slice v is given by*

$$\mathbb{E}^v\left[\frac{1}{R^v}\right] = \sum_{b\in\mathcal{B}} \tilde{\rho}_b^v \delta_b^v \left(1 - \tilde{\rho}_b^v + (\rho^v + 1)\left(\frac{\tilde{g}_b'}{s^v} + e^{-\rho^v}\tilde{\rho}_b^v\right)\right). \tag{2.5}$$

If $(\tilde{\boldsymbol{\rho}}^v : v \in \mathcal{V})$ are fixed, and $(\rho^v : v \in \mathcal{V})$ are large, then the mean BTD has

Table 2.1: Key notation used in Chapter 2.

Notation	Definition	Interpretation
ρ^v	$\mathbb{E}[N^v]$	Overall load of slice v.
$\boldsymbol{\rho}^v$	$(\rho_b^v := \mathbb{E}[N_b^v] : b \in \mathcal{B})$	Load distribution of slice v.
$\tilde{\boldsymbol{\rho}}^v$	$\left(\tilde{\rho}_b^v := \frac{\rho_b^v}{\rho^v} : b \in \mathcal{B}\right)$	Relative load distribution of slice v.
$\tilde{\boldsymbol{g}}$	$\left(\tilde{g}_b := \sum_{v \in \mathcal{V}} s^v \tilde{\rho}_b^v : b \in \mathcal{B}\right)$	Overall share weighted relative load distribution.
$\tilde{\boldsymbol{g}}'$	$(\tilde{g}'_b : b \in \mathcal{B})$, where $\tilde{g}'_b = \sum_{v \in \mathcal{V}} s^v (1 - e^{-\rho^v}) \tilde{\rho}_b^v$.	Overall active share weighted relative load distribution, i.e., weighted by probability of a slice being active $(1 - e^{-\rho^v})$.
$\bar{\boldsymbol{s}}^v$	$(\bar{s}_b^v : b \in \mathcal{B})$, where $\bar{s}_b^v = \mathbb{E}^v\left[\sum_{v' \neq v} s^{v'} \mathbf{1}_{\{N_b^{v'}=0\}}\right] = \sum_{v' \neq v} s^{v'} e^{-\rho_b^{v'}}$.	Average idle share distribution seen by a typical user on slice v.
$\boldsymbol{\delta}^v$	$\left(\delta_b^v := \mathbb{E}^v\left[\frac{1}{C_b^v}\right] : b \in \mathcal{B}\right)$.	Mean reciprocal capacity of slice v at each base station.
$\boldsymbol{\Delta}_v$	$\mathrm{diag}(\boldsymbol{\delta}^v)$	Diagonal matrix of mean reciprocal capacity of slice v.

the following asymptotic form:

$$\mathbb{E}^v\left[\frac{1}{R^v}\right] \approxeq \frac{\rho^v}{s^v} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{g}} \rangle_{\boldsymbol{\Delta}^v} + O(1). \tag{2.6}$$

For network slicing based on SS, the mean BTD for a typical customer on slice v is given by

$$\mathbb{E}^v\left[\frac{1}{R^{v,SS}}\right] = \sum_{b \in \mathcal{B}} \tilde{\rho}_b^v \delta_b^v \left(\frac{\rho_b^v + 1}{s^v}\right). \tag{2.7}$$

For network slicing based on GPS, the mean BTD for a typical customer on slice v is given by

$$\mathbb{E}^v\left[\frac{1}{R^{v,GPS}}\right] = \sum_{b \in \mathcal{B}} \tilde{\rho}_b^v \delta_b^v \left(\frac{\rho_b^v + 1}{s^v}\right)(1 - \bar{s}_b^v). \tag{2.8}$$

Remark: BTD under all 3 schemes increases with the overall load ρ^v and decreases with the share s^v when $(\tilde{\boldsymbol{\rho}}^v : \ v \in \mathcal{V})$ are fixed. Their dependencies on relative loads $(\tilde{\boldsymbol{\rho}}^v : \ v \in \mathcal{V})$ are different, implying that they exploit statistical multiplexing differently.

Proof. Recall that Poisson arrivals see time averages, i.e., see the remaining users in the product-form stationary distribution, given in Section 2.2.1. Thus the distribution as seen by a typical user on slice v at base station b is the same as the product-form distribution *plus an additional customer* on slice v at base station b. Using this fact and SCPF resource allocations as given by Eq. (2.3), the mean BTD of a typical slice v user at base station b can be expressed as follows:

$$\begin{aligned}
\mathbb{E}^v\left[\frac{1}{R_b^v}\right] &= \mathbb{E}^v\left[\frac{1}{C_b^v}\right]\mathbb{E}\left[\frac{s^v\frac{N_b^v+1}{N^v+1}+\sum_{v'\neq v}\frac{s^{v'}N_b^{v'}}{N^{v'}}\mathbf{1}_{\{N^{v'}>0\}}}{\frac{s^v}{(N^v+1)}}\right] \\
&= \delta_b^v\mathbb{E}\left[(N_b^v+1)+\frac{N^v+1}{s^v}\sum_{v'\neq v}\frac{s^{v'}N_b^{v'}}{N^{v'}}\mathbf{1}_{\{N^{v'}>0\}}\right] \\
&= \delta_b^v\left(\rho_b^v+1+\frac{\rho^v+1}{s^v}\sum_{v'\neq v}s^{v'}(1-e^{\rho^{v'}})\tilde{\rho}_b^{v'}\right) \\
&= \delta_b^v\left(1-\tilde{\rho}_b^v+\frac{(\rho^v+1)}{s^v}\tilde{g}_b'+(\rho^v+1)e^{-\rho^v}\tilde{\rho}_b^v\right),
\end{aligned}$$

where the second equality follows by noticing that (i) N^v is independent of $N_b^{v'}$ and $N^{v'}$ and (ii) $E\left[\frac{N_b^{v'}}{N^{v'}}\mathbf{1}_{\{N^{v'}>0\}}\right] = P(N^{v'}>0)E\left[\frac{N_b^{v'}}{N^{v'}}\middle| N^{v'}>0\right] = \frac{\rho_b^{v'}}{\rho^{v'}}P(N^{v'}>0)$. The latter result is given by the following lemma:

Lemma 2.2.3. *If $N_1, N_2, \ldots, N_n$ are independent Poisson random variables, such that $N_i \sim Poisson(\rho_i), i = 1, 2, \ldots, n$. Then for all i we have that:*

$$\mathbb{E}\left[\frac{N_i}{\sum_{j=1}^n N_j}\middle|\sum_{j=1}^n N_j > 0\right] = \frac{\rho_i}{\sum_{j=1}^n \rho_j}.$$

Proof. Suppose all ρ_i's are rational, and for some ϵ small enough, for all i, we have that $m_i = \frac{\rho_i}{\epsilon}$ is integer valued. Let $X_{i,j}, i = 1, \ldots, n, j = 1, \ldots, m_i$ be i.i.d. Poisson random variables with parameter ϵ. Since Poisson random variables are infinitely divisible we have that $N_i \sim \sum_{j=1}^{m_i} X_{i,j}$. Then

$$\begin{aligned}\mathbb{E}\left[\frac{N_i}{\sum_{j=1}^n N_j}\middle|\sum_{j=1}^n N_j > 0\right] &= \mathbb{E}\left[\frac{\sum_{j=1}^{m_i} X_{i,j}}{\sum_{i=1}^n \sum_{j=1}^{m_i} X_{i,j}}\middle|\sum_{j=1}^n N_j > 0\right] \\ &= m_i \cdot \mathbb{E}\left[\frac{X_{i,1}}{\sum_{i=1}^n \sum_{j=1}^{m_i} X_{i,j}}\middle|\sum_{j=1}^n N_j > 0\right] = \frac{m_i}{\sum_{j=1}^n m_j} = \frac{\rho_i}{\sum_{j=1}^n \rho_j},\end{aligned}$$

where the second and third equalities follow from the symmetry among $X_{i,j}$. Since the conditional expectation will be a continuous function of the parameter vector $\boldsymbol{\rho}$, the equality follows more generally for the case where $\boldsymbol{\rho}$ is a real valued vector. □

The asymptotic form given in Eq. (2.6) follows by noting that when ρ^v is large for all $v \in \mathcal{V}$, $\tilde{\boldsymbol{g}}' \approxeq \tilde{\boldsymbol{g}}$, and only the term scaling with ρ^v matters.

Under static slicing we have that for a typical user on slice v at base station b,

$$\mathbb{E}^v\left[\frac{1}{R_b^{v,SS}}\right] = \mathbb{E}^v\left[\frac{1}{C_b^v}\right]\mathbb{E}\left[\frac{N_b^v + 1}{s^v}\right] = \delta_b^v \frac{\rho_b^v + 1}{s^v}.$$

Similarly, under GPS we have that,

$$\begin{aligned}\mathbb{E}^v\left[\frac{1}{R^{v,GPS}}\right] &= \mathbb{E}^v\left[\frac{1}{C_b^v}\right]\mathbb{E}^v\left[\frac{N_b^v\sum_{v'\in\mathcal{V}}s^{v'}\mathbf{1}_{\{N_b^{v'}>0\}}}{s^v}\right]\\ &= \delta_b^v\left(\frac{\rho_b^v+1}{s^v}\right)(1-\bar{s}_b^v)\,.\end{aligned}$$

The theorem follows by taking a weighted average across base stations – weighted by the fraction of customers at each base station, i.e., $\tilde{\rho}_b^v$. □

2.2.2.2 Analysis of Gain

Using the results in Theorem 2.2.2 one can evaluate the gains in the mean BTD for a typical slice v user under SCPF vs. SS, defined as,

$$G_v^{SS} := \frac{\mathbb{E}^v\left[\frac{1}{R^{v,SS}}\right]}{\mathbb{E}^v\left[\frac{1}{R^v}\right]}.$$

In general, one would expect $G_v^{SS} \geq 1$ since under SCPF typical users should see higher allocated rates and thus lower BTDs. One can verify that is the case when slices have *uniform* loads across base stations but the general case is more subtle. Similarly, we define the gain of SCPF vs. GPS by

$$G_v^{GPS} := \frac{\mathbb{E}^v\left[\frac{1}{R^{v,GPS}}\right]}{\mathbb{E}^v\left[\frac{1}{R^v}\right]}.$$

By taking the ratio of the mean BTD perceived by a typical customer under SS and that under SCPF given in Theorem 2.2.2, we have the following corollary.

Corollary 2.2.4. *The BTD gain of SCPF over SS for slice v is given by*

$$G_v^{SS} = \frac{\rho^v\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} + \langle\boldsymbol{\delta}^v,\tilde{\boldsymbol{\rho}}^v\rangle}{s^v\langle\boldsymbol{\delta}^v,\tilde{\boldsymbol{\rho}}^v\rangle - s^v\left(1-(\rho^v+1)e^{-\rho^v}\right)\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} + (\rho^v+1)\langle\tilde{\boldsymbol{g}}',\tilde{\boldsymbol{\rho}}^v\rangle_{\boldsymbol{\Delta}^v}}. \quad (2.9)$$

For fixed relative loads $(\tilde{\boldsymbol{\rho}}^v : v \in \mathcal{V})$*, when slice* v *has a light load, i.e.,* $\rho^v \to 0$*, the gain is greater than 1 and given by:*

$$G_v^{SS,L} = \frac{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle}{s^v \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle + \langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}} > 1.$$

Furthermore, G_v^{SS} *is a nonincreasing function of* ρ^v*, and if all slices have high overall loads, i.e.,* $\rho^v \to \infty, \forall v \in \mathcal{V}$*, the gain is given by:*

$$G_v^{SS,H} = \frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}.$$

Proof. From Theorem 2.2.2 we have that for SCPF

$$\begin{aligned} \mathbb{E}^v \left[\frac{1}{R^v} \right] &= \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle - \left(1 - (\rho^v + 1)e^{-\rho^v}\right) \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} \\ &\quad + \frac{\rho^v + 1}{s^v} \langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}, \end{aligned} \tag{2.10}$$

while for SS we have that

$$\mathbb{E}^v \left[\frac{1}{R^{v,SS}} \right] = \frac{1}{s^v} \left(\rho^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} + \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle \right). \tag{2.11}$$

Taking the ratio of the overall mean BTDs we have Eq. (2.9) in Corollary 2.2.4.

Now setting $\rho^v = 0$, it is easy to see that

$$\begin{aligned} G_v^{SS,L} &= \frac{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle}{s^v \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle + \langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}} \\ &= \frac{1}{s^v + \frac{\langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle}} \geq \frac{1}{s^v + 1 - s^v} = 1, \end{aligned}$$

Note that when $\rho^v \approxeq 0$, $\tilde{g}'_b \approx \sum_{v' \neq v} s^{v'}(1 - e^{-\rho^{v'}})\tilde{\rho}_b^{v'}$. Therefore, the inequality follows from

$$\begin{aligned}\frac{\langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle} &= \frac{\sum_{b \in \mathcal{B}} \left(\sum_{v' \neq v} s^{v'} \left(1 - e^{-\rho^{v'}}\right) \tilde{\rho}_b^{v'} \right) \delta_b^v \tilde{\rho}_b^v}{\sum_{b \in \mathcal{B}} \delta_b^v \tilde{\rho}_b^v} \\ &\leq \frac{\sum_{b \in \mathcal{B}} \left(\sum_{v' \neq v} s^{v'} \tilde{\rho}_b^{v'} \right) \delta_b^v \tilde{\rho}_b^v}{\sum_{b \in \mathcal{B}} \delta_b^v \tilde{\rho}_b^v} \\ &\leq \frac{\sum_{v' \neq v} s^{v'} \left(\sum_{b \in \mathcal{B}} \tilde{\rho}_b^v \delta_b^v \right)}{\sum_{b \in \mathcal{B}} \delta_b^v \tilde{\rho}_b^v} = 1 - s^v.\end{aligned}$$

The last inequality follows from swapping the order of summation and $\tilde{\rho}_b^{v'} \leq 1, \forall b \in \mathcal{B}, v' \in \mathcal{V}$.

Let $\tilde{\boldsymbol{g}}'_{-v} = \sum_{v' \neq v} s^{v'}(1 - e^{-\rho^{v'}})\tilde{\boldsymbol{\rho}}^{v'} + s^v \tilde{\boldsymbol{\rho}}^v$. Eq. (2.9) can be written as:

$$\begin{aligned}G_v^{SS} = &\frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \tilde{\boldsymbol{g}}'_{-v}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}} \\ &+ \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle \frac{1 - \left(\frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle} + \left(1 - \frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle}\right) \frac{s^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \tilde{\boldsymbol{g}}'_{-v}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}} \right)}{(\rho^v + 1)\langle \tilde{\boldsymbol{g}}'_{-v}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v} + s^v(\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v})}. \end{aligned} \tag{2.12}$$

Note that $\langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle \geq \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}$. Then because $\frac{s^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \tilde{\boldsymbol{g}}'_{-v}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}} \leq 1$, the numerator of the second term is nonnegative. Therefore, G_v^{SS} is decreasing in ρ^v. When $\rho^v \to \infty, \forall v \in \mathcal{V}$, the second term in Eq. (2.12) vanishes, and $\tilde{\boldsymbol{g}}'_{-v} \to \tilde{\boldsymbol{g}}$. Then $G_v^{SS,H}$ is given by $G_v^{SS,H} = \frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}{\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}$. □

The result indicates that when the relative loads are fixed, the gain decreases with the overall load ρ^v, thus if $G_v^{SS,H} > 1$ SCPF always provides a gain. Let us consider the heavy load gain under the following simplifying assumption.

Assumption 1. *Base stations are said to be homogeneous for slice v if for all $b \in \mathcal{B}$:* $\mathbb{E}^v\left[\frac{1}{C_b^v}\right] = \delta_v$.

Assumption 1 only requires the *average* reciprocal capacity a given slices' customer sees across base stations is homogenous. In this case, the BTD gain for slice v under heavy load simplifies to

$$G_v^{SS,H} = \frac{\|\tilde{\boldsymbol{\rho}}^v\|_2}{\|\tilde{\boldsymbol{g}}\|_2} \times \frac{1}{\cos(\theta(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v))}, \tag{2.13}$$

where $\theta(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v)$ denotes the angle between the slice's relative load and the overall share weighted relative load on the network. A sufficient condition for gains under high loads is that $\|\tilde{\boldsymbol{g}}\|_2 \leq \|\tilde{\boldsymbol{\rho}}^v\|_2$. Since $\|\tilde{\boldsymbol{g}}\|_1 = \|\tilde{\boldsymbol{\rho}}^v\|_1 = 1$, this follows when the overall share weighted relative load on the network is more balanced than that of slice v. One would typically expect aggregated traffic to be more balanced than that of individual slices. This condition is fairly weak, i.e., it does not depend on where the loads are placed, but on how balanced they are. The corollary also suggests that gains are higher when $\cos(\theta(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v))$ is smaller. In other words, a slice with imbalanced relative loads whose relative load distribution is 'orthogonal' to the shared weighted aggregate traffic, i.e., $\cos(\theta(\tilde{\mathbf{g}}, \tilde{\boldsymbol{\rho}}^v)) \approx 0$, will tend to see higher gains. This is due to that SCPF can achieve sharing elasticity by aligning resource allocations with demands, i.e., load distributions. Thus when the load distributions are nearly orthogonal, sharing under SCPF is much better than that under SS, which is completely inelastic. Note that, if the aggregated traffic across all slices is more imbalanced

than that of an individual slice, it is possible for that slice to observe negative BTD gain. The simulations in Section 2.5 further explore these observations.

Similarly, for the BTD gain of SCPF over GPS, we have the following result:

Corollary 2.2.5. *The BTD gain of SCPF over GPS for slice v is given by*

$$G_v^{GPS} = \frac{\rho^v \left(\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v \boldsymbol{S}^v}\right) + \langle \tilde{\boldsymbol{\rho}}^v, \mathbf{1} - \bar{\boldsymbol{s}}^v \rangle_{\boldsymbol{\Delta}^v}}{s^v \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle - s^v \left(1 - (\rho^v + 1)e^{-\rho^v}\right) \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} + (\rho^v + 1)\langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}. \quad (2.14)$$

For fixed relative loads $(\tilde{\boldsymbol{\rho}}^v : v \in \mathcal{V})$, and fixed overall loads for other slices $(\rho^{v'} : v' \neq v)$, the gain for slice v under low overall load, $\rho^v \to 0$, is given by:

$$G_v^{GPS,L} = \frac{\langle \tilde{\boldsymbol{\rho}}^v, \mathbf{1} - \bar{\boldsymbol{s}}^v \rangle_{\boldsymbol{\Delta}^v}}{s^v \langle \boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v \rangle + \langle \tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}.$$

Furthermore, if all slices have low load $\rho^v \to 0, \forall v \in \mathcal{V}$, then

$$G_v^{GPS,L} \to 1.$$

Also, if all slices have high loads, i.e., $\rho^v \to \infty, \forall v \in \mathcal{V}$, the BTD gain over GPS for slice v is given by:

$$G_v^{GPS,H} = \frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v \boldsymbol{S}^v}}{\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^v}}.$$

Note that when $(\tilde{\boldsymbol{\rho}}^v : v \in \mathcal{V})$ are fixed and $\forall v, b, \tilde{\rho}_b^v > 0$, under heavy load, i.e., $\rho^v \to \infty, \forall v \in \mathcal{V}$, we have $\bar{\boldsymbol{s}}^v \to \mathbf{0}$, thus $\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v \boldsymbol{S}^v} \to \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}$, which means GPS obtains a similar performance as SS under heavy load. However, unlike the gain over SS, $G_v^{GPS,L}$ might not be strictly greater than 1 and G_v^{GPS} might not be monotonic in ρ^v.

Proof. The BTD under SCPF is given in Eq. (2.10). Similarly for GPS we have that

$$\mathbb{E}^v\left[\frac{1}{R^{v,GPS}}\right] = \frac{1}{s^v}\left(\rho^v(\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v\boldsymbol{S}^v}) + \langle\tilde{\boldsymbol{\rho}}^v, \mathbf{1}-\bar{\boldsymbol{s}}^v\rangle_{\boldsymbol{\Delta}^v}\right). \quad (2.15)$$

Taking the ratio of the overall mean BTDs gives Eq. (2.14).

Setting $\rho^v = 0$, we have that $G_v^{GPS,L} = \frac{\langle\tilde{\boldsymbol{\rho}}^v, \mathbf{1}-\bar{\boldsymbol{s}}^v\rangle_{\boldsymbol{\Delta}^v}}{s^v\langle\boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v\rangle + \langle\tilde{\boldsymbol{g}}', \tilde{\boldsymbol{\rho}}^v\rangle_{\boldsymbol{\Delta}^v}}$. If we further have $\rho^v \to 0, \forall v \in \mathcal{V}$, then $\tilde{\boldsymbol{g}}' \to \mathbf{0}$ and

$$G_v^{GPS,L} = \frac{\langle\tilde{\boldsymbol{\rho}}^v, \mathbf{1}-\bar{\boldsymbol{s}}^v\rangle_{\boldsymbol{\Delta}^v}}{s^v\langle\boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v\rangle} = \frac{\langle\boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v\rangle - (1-s^v)\langle\boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v\rangle}{s^v\langle\boldsymbol{\delta}^v, \tilde{\boldsymbol{\rho}}^v\rangle} = 1.$$

When $\rho^v \to \infty, \forall v \in \mathcal{V}$, all the terms without ρ^v vanishes and the gain becomes $G_v^{GPS,H} \to \frac{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v} - \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v\boldsymbol{S}^v}}{\langle\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v\rangle_{\boldsymbol{\Delta}^v}}$. Note that even we have $\rho^v \to \infty$, we cannot guarantee that $\rho_b^v \to \infty, \forall b \in \mathcal{B}$ if for some $b \in \mathcal{B}, \tilde{\rho}_b^v = 0$. Thus $\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{S}^v}$ might not approach 0. □

One can observe that, different slices may experience different BTD gains, depending on the share and load distributions. However, to compare the performance of different sharing criteria, a network-wide metric of gain needs to be defined. To be able to compare scenarios with different load distributions and shares, it is of particular interest to consider a metric which accounts for differences in slices' shares s^v, loads ρ^v, and base-station capacities $\boldsymbol{\delta}^v$. Note that users experiencing a low average capacity from their associated base stations, $\frac{1}{\delta_b^v}$, and/or are allocated a small share per user, i.e., $\frac{s^v}{\rho^v}$, are expected to experience higher BTDs. Thus to account for these differences,

let us define the normalized BTD for a typical user on slice v at base station b under SCPF as

$$\bar{\mathbb{E}}^v\left[\frac{1}{R_b^v}\right] := \frac{1}{\delta_b^v}\frac{s^v}{\rho^v}\mathbb{E}^v\left[\frac{1}{R_b^v}\right], \tag{2.16}$$

and thus the normalized BTD for a typical user on slice v under SCPF is given by

$$\bar{\mathbb{E}}^v\left[\frac{1}{R^v}\right] = \sum_{b\in\mathcal{B}}\tilde{\rho}_b^v\bar{\mathbb{E}}^v\left[\frac{1}{R_b^v}\right]. \tag{2.17}$$

Similarly, one can define $\bar{\mathbb{E}}^v\left[\frac{1}{R_b^{v,SS}}\right]$, $\bar{\mathbb{E}}^v\left[\frac{1}{R^{v,SS}}\right]$, and $\bar{\mathbb{E}}^v\left[\frac{1}{R_b^{v,GPS}}\right]$, $\bar{\mathbb{E}}^v\left[\frac{1}{R^{v,GPS}}\right]$. For the overall performance of the system, let us consider the share weighted sum of the normalized BTD since the system should be tuned to put more emphasis on the slices with higher shares, and define the overall weighted BTD gain of SCPF over SS as

$$G_{\text{all}}^{SS} := \frac{\sum_{v\in\mathcal{V}} s^v\bar{\mathbb{E}}^v\left[\frac{1}{R^{v,SS}}\right]}{\sum_{v\in\mathcal{V}} s^v\bar{\mathbb{E}}^v\left[\frac{1}{R^v}\right]}, \tag{2.18}$$

and the overall weighted BTD gain of SCPF over GPS as

$$G_{\text{all}}^{GPS} := \frac{\sum_{v\in\mathcal{V}} s^v\bar{\mathbb{E}}^v\left[\frac{1}{R^{v,GPS}}\right]}{\sum_{v\in\mathcal{V}} s^v\bar{\mathbb{E}}^v\left[\frac{1}{R^v}\right]}, \tag{2.19}$$

The following results capture the overall weighted BTD gains.

Corollary 2.2.6. *When $\rho^v \to \infty, \forall v \in \mathcal{V}$, the overall weighted BTD gains of SCPF over SS and GPS under heavy load are given by*

$$G_{all}^{SS,H} = \frac{\sum_{v\in\mathcal{V}} s^v\|\tilde{\boldsymbol{\rho}}^v\|_2^2}{\sum_{v\in\mathcal{V}} s^v\langle\tilde{\boldsymbol{g}},\tilde{\boldsymbol{\rho}}^v\rangle}, \quad G_{all}^{GPS,H} = \frac{\sum_{v\in\mathcal{V}} s^v\|\tilde{\boldsymbol{\rho}}^v\|_{\boldsymbol{I}-\boldsymbol{S}^v}^2}{\sum_{v\in\mathcal{V}} s^v\langle\tilde{\boldsymbol{g}},\tilde{\boldsymbol{\rho}}^v\rangle}, \tag{2.20}$$

and

$$G_{all}^{SS,H} \geq 1, \; G_{all}^{GPS,H} \geq 1.$$

It is easy to see that if $\tilde{\boldsymbol{\rho}}^v$ are the same for all $v \in \mathcal{V}$, then both $G_{\text{all}}^{SS,H}$ and $G_{\text{all}}^{GPS,H}$ are 1 when the loads are heavy. By contrast, if the relative loads of different slices are (approximately) all orthogonal, i.e. , $\langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^{v'} \rangle \cong 0, v \neq v'$ and each slice has the same share $s^v = \frac{1}{V}, \forall v \in \mathcal{V}$, the overall gain can be as high as V.

Proof. From Theorem 2.2.2, one can express the overall weighted BTD gain over SS under heavy load as

$$G_{\text{all}}^{SS,H} = \frac{\sum_v s^v \|\tilde{\boldsymbol{\rho}}^v\|_2^2}{\sum_v s^v \langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle} = \frac{\sum_v \sum_b s^v (\tilde{\rho}_b^v)^2}{\sum_v s^v \sum_b (\sum_{v'} s^{v'} \tilde{\rho}_b^{v'}) \tilde{\rho}_b^v}$$
$$= \frac{\sum_b \sum_v s^v (\tilde{\rho}_b^v)^2}{\sum_b (\sum_{v'} s^{v'} \tilde{\rho}_b^{v'})(\sum_v s^v \tilde{\rho}_b^v)} = \frac{\sum_b \sum_v s^v (\tilde{\rho}_b^v)^2}{\sum_b (\sum_v s^v \tilde{\rho}_b^v)^2}.$$

According to Jensen's inequality, we have $\forall b \in \mathcal{B}$, $\sum_v s^v (\tilde{\rho}_b^v)^2 \geq (\sum_v s^v \tilde{\rho}_b^v)^2$. Thus $G_{\text{all}}^{SS} \geq 1$.

Similarly, for GPS, we have

$$G_{\text{all}}^{GPS,H} = \frac{\sum_v \sum_b s^v (\tilde{\rho}_b^v)^2 \left(1 - \sum_{v' \neq v} s^{v'} e^{-\rho_b^{v'}}\right)}{\sum_v \sum_b s^v \tilde{\rho}_b^v \sum_{v'} s^{v'} \tilde{\rho}_b^{v'}}$$
$$= \frac{\sum_b \sum_v s^v (\tilde{\rho}_b^v)^2 - \sum_b \sum_v s^v (\tilde{\rho}_b^v)^2 \sum_{v' \neq v} s^{v'} e^{-\rho_b^{v'}}}{\sum_b (\sum_v s^v \tilde{\rho}_b^v)^2}.$$

As $\rho^v \to \infty$, for each base station $b \in \mathcal{B}$, if $\rho_b^v \to \infty$, $e^{-\rho_b^v} \to 0$, otherwise $\tilde{\rho}_b^v = \frac{\rho_b^v}{\rho^v} \to 0$. Based on such observation, let us define a set of slices at each base station b, whose local loads approach infinity, $\mathcal{V}_b^{\text{inf}} \triangleq \{v \in \mathcal{V} : \rho_b^v \to \infty\}$.

Then the above equation can be rewritten as:

$$G_{\text{all}}^{GPS,H} = \frac{\sum_b \left(\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v (\tilde{\rho}_b^v)^2 (1 - \sum_{v' \notin \mathcal{V}_b^{\text{inf}}} s^{v'} e^{-\rho_b^{v'}}) \right)}{\sum_b (\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v \tilde{\rho}_b^v)^2}$$
$$= \frac{\sum_b \left((1 - \sum_{v' \notin \mathcal{V}_b^{\text{inf}}} s^{v'} e^{-\rho_b^{v'}}) (\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v) \sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v (\tilde{\rho}_b^v)^2 \right)}{\sum_b (\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v)^2 (\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v \tilde{\rho}_b^v)^2},$$

where for $v \in \mathcal{V}_b^{\text{inf}}$, $\tilde{s}^v \triangleq \frac{s^v}{\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v}$. Therefore $\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v = 1$. Now for each base station $b \in \mathcal{B}$, we have

$$\frac{(1 - \sum_{v' \notin \mathcal{V}_b^{\text{inf}}} s^{v'} e^{-\rho_b^{v'}}) (\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v) \sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v (\tilde{\rho}_b^v)^2}{(\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v)^2 (\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v \tilde{\rho}_b^v)^2}$$
$$= \frac{(1 - \sum_{v \notin \mathcal{V}_b^{\text{inf}}} s^v e^{-\rho_b^v}) \sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v (\tilde{\rho}_b^v)^2}{(\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v) (\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v \tilde{\rho}_b^v)^2}$$
$$\geq \frac{(1 - \sum_{v \notin \mathcal{V}_b^{\text{inf}}} s^v) \sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v (\tilde{\rho}_b^v)^2}{(\sum_{v \in \mathcal{V}_b^{\text{inf}}} s^v) (\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v \tilde{\rho}_b^v)^2} = \frac{\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v (\tilde{\rho}_b^v)^2}{(\sum_{v \in \mathcal{V}_b^{\text{inf}}} \tilde{s}^v \tilde{\rho}_b^v)^2},$$

where the last equality holds true because $\sum_v s^v = 1$. Then by Jensen's inequality, for all $b \in \mathcal{B}$, the above ratio is no less than 1, thus $G_{\text{all}}^{GPS,H} \geq 1$. $\square$

2.3 Share Dimensioning in Network Slicing

2.3.1 Feasibility of Share Dimensioning

In practice each slice v may wish to provide service guarantees to its customers, i.e., ensure that the mean BTD does not exceed a performance target d_v. Below we investigate how to dimension network shares to support slice loads subject to such mean BTD requirements.

Henceforth we shall assume the following assumption is in effect.

Assumption 2. *The network is said to see high overall slice loads, if for all $v \in \mathcal{V}$ we have $\rho^v \gg 1$.*

Consider a network supporting the traffic loads of a *single* slice, say v, so $s^v = 1$ and $\tilde{\boldsymbol{g}} = \tilde{\boldsymbol{\rho}}^v$. Note that $\langle \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v \rangle$ is the minimum average BTD achievable across the network when a slice gets *all* the base station resources, so a target requirement satisfies $d_v > \langle \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v \rangle$. For slice v to meet a mean BTD constraint d_v, it follows from Eq. (2.6) that:

$$\rho^v \leq l(d_v, \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v) \triangleq \frac{d_v - \langle \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v \rangle}{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}.$$

We can interpret $l(d_v, \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v)$ as the maximal admissible carried load ρ^v given a fixed relative load distribution $\tilde{\boldsymbol{\rho}}^v$, BTD requirement d_v, and mean reciprocal capacities $\boldsymbol{\delta}^v$. As might be expected, if the relative load distribution $\tilde{\boldsymbol{\rho}}^v$ is more balanced (normalized by the mean base station capacity), i.e., $\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}$ is smaller, or if the BTD constraint is relaxed, i.e., d_v is higher, or the base station capacities scale up, i.e., $\boldsymbol{\delta}^v$ is smaller, the slice can carry a higher overall load ρ^v.

Next, let us consider SCPF based sharing amongst a set of slices $\mathcal{V}$ each with its own BTD requirements. It follows from Eq. (2.6) that to meet such requirements on each slice the following should hold: for all $v \in \mathcal{V}$

$$s^v \geq \frac{1+\rho^v}{l(d_v, \tilde{\boldsymbol{\rho}}^v, \boldsymbol{\delta}^v) - \rho^v} \sum_{u \neq v} s^u \frac{\langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}^v}}{\|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}^v}}. \tag{2.21}$$

This can be written as:

$$\sum_{v \in \mathcal{V}} s^v \boldsymbol{h}^v \succeq \mathbf{0}, \tag{2.22}$$

where we refer to $\boldsymbol{h}^v = (h^v_u : u \in \mathcal{V})$ as v's *share coupling vector*, given by

$$h^v_u = \begin{cases} 1 & v = u, \\ -\frac{1+\rho^u}{l(d_u, \tilde{\boldsymbol{\rho}}^u, \boldsymbol{\delta}^u) - \rho^u} \frac{\langle \tilde{\boldsymbol{\rho}}^u, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^u}}{\|\tilde{\boldsymbol{\rho}}^u\|^2_{\boldsymbol{\Delta}^u}} & v \neq u. \end{cases}$$

We can interpret $h^v_v = 1$ as the benefit to slice v of allocating unit share to itself. When $v \neq u$, h^v_u depends on two factors. The first $\frac{1+\rho^u}{l(d_u, \tilde{\boldsymbol{\rho}}^u, \boldsymbol{\delta}^u) - \rho^u}$ captures the sensitivity of slice u to the 'share weighted congestion' from other slices. If ρ^u is close to its limit $l(d_u, \tilde{\boldsymbol{\rho}}^u, \boldsymbol{\delta}^u)$, its sensitivity is naturally very high. The second term, $\frac{\langle \tilde{\boldsymbol{\rho}}^u, \tilde{\boldsymbol{\rho}}^v \rangle_{\boldsymbol{\Delta}^u}}{\|\tilde{\boldsymbol{\rho}}^u\|^2_{\boldsymbol{\Delta}^u}}$ captures the impact of slice v's load distribution on slice u. Note that if two slices load distributions are orthogonal, they do not affect each other.

The following result summarizes the above analysis.

Theorem 2.3.1. *There exists a share allocation such that slice loads and BTD constraints* $((\rho^v, \tilde{\boldsymbol{\rho}}^v, d_v) : v \in \mathcal{V})$ *are admissible under SCPF sharing if and only if there exists an* $\boldsymbol{s} = (s^v : v \in \mathcal{V})$ *such that* $\|\boldsymbol{s}\|_1 = 1$, $\boldsymbol{s} \succeq \mathbf{0}$ *and* $\sum_{v \in \mathcal{V}} s^v \boldsymbol{h}^v \succeq 0$.

Admissibility can then be verified by solving the following maxmin problem:

$$\max_{\boldsymbol{s}\succeq\mathbf{0}}\{\ \min_{i}\sum_{v\in\mathcal{V}}s^{v}h_{i}^{v}:\|\boldsymbol{s}\|_{1}=1\ \}. \tag{2.23}$$

If the optimal objective function is positive, the traffic pattern is admissible. Moreover, if there are multiple feasible share allocations, then the optimizer is a 'robust' choice in that it maximizes the minimum share given to any slice, giving slices margins to tolerate perturbations in the slice loads satisfying Eq. (2.22).

2.3.2 Pricing in Share Dimensioning

2.3.2.1 Motivation

So far, we studied a novel resource sharing criterion in network slicing, which can be viewed as achieving a service level agreement (SLA) on the algorithm for mapping the parameter – share of overall network resources $(s^v : v \in \mathcal{V})$ to resource provisioning at each base station. Such SLA provides an efficient and elastic way to better accommodate requirements of mobile users. However, how to determine the share for each slice/tenant based upon their individual load distribution and/or service requirement remains an open problem.

In this section we will discuss an price-based approach where each slice is charged a certain amount of 'money' according to their allocated share. This is similar to the current usage-based pricing scheme in nowadays network

service. By charging based on share of overall network resources instead of usage of resources at each base station, we reduced the complexity of the scheme and allow the system to scale up, where pricing based on actual usage requires us to keep track of the dynamic user distribution and their interaction with base stations, which will put an excessive computational overhead on the network operation.

A desirable pricing scheme shall provide incentive for each slice to purchase an appropriate share of resource, given its service requirement and load distribution, in order to achieve social welfare. Implicitly, it also encourages individual slices to manage their traffic so that their service requirement is achievable within limited resource provisioned, thus inter-slice protection is achieved.

2.3.2.2 Share Dimensioning Revisit: An Optimization Perspective

In Section 2.3.1, we discussed how to determine the feasibility of a (loads, shares, QoS requirements)-tuple, as the *Share dimensioning* problem. In this section, we would like to revisit this problem, to discuss from an economic perspective. First we consider the parallel resource case, as in [7], in which we obtain the closed-form expression for the resource allocation.

To be consistent, we shall consider the utility of each slice to be proportional to its average BTD, e.g., load times BTD, and the social welfare/utility as the negate of the weighted sum of BTDs. In view of this, we assume that there is a central entity controlling the allocation of share $\mathbf{s} := (s^v : v \in \mathcal{V})$, in

order to minimize the social cost.

Social Optimal:

$$\begin{aligned} \max_{\mathbf{s}} \quad & -\sum_{v\in\mathcal{V}} \rho^v \frac{\rho^v}{s^v} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\mathbf{g}} \rangle_{\boldsymbol{\Delta}^v} \\ \text{such that} \quad & \sum_{v\in\mathcal{V}} s^v \le 1 \\ & s^v \ge \frac{\rho^v}{d_v - \rho^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}_v}} \sum_{u\neq v} s^u \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}_v}, \forall v \in \mathcal{V}. \end{aligned} \tag{2.24}$$

In general, such problem is nonconvex, due to the bilinear objective function. However, through a geometric programming transformation we have the following result.

Theorem 2.3.2. *There is a unique solution $\boldsymbol{s}^*$ such that $\sum_{v\in\mathcal{V}} s^{v,*} = 1$, and $\boldsymbol{s}^*$ can be obtained by solving the KKT condition associated with Prob. (2.24) after changing the first inequality constraint to equality.*

Proof. By transformation $b_v = \log s^v$, the above problem is equivalent to

$$\begin{aligned} \min_{\mathbf{b}:=(b_v:v\in\mathcal{V})} \quad & \log\left(\sum_{v\in\mathcal{V}} \sum_{u\in\mathcal{V}} (\rho^v)^2 \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}^v} e^{b_u - b_v} \right) \\ \text{such that} \quad & \sum_{v\in\mathcal{V}} e^{b_v} \le 1 \\ & \rho^v \sum_{u\in\mathcal{V}} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}^v} e^{b_v - b_u} \le d_v, \end{aligned} \tag{2.25}$$

which is a convex optimization problem. Thus its solution can be obtained by solving the KKT conditions. The following proposition provides some insight

on the share allocations at the minimum even the objective function is not strongly convex.

Proposition 2.3.3. *There is a unique solution* $\mathbf{b}^*$ *to the above problem such that* $\sum_{v\in\mathcal{V}} e^{b_v^*} = 1$.

Proof. First, the Hessian of log-sum-exp function is given in [34], and it is PSD. We have for a direction vector $\boldsymbol{v} := (v_{u,v} : u, v \in \mathcal{V}) \in \mathbb{R}^{|\mathcal{V}|^2}$, $\boldsymbol{v}^T\nabla^2 f(x)\boldsymbol{v} = 0$, only when it is such that $\forall v, u \in \mathcal{V}, e^{b_u - b_v} \parallel e^{b_u - b_v} v_{u,v}^2$. That implies $\boldsymbol{v} \parallel \mathbf{1}$. Therefore, if we have two different optimal solutions, they must be of the form $\boldsymbol{b}^* = \boldsymbol{b}_0^* + C\mathbf{1}$, where $\boldsymbol{b}_0^*$ is fixed and $C \in \mathbb{R}$ can be different for different solutions.

Therefore, among the family of the solutions there is a unique one such that $\sum_{v\in\mathcal{V}} e^{b_v^*} = 1$. □

Because the associated constraints in Problem (2.24) are all affine, the KKT condition provides at least local minimum. Say the actual minimizer for $\sum_{v\in\mathcal{V}} s^v = 1$ is $\boldsymbol{s}^*$ and that given by the KKT conditions is $\boldsymbol{s}'$. Then, if $\boldsymbol{s}^* \neq \boldsymbol{s}'$, by Proposition 2.3.3, the objective function at $\boldsymbol{s}^*$ is strictly less than that at $\boldsymbol{s}'$. By a convex combination of form $\boldsymbol{s} = \lambda\boldsymbol{s}^* + (1-\lambda)\boldsymbol{s}'$ with λ small enough, one can find a point obtaining a smaller objective value in the neighborhood of $\boldsymbol{s}'$, violating the definition of local minimum. Therefore, by contradiction we have the optimal $\boldsymbol{s}^*$ can be obtained via KKT condition. □

Moreover, if the BTD constraints are loose, we have a closed form expression for the optimal solution.

Corollary 2.3.4. *If we assume homogeneous resources, that is* $\mathbf{\Delta}_v = \delta_v \mathbf{I}$*, and all the BTD constraints are inactive, by solving the KKT conditions, we can show that the optimal share allocation is such that*

$$s^{v,*} \propto \sqrt{\delta_v} \rho^v.$$

Proof. This is straightforward to show by setting $\mu_v = 0, \forall v \in \mathcal{V}$, and setting the gradient of the Lagrangian to $\mathbf{0}$. □

2.3.2.3 Pricing Strategy

We are interested in a pricing scheme that assigns a price π_v for a unit share purchased to slice v. Suppose each slice views the negate of its BTD, less the price paid for its share, i.e., $-\text{BTD}_v - \pi_v s^v$ as its own utility and wishes to maximize it subject to the BTD constraint. Such pricing scheme introduces a game among slices. Formally, the joint strategy is $\boldsymbol{s} := (s^v : v \in \mathcal{V})$, and $\boldsymbol{s}^{-v} := (s^u : u \neq v)$ denotes the strategy of slices other than v. Then each slice solves the following problem

Slice-based optimization:

$$\begin{aligned} \min_{s^v} \quad & \phi_v(s^v; \boldsymbol{s}^{-v}) := \frac{\rho^v}{s^v} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{g}} \rangle_{\mathbf{\Delta}_v} + \pi_v s^v \\ \text{such that:} \quad & s^v \geq \frac{\rho^v}{d_v - \rho^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\mathbf{\Delta}_v}} \sum_{u \neq v} s^u \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\mathbf{\Delta}_v}, \end{aligned} \tag{2.26}$$

where π_v is the price per share central entity charges to slice v. One can see that the strategic space for different slices are coupled together through the BTD constraint. However, we can still use the notion of generalized Nash equilibrium (GNE) as in [7]

Theorem 2.3.5. *The game induced by Problem (2.26) has GNE.*

Proof. This is a direct result following Theorem 3.1 in [35]. □

In practice, the right of specifying the amount of share s^v should be delegated to slice v instead of being dictated by the central entity due to privacy reason, and also providing each slice the ability to conduct performance management. Thus, the central entity shall devise a pricing strategy, represented as $\boldsymbol{\pi} := (\pi_v : v \in \mathcal{V})$, in order to motivate a share allocation $\mathbf{s}$ that achieves desirable global performance, e.g., low weighted average BTD. Such pricing strategy is provided by the following theorem.

Theorem 2.3.6. *If we set* $\pi_v = \sum_{u \neq v} h_u^v \mu_u^* + \sum_{u \neq v} \frac{\rho^u}{s^{u,*}} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}_u}$, *where* $h_u^v := \frac{\rho^u}{d_u - \rho^u \|\tilde{\boldsymbol{\rho}}^u\|^2_{\boldsymbol{\Delta}_u}} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}_u}$, *and* μ_u^* *is the optimal dual variable associated with the BTD constraint of slice* u *in Problem (2.24), the solution to Problem (2.24) is a GNE induced by Problem (2.26)*

Proof. The Lagrangian associated with Problem (2.24) is given by

$$L(\mathbf{s}; \lambda, \boldsymbol{\mu}) := \sum_{v \in \mathcal{V}} \frac{\rho^v}{s^v} \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\mathbf{g}} \rangle_{\boldsymbol{\Delta}^v} + \lambda \left(\sum_{v \in \mathcal{V}} s^v - 1 \right) + \sum_{v \in \mathcal{V}} \mu_v \left(\frac{\rho^v}{d_v - \rho^v \|\tilde{\boldsymbol{\rho}}^v\|^2_{\boldsymbol{\Delta}_v}} \sum_{u \neq v} s^u \langle \tilde{\boldsymbol{\rho}}^v, \tilde{\boldsymbol{\rho}}^u \rangle_{\boldsymbol{\Delta}_v} - s^v \right). \tag{2.27}$$

By noting that only the ratio between s^v matters, we conclude that relaxing the constraint that $\sum_{v\in\mathcal{V}} s^v \leq 1$ does not change the solution to the problem, thus, the optimal dual λ^* should be 0. Therefore, the social optimum of Problem (2.24) satisfies the following conditions:

$$\forall v \in \mathcal{V} : \frac{\partial L(\mathbf{s};\lambda,\boldsymbol{\mu})}{\partial s^v} = -\frac{\rho^v}{(s^v)^2}\sum_{u\neq v}\langle\tilde{\boldsymbol{\rho}}^v,\tilde{\boldsymbol{\rho}}^u\rangle_{\boldsymbol{\Delta}^v}s^u + \sum_{u\neq v}\frac{\rho^u}{s^u}\langle\tilde{\boldsymbol{\rho}}^v,\tilde{\boldsymbol{\rho}}^u\rangle_{\boldsymbol{\Delta}^u} - \mu_v + \sum_{u\neq v}h_u^v\mu_u, \tag{2.28}$$

$$\forall v \in \mathcal{V} : \mu_v\left(\sum_{u\neq v}h_v^u s^u - s^v\right) = 0, \tag{2.29}$$

Also the BTD constraints are satisfied. According to Thm. 2.3.2, optimal $\mathbf{s}^*$ satisfies above conditions with optimal dual variable $\boldsymbol{\mu}^*$.

For slice v, the Lagrangian of Problem (2.26) is

$$L^v(s^v;\mu_v) = \frac{\rho^v}{s^v}\langle\tilde{\boldsymbol{\rho}}^v,\tilde{\boldsymbol{g}}\rangle_{\boldsymbol{\Delta}_v} + \pi_v s^v + \mu_v\left(\sum_{u\neq v}h_v^u s^u - s^v\right). \tag{2.30}$$

Then the optimal s^v should satisfy following conditions:

$$\frac{\partial L^v(s^v;\mu_v)}{\partial s^v} = -\frac{\rho^v}{(s^v)^2}\sum_{u\neq v}\langle\tilde{\boldsymbol{\rho}}^v,\tilde{\boldsymbol{\rho}}^u\rangle_{\boldsymbol{\Delta}^v}s^u + \pi_v - \mu_v = 0,$$

and

$$\mu_v\left(\sum_{u\neq v}h_v^u s^u - s^v\right) = 0.$$

If $\pi_v = \sum_{u\neq v}h_u^v\mu_u^* + \sum_{u\neq v}\frac{\rho^u}{s^{u,*}}\langle\tilde{\boldsymbol{\rho}}^v,\tilde{\boldsymbol{\rho}}^u\rangle_{\boldsymbol{\Delta}_u}$, one can easily verify that $\mathbf{s}^*$ and $\boldsymbol{\mu}^*$ satisfies the combination of above conditions across $v\in\mathcal{V}$, thus the social optimum is also a NE induced by Problem (2.26). □

2.3.2.4 Two Slice Case

One critical question regarding the pricing strategy is to study how the price $\boldsymbol{\pi}$ depend on the performance requirements and the load distribution of each slice. Because of the dependency on the optimal dual variables, generally it is hard to find a closed form expression for the pricing scheme when there are multiple slices and the load distributions are complex. However, if there are only two slices things can be simplified significantly.

Theorem 2.3.7. *When $|\mathcal{V}| = 2$, we have following cases:*

1. *If both slices are not bound by their BTD constraints, $\mu_1 = \mu_2 = 0$, then we have*

$$\begin{aligned} \pi_1 &= \sqrt{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}} + \rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}, \\ \pi_2 &= \sqrt{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}} + \rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}. \end{aligned} \tag{2.31}$$

2. *If both slices are bound by their BTD constraints, then $\mathbf{s}$ is given by solving linear equations, and the price is given by*

$$\begin{aligned} \pi_1 &= \frac{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}{d_2 - \rho^2\|\tilde{\boldsymbol{\rho}}^2\|^2_{\boldsymbol{\Delta}_2}}\left(1 + \frac{\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}{d_2 - \rho^2\|\tilde{\boldsymbol{\rho}}^2\|^2_{\boldsymbol{\Delta}_2}}\right), \\ \pi_2 &= \frac{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}{d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}}\left(1 + \frac{\rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}}.\right) \end{aligned} \tag{2.32}$$

3. *W.L.O.G., assume slice 1 is bound by the BTD constraint but slice 2 is not, then one can show that the pricing should be*

$$\begin{aligned} \pi_1 &= \rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}\left(1 + \frac{\rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}}\right), \\ \pi_2 &= \frac{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}{d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}}\left(1 + \frac{\rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}}\right). \end{aligned} \tag{2.33}$$

Remark: We can summarize following interpretations of the pricing strategy. First, in all three cases, both π_1 and π_2 increases with ρ^1, ρ^2 and the inter-slice intervention $\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}$, $\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}$,

1. In case 1, besides the common price $\sqrt{\rho^1\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}$, which increases with the overall level of congestion of the system, π_1 has an additional term $\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}$, which can be viewed as the sensitivity of slice 2's BTD against slice 1's share. If the load distribution of slice 1 is such that it causes a lot of congestion to slice 2's users, the central entity tends to charge more to slice 1.

2. In case 3, both price increase with ρ^1, ρ^2, inter-slice contention $\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}$, $\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}$ and decrease with d_1. It is independent of d_2 because d_2 is not binding. Moreover, when $\rho_1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1} \leq d_1 - \rho^1\|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}$, we have $s^2 \geq s^1$. Such condition requires that the contention brought by slice 2 to slice 1 is upperbounded. Specifically, if we assume $\boldsymbol{\Delta}_1 = \mathbf{I}$, it is equivalent to $\|\tilde{\boldsymbol{\rho}}^2\|\cos\theta(1,2) \leq (\frac{l^1}{\rho^1} - 1)\|\tilde{\boldsymbol{\rho}}^1\|$, where $l^1 := \frac{d_1}{\|\tilde{\boldsymbol{\rho}}^1\|^2}$ is the maximal allowed traffic on slice 1.

Proof. When there are two slices, we have $s^2 = 1 - s^1$. Then setting the gradient to s^1 of the Lagrangian to 0 we have

$$s^{1,*} \quad = \sqrt{\frac{\rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}(1 - s^{1,*})}{-\mu_1^* + \frac{\rho^2}{1-s^{1,*}}\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}}.$$

Rearranging terms we have

$$\mu_1^* = \frac{\rho^2\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_2}}{1 - s^{1,*}} - \frac{\rho^1\langle\tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2\rangle_{\boldsymbol{\Delta}_1}(1 - s^{1,*})}{(s^{1,*})^2}. \tag{2.34}$$

Similarly we can have μ_2^*.

1. When both BTD constraints are inactive, we know $\mu_1^* = \mu_2^* = 0$, then we have $s^{i,*} \propto \sqrt{\rho^i \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_i}}$. Plugging in Eq. (2.34) we have

$$\begin{aligned} \pi_1 &= \sqrt{\rho^1 \rho^2 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_2}} + \rho^2 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_2} \\ \pi_2 &= \sqrt{\rho^1 \rho^2 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_2}} + \rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \end{aligned}$$

2. By solving the BTD constraint for slice 1 we have

$$s^{1,*} = \frac{\rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1 \|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1} + \rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}},$$

   and plugging in Eq. (2.34), we have the μ_1^*. Similarly we can obtain μ_2^*, then π_1, π_2 follow.

3. In case 3, we know the BTD constraint of slice 1 is active. By solving it we have

$$s^{1,*} = \frac{\rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1 \|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1} + \rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}}.$$

   Therefore we have the price

$$\begin{aligned} \pi_1 &= \frac{\rho^2}{1 - s^{1,*}} \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \\ &= \rho^2 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_2} \left(1 + \frac{\rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1 \|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}} \right), \\ \pi_2 &= h_1^2 \mu_1^* + \frac{\rho^1}{s^{1,*}} \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \\ &= \frac{\rho^1 \rho^2 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1} \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_2}}{d_1 - \rho^1 \|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}} \left(1 + \frac{\rho^1 \langle \tilde{\boldsymbol{\rho}}^1, \tilde{\boldsymbol{\rho}}^2 \rangle_{\boldsymbol{\Delta}_1}}{d_1 - \rho^1 \|\tilde{\boldsymbol{\rho}}^1\|^2_{\boldsymbol{\Delta}_1}} \right). \end{aligned}$$

□

If a set of network slice loads and BTD constraints are not admissible, admission control will need to be applied. We discuss this in the next section.

2.4 Admission Control and Traffic Shaping Games

A natural approach to managing performance in overloaded systems is to perform admission control. In the context of slices supporting mobile services where spatial loads may vary substantially, this may be unavoidable. Below we consider admission control policies that adapt to changes in load. Specifically, an *admission control policy* for slice v is parameterized by $\boldsymbol{a}^v \triangleq (a_b^v : b \in \mathcal{B}) \in [0,1]^B$ where a_b^v is the probability a new customer at base station b is admitted. Such decisions are assumed to be made independently thus admitted customers for slice v at base station b still follow a Poisson Process with rate $\gamma_b^v a_b^v$. Based on the flow conservation equation Eq. (2.1) one can obtain the carried load $\boldsymbol{\rho}^v$ induced by admission control policy $\boldsymbol{a}^v$ via

$$\boldsymbol{\rho}^v \;=\; (\boldsymbol{M}^v)^{-1}\boldsymbol{a}^v \;=\; \mathrm{diag}(\boldsymbol{\mu}^v)(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)^{-1}\mathrm{diag}(\boldsymbol{\gamma}^v)\boldsymbol{a}^v$$

where $\boldsymbol{M}^v \triangleq \mathrm{diag}(\boldsymbol{\gamma}^v)^{-1}(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)\mathrm{diag}(\boldsymbol{\mu}^v)^{-1}$ is invertible because $\boldsymbol{I} - (\boldsymbol{Q}^v)^T$ is irreducibly diagonally dominant.[1] By contrast with Section 2.2.1, note that $\boldsymbol{\rho}^v$ now represents the load after admission control, which may have a reduced overall load and possibly changed relative loads across base stations–i.e., *shape* the traffic on the slice. We also let $\tilde{\boldsymbol{g}}$ be the overall share weighted relative loads after admission control, see Section 2.2.2.1. Note that we have assumed only exogenous arrivals can be blocked, thus once a customer is admitted it will not be dropped–the intent is to manage performance to maintain *service continuity*.

[1] If $\boldsymbol{\gamma}^v$ is not strictly positive one can reduce the dimensionality.

Below we consider a setting where slices *unilaterally* optimize their admission control policies in response to network congestion, rather than a single joint global optimization. The intent is to allow slices (which may correspond to competing virtual operators/services) to optimize their own performance, and/or enable decentralization in settings with SCPF based sharing.

For simplicity we assume that Assumption 1 holds true throughout this section, and define the capacity normalized mean BTD requirement $\tilde{d}_v \triangleq \frac{d_v}{\delta_v}$. Suppose each slice v optimizes its admission control policy so as to maximize its overall carried load ρ^v, i.e., the average number of active users on the network, subject to a normalized mean BTD constraint $\tilde{d}_v$. Under Assumption 1 the optimal policy for slice v is the solution to the following optimization problem:

$$\max_{\tilde{\boldsymbol{\rho}}^v, \rho^v} \quad \rho^v \tag{2.35}$$

$$\text{s.t.} \quad \boldsymbol{a}^v = \rho^v \boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v, \quad \boldsymbol{a}^v \in [0,1]^B, \quad \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle = 1 \tag{2.36}$$

$$\frac{(\rho^v + 1)}{s^v} \langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle - \left(1 - (\rho^v + 1) e^{-\rho^v}\right) \|\tilde{\boldsymbol{\rho}}^v\|_2^2 \leq \tilde{d}_v - 1 \tag{2.37}$$

Note that Equation (2.36) establishes a one-to-one mapping between $(\tilde{\boldsymbol{\rho}}^v, \rho^v)$ and $\boldsymbol{a}^v$. We will use $\tilde{\boldsymbol{\rho}}^v$ and ρ^v to parameterize admission control decisions for slice v. The BTD constraint in Eq. (2.37) follows from Eq. (2.6). Also note that this admission control policy depends on both the overall share weighted loads on the network $\tilde{\boldsymbol{g}}$, the slice's load and its customer mobility patterns (i.e., $\boldsymbol{M}^v$). Unfortunately, for general loads ρ^v, this problem is not

convex due to the BTD constraint Eq. (2.37); however, for high overall per slice loads it is easily approximable by a convex function.

Under Assumption 2 we have that $1 + \rho^v \approx \rho^v$ and the left hand side of Eq. (2.37) becomes:

$$\frac{(\rho^v + 1)}{s^v}\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle - \|\tilde{\boldsymbol{\rho}}^v\|_2^2 \approx \frac{\rho^v}{s^v}\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle = (s^v x_v)^{-1}\langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle \tag{2.38}$$

where we have defined $x_v \triangleq (\rho^v)^{-1}$. Further defining $\tilde{\boldsymbol{\rho}}^{-v} \triangleq (\tilde{\boldsymbol{\rho}}^{v'} : v' \in \mathcal{V}\backslash\{v\})$, Equation (2.37) can be replaced by:

$$f_v(\tilde{\boldsymbol{\rho}}^v; \tilde{\boldsymbol{\rho}}^{-v}) \triangleq \langle \tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v \rangle \leq s^v(\tilde{d}_v - 1)x_v. \tag{2.39}$$

Thus, by defining $\boldsymbol{y}^v \triangleq (\tilde{\boldsymbol{\rho}}^v, x^v)$, which is equivalent to $(\tilde{\boldsymbol{\rho}}^v, \rho^v)$, together with $\boldsymbol{y}^{-v} \triangleq (\boldsymbol{y}^{v'} : v' \in \mathcal{V}\backslash\{v\})$, each slice can unilaterally optimize its admission control policy by solving the following problem:

Admission control for slice v under SCPF ($\mathbf{AC}_v$): Given other slices' admission decisions $\boldsymbol{y}^{-v}$, slice v determines its admission control policy $\boldsymbol{y}^v = (\tilde{\boldsymbol{\rho}}^v, x^v)$ by solving

$$\min_{\boldsymbol{y}^v}\{\ x_v \mid \boldsymbol{y}^v \in Y^v(\boldsymbol{y}^{-v})\ \} \tag{2.40}$$

where $Y^v(\boldsymbol{y}^{-v})$ denotes slice v's feasible policies and is given by

$$\begin{aligned} Y^v(\boldsymbol{y}^{-v}) \ \triangleq \{\ \boldsymbol{y}^v \mid \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle = 1,\ \mathbf{0} \preceq \boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v \preceq x_v \mathbf{1}, \\ f_v(\tilde{\boldsymbol{\rho}}^v; \tilde{\boldsymbol{\rho}}^{-v}) \leq s^v(\tilde{d}_v - 1)x_v\ \}. \end{aligned} \tag{2.41}$$

Note that $\mathbf{AC}_v$ is coupled to the decisions of other slices through the feasible set $Y_v(\boldsymbol{y}^{-v})$. Thus, one cannot independently solve each slice's admission control problem to obtain an efficient solution. Furthermore, devising a

global optimization for all slices brings both complexity and nonconvexity from the BTD constraints. A natural approach requiring minimal communication and cooperation overhead is to consider a game setup where network slices are players, each seeking to maximize their carried loads (and the corresponding revenue) subject to BTD constraints.

We formally define the traffic shaping game for a set of network slices $\mathcal{V}$ as follows. We let $\boldsymbol{y} \triangleq (\boldsymbol{y}_v : v \in \mathcal{V})$ denote the simultaneous strategies of all slices (given by the respective admission control policies). As in $\mathbf{AC}_v$, each slice v picks a feasible strategy, i.e., $\boldsymbol{y}^v \in Y^v(\boldsymbol{y}^{-v})$ to minimize its objective function $\theta_v(\boldsymbol{y}^v, \boldsymbol{y}^{-v}) \triangleq x_v$. Note in the sequel we will modify $\theta_v(\cdot, \cdot)$ to ensure the games convergence. A Nash equilibrium is a simultaneous strategy $\boldsymbol{y}^*$ such that no slice can unilaterally improve its carried load, i.e., for all $v \in \mathcal{V}$

$$\theta_v(\boldsymbol{y}^{v,*}, \boldsymbol{y}^{-v,*}) \leq \theta_v(\boldsymbol{y}^v, \boldsymbol{y}^{-v,*}), \quad \forall \boldsymbol{y}^v \in Y^v(\boldsymbol{y}^{-v,*}).$$

The following result follows from Theorem 3.1 in [35].

Theorem 2.4.1. *The traffic shaping game defined above has a Nash equilibrium.*

Note that at the Nash equilibrium, no slice can unilaterally improve its performance. Therefore, finding the Nash equilibrium is also a way to achieve fairness under our sharing scheme. In the next subsection, we will design an algorithm to achieve such allocation.

2.4.1 Algorithm

In our setting finding the Nash equilibrium is not a simple matter. The difficulty arises from the fact that slices' strategy spaces depend on other's choices, so oscillation is possible. In the literature such settings are specifically referred to as Generalized Nash Equilibrium Problem (GNEP), see, e.g., [36] and [37]. However, the algorithm proposed in [36] assumed an algorithm capable of solving a penalized unconstrained Nash Equilibrium Problem, which satisfies a set of conditions, and that in [37] relies on the convexity of the joint strategy space. Thus none of them can be directly applied in our setting. Below we propose an algorithm involving slices and a central entity which is guaranteed to converge to the equilibrium.

We summarize the main ideas as follows. To decouple dependencies among strategy spaces, we shall move slice v's BTD constraint into its objective function as a penalty term with an associated multiplier λ_v. Let $\boldsymbol{\lambda} \triangleq (\lambda_v : v \in \mathcal{V})$. By adjusting the value of $\boldsymbol{\lambda}$ according to $\boldsymbol{y}$ at each iteration, one can determine a setting such that, at the induced Nash Equilibrium, all slices meet their BTD constraints, and the equilibrium is identical to that of the traffic shaping game. In addition, in order to prevent overshooting, at each iteration each slice's objective function is regularized by the distance to the previous reciprocal carried load x_v.

Specifically, the admission control strategy of slice v in response to

other slices is now given as the solution to the following optimization problem:

$$L_\epsilon^v(\boldsymbol{y};\lambda^v) = \underset{(\boldsymbol{y}^v)'\in\bar{Y}^v}{\operatorname{argmin}}\, \theta_v((\boldsymbol{y}^v)', \boldsymbol{y}^{-v};\lambda_v) + \frac{\epsilon}{2}(x_v - x_v')^2, \tag{2.42}$$

where we define a BTD penalty function for slice v as

$$h_v(\boldsymbol{y}) \triangleq f_v(\tilde{\boldsymbol{\rho}}^v;\tilde{\boldsymbol{\rho}}^{-v}) - s^v(\tilde{d}_v - 1)x_v.$$

and the objective function for slice v is now (different from what is previously defined): $\theta_v(\boldsymbol{y}^v, \boldsymbol{y}^{-v};\lambda_v) \triangleq e^{x_v} + \lambda_v[h_v(\boldsymbol{y})]_+$, with $[x]_+ \triangleq \max(0, x)$. The last term in Eq. (2.42) serves as a regularization term. The strategy space is now $\bar{Y}^v \triangleq \{\boldsymbol{y}_v | \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v\rangle = 1, \mathbf{0} \preceq \boldsymbol{M}^v\tilde{\boldsymbol{\rho}}^v \preceq x_v\mathbf{1}\}$ and x_v is substituted by e^{x_v} to ensure strong convexity, which is required for convergence (note that due to the monotonicity, e^{x_v} and x_v should result in the same optimizer).

We propose to use the inexact line search update introduced in [37]. In order to make sure the iteration is proceeding towards the equilibrium, we use

$$\begin{aligned}\Omega_\epsilon(\boldsymbol{y};\boldsymbol{\lambda}) \triangleq &\sum_{v\in\mathcal{V}} \theta_v(\boldsymbol{y}^v, \boldsymbol{y}^{-v};\lambda_v) - \theta_v(L_\epsilon^v(\boldsymbol{y};\boldsymbol{\lambda}), \boldsymbol{y}^{-v};\lambda_v) \\ &- \frac{\epsilon}{2}(x_v - x_v')^2 \geq 0\end{aligned}$$

as a metric, observing that the equilibrium is given by $\boldsymbol{y}^*$ if and only if $\Omega_\epsilon(\boldsymbol{y}^*;\boldsymbol{\lambda}) = 0$. Therefore we seek to decrease $\Omega_\epsilon(\boldsymbol{y};\boldsymbol{\lambda})$ by a sufficient amount at each iteration. The task executed by each slice v is given in Algorithm 1, while the central entity, which is responsible for collecting and delivering information and updating $\boldsymbol{\lambda}$, executes Algorithm 2. This then follows the algorithm proposed in [36].

Algorithm 1 Algorithm of Slice v

1: Set $k \leftarrow 0$ and collect ϵ from central entity.
2: Receive $\lambda_v(k)$ and $\boldsymbol{y}(k)$ from central entity.
3: Compute $L_\epsilon^v(\boldsymbol{y}(k); \boldsymbol{\lambda})$ and transmit it back to the central entity. Set $k \leftarrow k+1$. Go to step 2

Algorithm 2 Penalized Update in Central Entity

1: Choose a starting point $\boldsymbol{y}(0)$, $\boldsymbol{\lambda}(0) \succeq \mathbf{0}$, $\eta_v \in (0,1)$, for $v \in \mathcal{V}$, $\beta, \sigma \in (0,1)$, $\epsilon > 0$ but small enough (see following theorem for convergence) and set $k \leftarrow 0$.
2: If a termination criterion is met then STOP. Otherwise, communicate $\boldsymbol{y}(k)$ together with $\lambda_v(k)$ to all slices.
3: All slices compute $L_\epsilon^v(\boldsymbol{y}(k); \boldsymbol{\lambda})$ and feedback to central entity.
4: Compute $t(k) = \max\{\beta^l | l = 0, 1, 2, \dots\}$ such that if we assume $\xi(k) = (L_\epsilon^v(\boldsymbol{y}(k); \boldsymbol{\lambda}(k)) : v \in \mathcal{V}) - \boldsymbol{y}(k)$:

$$\Omega_\epsilon(\boldsymbol{y}(k) + t(k)\xi(k)) \leq \Omega_\epsilon(\boldsymbol{y}(k)) - \sigma(t(k))^2 \|\xi(k)\|. \tag{2.43}$$

Then set $\boldsymbol{y}(k+1) = \boldsymbol{y}(k) + t(k)\xi(k)$.
5: Set $I(k) = \{v | h_v(\boldsymbol{y}(k)) > 0\}$. For every $v \in I(k)$, if

$$e^{x_v(k)} > \eta_v(\lambda_v \|\nabla_{\boldsymbol{y}^v} h_v \boldsymbol{y}(k))\|), \tag{2.44}$$

then $\lambda_v(k+1) \leftarrow 2\lambda_v(k)$. Set $k \leftarrow k+1$. Broadcast $\boldsymbol{y}(k)$ and $\boldsymbol{\lambda}(k)$ to slices and go to step 2.

Theorem 2.4.2. *Let $\{\boldsymbol{y}(k)\}$ be the sequence of admission control decisions generated by Algorithm 1 and Algorithm 2, then every limit point of this sequence is a Nash equilibrium of the traffic shaping game induced by $\boldsymbol{AC}_v$.*

Proof. First we need to verify the Assumption 5.1 in [37] to guarantee that for a given $\boldsymbol{\lambda}$, step 4 in Algorithm 2 converges to a Nash equilibrium. The non-constant part of $\Psi_\epsilon(\boldsymbol{y}, \boldsymbol{y}'; \boldsymbol{\lambda})$ (defined in [37]) when $\boldsymbol{y}'$ is fixed is: $\sum_v e^{x_v} + \lambda_v[h_v(\boldsymbol{y})]_+ - \frac{\epsilon}{2}\|x_v - x'_v\|^2$. If ϵ is small enough, the concavity of the last term will be canceled out by e^{x_v}. Then the non-constant part is always convex in $\boldsymbol{y}$. Hence, the Assumption 5.1 holds true together with the propositions 2.1(a) - (d) in [37]. Therefore, the proposed algorithm generates Nash equilibrium of the game.

One can easily verify that the EMFCQ condition given by Definition 2.7 in [36] is satisfied. Thus for all v, λ_v gets updated a finite number of times. According to Theorem 2.5 in [36], the claim is true. □

2.4.2 Characterization of Traffic Shaping Equilibrium

Next we study the characteristics of the resulting traffic shaping Nash equilibrium. To make this tractable we consider networks which are saturated and subsequently (in Section 2.5) provide simulations to evaluate other settings.

Assumption 3. (Saturated Regime) *Suppose the system is such that for each*

network slice, the optimal admission control for both SCPF and SS [2] *in response to other slices' loads is such that for all* $v \in \mathcal{V}$, $\boldsymbol{a}^v \prec \mathbf{1}$.

Assumption 3 depends on many factors including the BTD constraints, the mobility patterns, and network slices' shares, but it is generally true when the exogenous traffic of all slices at all base stations γ_b^v is high. When this is the case we have the following result:

Theorem 2.4.3. *Under Assumptions 1, 2 and 3, the relative load distributions at the Nash equilibrium of the traffic shaping game* $\tilde{\boldsymbol{\rho}}^* \triangleq (\tilde{\boldsymbol{\rho}}^{v,*} : v \in \mathcal{V})$ *are the unique solution to:*

$$\min_{(\tilde{\boldsymbol{\rho}}^v \in \Gamma^v : v \in \mathcal{V})} \left\| \sum_v s^v \tilde{\boldsymbol{\rho}}^v \right\|_2^2 + \sum_v (s^v)^2 \|\tilde{\boldsymbol{\rho}}^v\|_2^2, \tag{2.45}$$

where $\Gamma^v \triangleq \{\ \tilde{\boldsymbol{\rho}}^v \mid \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle = 1, \boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v \succeq \mathbf{0}\ \}$, *and the associated carried load for slice* v *is* $\rho^{v,*} = \frac{s^v(\tilde{d}_v - 1)}{\langle \tilde{\boldsymbol{g}}^*, \tilde{\boldsymbol{\rho}}^{v,*} \rangle}$, *where* $\tilde{\boldsymbol{g}}^*$ *corresponds to the overall share weighted relative loads distributions at the equilibrium.*

Remark: Theorem 2.4.3 also holds true under a more general traffic model, see, for example, Section 2.6.2.

Proof. Under the saturated regime, BTD constraint of each $v \in \mathcal{V}$ must be binding because we are blocking traffics. Thus we have: $x_v = \frac{f_v(\tilde{\boldsymbol{\rho}}^v; \tilde{\boldsymbol{\rho}}^{-v})}{s^v(\tilde{d}_v - 1)}$, Also

[2] Admission control under SS is defined in the sequel.

Assumption 3 guarantees that for all $v \in \mathcal{V}$, $\boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v \preceq x_v \mathbf{1}$ is not binding then the $\mathbf{AC}_v$ can be reformulated as:

$$\min_{\tilde{\boldsymbol{\rho}}^v \in \Gamma^v} \quad s^v f_v(\tilde{\boldsymbol{\rho}}^v; \tilde{\boldsymbol{\rho}}^{-v}), \tag{2.46}$$

A constant factor s^v in the objective function has no impact to the minimizer. We can show that the optimality condition of Problem (2.45) is the same as each slice v optimizing its own problem given by Eq. (2.46). Dividing the objective function by 2, the Lagrangian of Problem (2.45) is:

$$\begin{aligned} L(\tilde{\boldsymbol{\rho}}; \boldsymbol{\zeta}, \boldsymbol{\chi}) = \frac{1}{2}\Big(\Big\|\sum_v s^v \tilde{\boldsymbol{\rho}}^v\Big\|_2^2 + \sum_v (s^v)^2 \|\tilde{\boldsymbol{\rho}}^v\|_2^2\Big) \\ + \sum_{v \in \mathcal{V}} \zeta_v(\langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle - 1) - \sum_{v \in \mathcal{V}} (\boldsymbol{\chi}^v)^T \boldsymbol{M}_v \tilde{\boldsymbol{\rho}}^v, \end{aligned} \tag{2.47}$$

where $\tilde{\boldsymbol{\rho}} \triangleq (\tilde{\boldsymbol{\rho}}^v : v \in \mathcal{V})$, dual variables $\boldsymbol{\zeta} \triangleq (\zeta_v : v \in \mathcal{V})$, and $\boldsymbol{\chi} \triangleq (\boldsymbol{\chi}^v : v \in \mathcal{V})$. According to the KKT condition, the solution $\tilde{\boldsymbol{\rho}}^*$ must be such that, for all $v \in \mathcal{V}$:

$$\nabla_{\tilde{\boldsymbol{\rho}}^v} L(\boldsymbol{\rho}^*; \boldsymbol{\zeta}^*, \boldsymbol{\chi}^*) = (s^v)^2 \tilde{\boldsymbol{\rho}}^{v,*} + s^v \tilde{\boldsymbol{g}} + \zeta_v^* \mathbf{1} - (\boldsymbol{M}^v)^T \boldsymbol{\chi}^{v,*} = \mathbf{0}, \tag{2.48}$$

and $\boldsymbol{\chi}^{v,*} \succeq \mathbf{0}, \tilde{\boldsymbol{\rho}}^{v,*} \in \Gamma^v$. The Lagrangian of Problem (2.46) is:

$$L_v(\tilde{\boldsymbol{\rho}}^v; \zeta_v, \boldsymbol{\chi}^v) = \langle \tilde{\boldsymbol{g}}, s^v \tilde{\boldsymbol{\rho}}^v \rangle + \zeta_v(\langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle - 1) - (\boldsymbol{\chi}^v)^T \boldsymbol{M}_v \tilde{\boldsymbol{\rho}}^v. \tag{2.49}$$

If slice v's relative load $\tilde{\boldsymbol{\rho}}^{v,*}$ optimizes Problem (2.46) given other slices' $\tilde{\boldsymbol{\rho}}^{-v,*}$, following KKT condition should be met:

$$\nabla_{\tilde{\boldsymbol{\rho}}^{v,*}} L_v(\tilde{\boldsymbol{\rho}}^{v,*}; \zeta_v^*, \boldsymbol{\chi}^{v,*}) = (s^v)^2 \tilde{\boldsymbol{\rho}}^{v,*} + s^v \tilde{\boldsymbol{g}} + \zeta_v^* \mathbf{1} - (\boldsymbol{M}_v^{-1})^T \boldsymbol{\chi}^{v,*} = \mathbf{0}, \tag{2.50}$$

and $\boldsymbol{\chi}^{v,*} \succeq \mathbf{0}, \tilde{\boldsymbol{\rho}}^{v,*} \in \Gamma^v$, which is exactly the same as the KKT condition of Problem (2.45). Therefore, Problem (2.45) is solved at the Nash equilibrium. Moreover, we could compute the total carried load of slice v by setting $f_v(\tilde{\boldsymbol{\rho}}^{v,*}; \tilde{\boldsymbol{\rho}}^{-v,*}) = s^v(\tilde{d}_v - 1)/\rho^{v,*}$, which gives us $\rho^{v,*} = \frac{s^v(\tilde{d}_v - 1)}{\langle \tilde{\boldsymbol{g}}^*, \tilde{\boldsymbol{\rho}}^{v,*} \rangle}$. □

The first term in the objective function in Eq. (2.45) rewards balancing the overall share weighted relative loads on network. The second term rewards a slice for balancing its own relative loads. The Nash equilibrium in the saturated regime is thus a compromise between those two objectives while constrained by the network slices mobility patterns and feasible admission control policies. Note that as long as $\tilde{\rho}_b^v > 0, \forall v \in \mathcal{V}, b \in \mathcal{B}$, GPS and SS are approximately the same under heavy load. Therefore, we use SS as the benchmark to characterize the carried load at the Nash equilibrium under SCPF.

Admission control for slice v under SS (ACSS$_v$): Under SS slice v can determine its optimal admission control $\boldsymbol{y}^v$ by solving:

$$\begin{aligned} \max_{\tilde{\boldsymbol{\rho}}^v, \rho^v} \quad & \rho^v \\ \text{s.t.} \quad & \boldsymbol{a}^v = \rho^v \boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v, \quad \boldsymbol{a}^v \in [0,1]^B \\ & \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle = 1 \text{ and } \rho^v \|\tilde{\boldsymbol{\rho}}^v\|_2^2 \leq (s^v \tilde{d}_v - 1). \end{aligned}$$

Note slices' admission control decisions are clearly decoupled under SS, but paralleling Theorem 2.4.3 we have following result.

Theorem 2.4.4. *Under Assumptions 1 and 3, the optimal admission control policy under SS are decoupled. The optimal choice for slice v, $\tilde{\boldsymbol{\rho}}^{v,SS,*}$, is the unique solution to:*

$$\min_{\tilde{\boldsymbol{\rho}}^v \in \Gamma^v} \|\tilde{\boldsymbol{\rho}}^v\|_2^2, \tag{2.51}$$

and the associated carried load is given by $\rho^{v,SS,*} = \frac{s^v \tilde{d}_v - 1}{\|\tilde{\boldsymbol{\rho}}^{v,SS,*}\|_2^2}$.

Proof. Under the saturated regime, BTD constraint of each $v \in \mathcal{V}$ must be binding. Thus we have: $\rho^v = \frac{s^v \tilde{d}_v - 1}{\|\tilde{\boldsymbol{\rho}}^v\|_2^2}$. Moreover, the constraint $\boldsymbol{a}^v \preceq 1$ should be satisfied with strict inequality, thus the optimal policy $\tilde{\boldsymbol{\rho}}^{v,SS,*}$ under SS is given by:

$$\min_{\tilde{\boldsymbol{\rho}}^v} \{ \ \|\tilde{\boldsymbol{\rho}}^v\|_2^2 \mid \boldsymbol{M}^v \tilde{\boldsymbol{\rho}}^v \succeq 0, \quad \langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle = 1 \}.$$

Then the optimal load is obtained by plugging the result in the BTD constraint. □

By comparing Eq. (2.45) and Eq. (2.51), one can see that under SS, slices simply seek to balance their own relative loads on the network. By taking the ratio between ρ_v^* and $\rho_v^{SS,*}$ given in Theorem 2.4.3 and 2.4.4, one can show that under Assumptions 1, 2, and 3 the gain in carried load for slice v is given by

$$G_v^{\text{load}} \triangleq \frac{\rho^{v,*}}{\rho^{v,SS,*}} = \frac{\|\tilde{\boldsymbol{\rho}}^{v,SS,*}\|_2^2}{\langle \tilde{\boldsymbol{g}}^*, \tilde{\boldsymbol{\rho}}^{v,*} \rangle} \times \frac{s^v(\tilde{d}_v - 1)}{s^v \tilde{d}_v - 1}. \tag{2.52}$$

The first factor captures a traffic shaping dependent gain for slice v. The second factor is a result of statistical multiplexing gains. A simple special case is highlighted in the following corollary.

Corollary 2.4.5. *Under Assumptions 1, 2 and 3, if user mobility patterns are such that* $\frac{1}{B}\mathbf{1} \in \Gamma^v$ *for all* $v \in \mathcal{V}$*, the gain in the total carried load under the SCPF traffic shaping Nash equilibrium vs. optimal admission control for SS is given by:*

$$G_v^{\text{load}} = \frac{s^v \tilde{d}_v - s^v}{s^v \tilde{d}_v - 1} \geq 1, \quad \forall v \in \mathcal{V}. \tag{2.53}$$

Proof. Under SS, it is obvious that the optimal relative load distribution of slice v is $\tilde{\boldsymbol{\rho}}^{v,SS,*} = \frac{1}{B}\mathbf{1}$. Plugging it in the BTD constraint one can get $\rho^{v,SS,*} = B(s^v \tilde{d}_v - 1)$, thus $\boldsymbol{\rho}^{v,SS,*} = (s^v \tilde{d}_v - 1)\mathbf{1}$. Dividing the objective function by s^v and discarding the routing constraints, the Lagrangian of Eq. (2.46) is: $\langle \tilde{\boldsymbol{g}}^v, \tilde{\boldsymbol{\rho}}^v \rangle + \nu(\langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^v \rangle - 1)$, where ν is the dual variable. Solving its KKT condition we have:

$$\tilde{\boldsymbol{\rho}}^{v,*} = -\frac{1}{2s^v}\left(\nu\mathbf{1} + \sum_{v' \neq v} s_{v'} \tilde{\boldsymbol{\rho}}^{v'}\right). \tag{2.54}$$

Substituting in $\langle \mathbf{1}, \tilde{\boldsymbol{\rho}}^{v,*} \rangle = 1$ we have $\nu = -\frac{s^v+1}{B}$. When $\frac{1}{B}\mathbf{1} \in \Gamma^v, \forall v \in \mathcal{V}$, Eq. (2.54) implies that if all other slices $v' \neq v$ pick their relative loads as $\frac{1}{B}\mathbf{1}$, then $(\sum_{v' \neq v} s^{v'} \tilde{\boldsymbol{\rho}}^{v'}) \parallel \mathbf{1}$, meaning that this is the Nash equilibrium of the game. Note that since $\tilde{\boldsymbol{\rho}}^{v,*}$ is feasible and optimal for a relaxed feasible set, it will still be optimal if we put back the routing constraints. Thus we have that: $\langle \tilde{\boldsymbol{g}}^*, \tilde{\boldsymbol{\rho}}^{v,*} \rangle = \frac{1}{B}$. Then the carried load gain is obtained by plugging the result in Eq. (2.52). □

Note that in order for a BTD constraint to be feasible under SS, one must require $s^v \tilde{d}_v > 1$. It can be seen that the gain exhibited in Corollary 2.4.5 can be very high when $s^v \downarrow 1/\tilde{d}_v$. Furthermore, if $s^v \uparrow 1$ we have that $G_v^{\text{load}} \downarrow$

1, i.e., no actual gain. This result implies that slices with small shares or tight BTD constraints will benefit most from sharing, coinciding with our observations in Corollary 2.2.4.

2.5 Performance Evaluation Results

In this section, we validate the theoretical results in previous sections, and provide quantitative characterizations via numerical experiments. We simulated a wireless network shared by multiple slices supporting mobile customers following the IMT-Advanced evaluation guidelines [38]. The system consists of 19 base stations in a hexagonal cell layout with an inter site distance of 200 meters and 3 sector antennas, mimicking a dense 'small cell' deployment. Thus, in this system, $\mathcal{B}$ corresponds to 57 sectors. Users associate to the sector offering the strongest SINR, where the downlink SINR is modeled as in [39]:

$$\text{SINR}_{ub} = \frac{P_b G_{ub}}{\sum_{k \in \mathcal{B}, k \neq b} P_k G_{uk} + \sigma^2},$$

where, following [38], the noise $\sigma^2 = -104$dB, the transmit power $P_b = 41$dB and the channel gain between user u and BS sector b, denoted by G_{ub}, accounts for path loss, shadowing, fast fading and antenna gain. Letting d_{ub} denote the current distance in meters from the user u to sector b, the path loss is defined as $36.7 \log_{10}(d_{ub}) + 22.7 + 26 \log_{10}(f_c)$dB, for a carrier frequency $f_c = 2.5$GHz. The antenna gain is set to 17 dBi, shadowing is updated every second and modeled by a log-normal distribution with standard deviation of 8dB, as in [39]; and fast fading follows a Rayleigh distribution depending on the mobile's speed and the angle of incidence. The downlink rate c_u currently achievable to user u is based on discrete set modulation and coding schemes (MCS) and associated SINR thresholds given in [40]. This MCS value is selected based on the averaged $\overline{\text{SINR}}_{ub}$, where channel fast fading is averaged over a second.

We model slices' with different spatial loads by modeling different customer mobility patterns. Roughly uniform spatial loads are obtained by simulating the Random Waypoint model [41], while non-uniform loads obtained by simulating the SLAW model [42]. Instead of the open network assumed in the theoretical analysis, in the simulations we use a closed network where the total number of users on each slice keeps fixed. Moreover, the simulated mobility models would not induce Markovian motion amongst base stations assumed in our analysis, yet the analytical results are robust to these assumptions.

2.5.1 Statistical Multiplexing and BTD Gains

We evaluated the BTD gains of SCPF vs. both SS and GPS for four simulation scenarios, each including 4 slices, each with equal shares but different spatial load patterns. For each scenario, we provide results for simulated BTD gains, and results from our theoretical analysis (Corollary 2.2.4 and Corollary 2.2.5) based on the empirically obtained spatial traffic loads. More detailed information regarding simulated scenarios and resulting empirical spatial traffic loads for high load regime are displayed in Table 2.2 and a snapshot of locations for the 4 slices' users in a network with a load of 4 users per sector is displayed in Figure 2.1.

The results given in Figure 2.2 show the BTD gains over SS for each scenario as the overall network load increases. In Scenario 3, the aggregate network traffic is 'smoother' than the individual slice's traffic, and the gains are indeed higher. This is also the case for Slice 1 and 2 in Scenario 4, since these

Scenario Slices	Spatial loads	$\|\tilde{\boldsymbol{\rho}}^v\|_2$	$\|\tilde{\boldsymbol{g}}\|_2$	$\theta(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{\rho}}^v)$	$G_v^{SS,H}$
1 Homogeneous	uniform.	0.27	0.27	7.09	1.01
2 Homogeneous	non-uniform	0.32	0.32	6.18	1.01
3 Heterogeneous	orthogonal	0.36	0.26	41.78	1.83
4 Mixed Slices	1&2 non-uniform	0.36	0.23	25.52	1.70
	3&4 uniform	0.19	0.23	48.00	1.24

Table 2.2: Measured normalized slice and network traffic norms and angles for highest load case of each scenario.

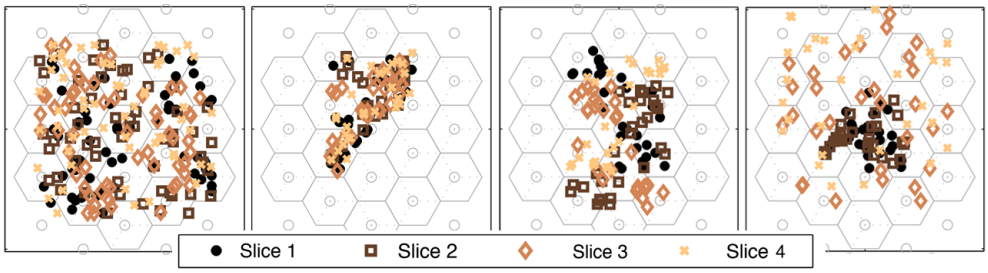


Figure 2.1: Snapshot of users positions per slice and scenario exhibiting the different characteristics of traffic spatial loads. Left to right: Scenarios 1 to 4.

slices loads are more 'imbalanced' than the other two slices, they experience higher gains. In Scenario 2, where slices non-homogenous spatial loads are 'aligned', aggregation does not lead to smoothing and the gains are least.

Similarly, the results given in Figure 2.3 show the BTD gains over GPS for each scenario as the overall network load increases. As can be seen, the gain is not necessarily monotonic in the load. In Scenario 4, the Slice 1 and 2 have significant gains because their loads are more imbalanced, while Slice 3 and 4 see negative gains. However, the overall gain defined in Eq. (2.19) is still positive, ranging from 1.26 to 1.5 for varying overall load. As discussed

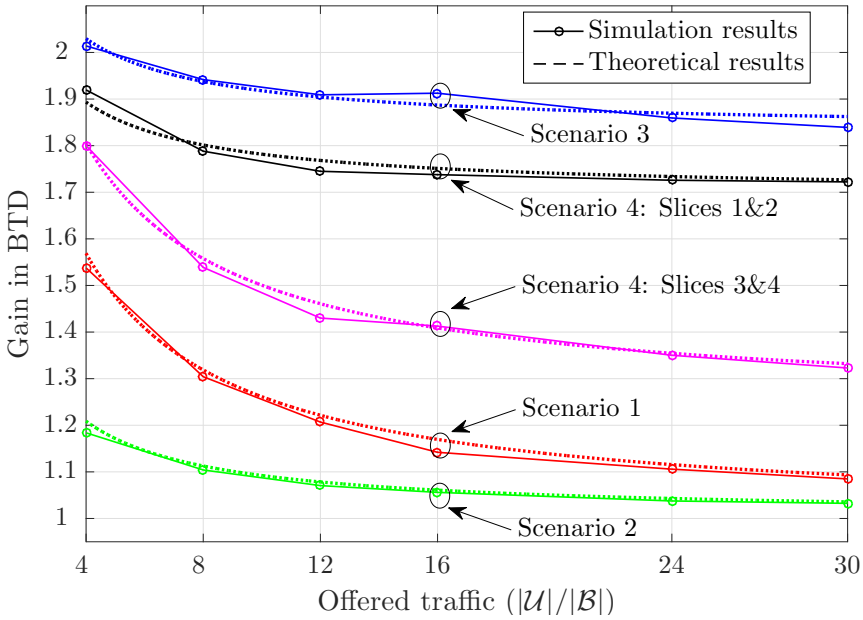


Figure 2.2: BTD gain over SS for our 4 different scenarios.

in Section 2.2.2.2, Slice 3 and 4 observe negative gains. Intuitively in this setting this is due to the slices with homogeneous loads not being sufficiently protected (under SCPF) of slices with concentration of loads on a small set of base stations.

As can be seen in Figures 2.2 and 2.3 the simulated and theoretical gains (dashed lines) of Corollary 2.2.4 are an excellent match. The theoretical model has been calibrated to the mean reciprocal capacities seen by slice customers (i.e., δ_b^y's) and the measured induced loads resulting from the slice mobility patterns.

In addition to performance averaged over time, to illustrate the dynamic

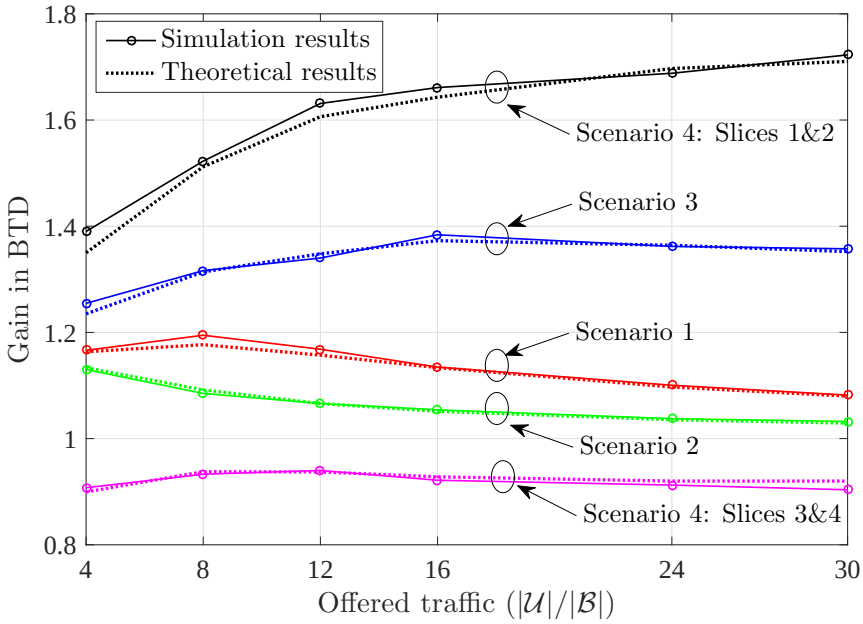


Figure 2.3: BTD gain over GPS for our 4 different scenarios.

of the BTD perceived by a typical user, we plot the BTD vs. time for a randomly picked user on Slice 1 in Scenario 1, as shown in Fig. 2.4, where the left/right part is under light/heavy load regime, respectively. Note that under heavy load, GPS and SS are approximately the same. SCPF outperforms for most of the time. Under light load, the mean BTD under SCPF is 4.2044, while that under SS (GPS) is 6.6157 (5.2862), respectively. The standard deviation of BTD under SS (GPS) is 3.9011 (3.0449), and SCPF reduces it to 1.93. Similar phenomenon is observed under heavy load, when both SS and GPS provide mean BTD of 19.65 and associated standard deviation of 13.79, SCPF reduce them to 16.79 and 13.45, respectively. Therefore, SCPF can effectively improve the perceived BTD and also 'smooth' the user perceived

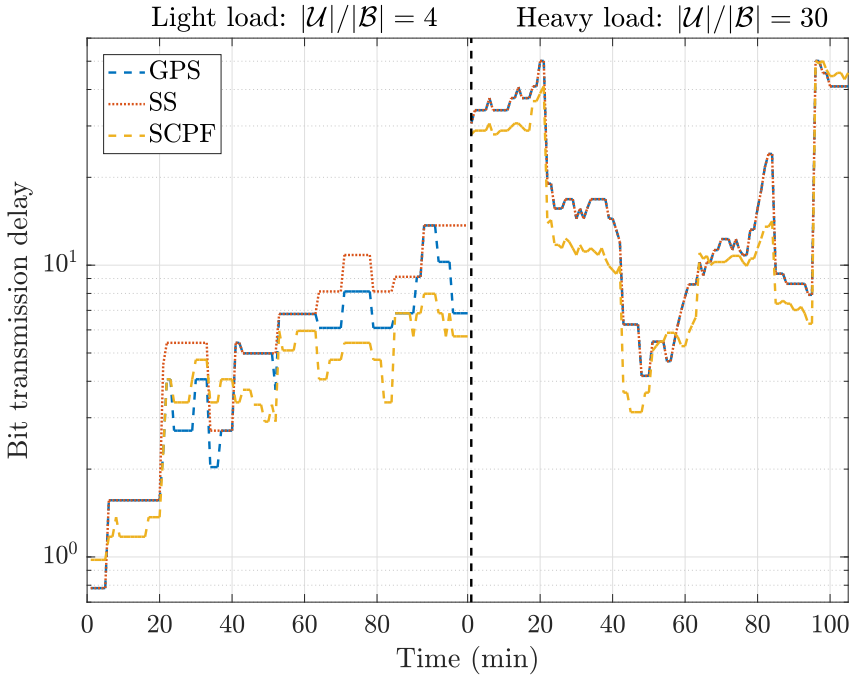


Figure 2.4: BTD vs. time for a randomly picked user under Scenario 3

QoS.

2.5.2 Traffic Shaping Equilibrium and Carried Load Gains

In order to study the equilibria reached by the traffic shaping game, we measured the underlying user mobility patterns in Section 2.5.1, and modeled it via a random routing matrix. We further assumed uniform intensity of arrivals rates at all base stations and uniform exit probabilities of 0.1. The mean holding time at each base station was again calibrated with the simulations in Section 2.5.1. We considered a traffic shaping game for a network shared by 3 slices, where Slice 1 has uniform spatial loads and Slice 2 and 3

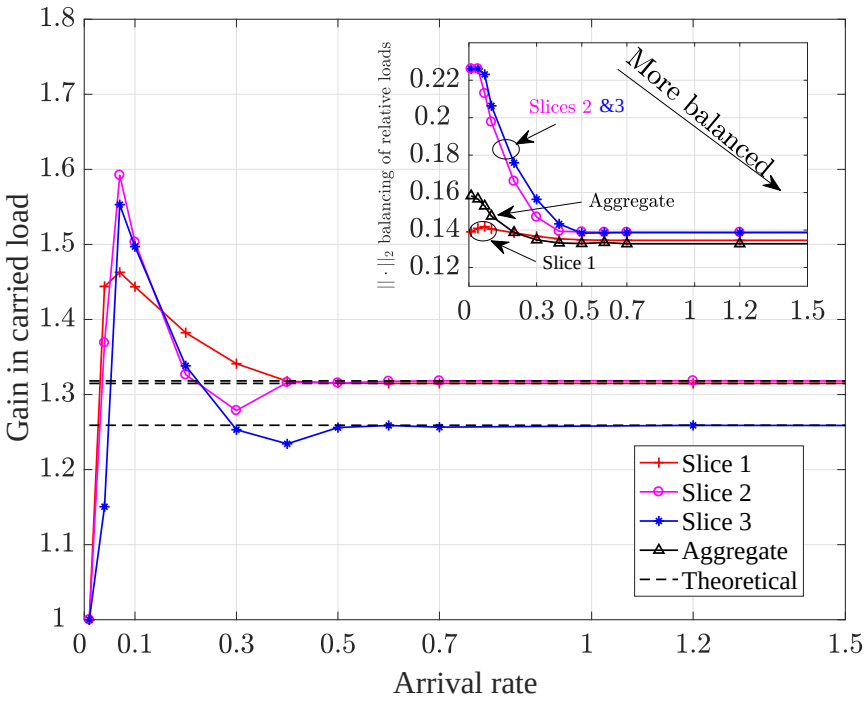


Figure 2.5: Gain in carried load for various arrival rates.
Subfigure: Balancing in relative load.

have different non-uniform spatial loads. All slices have equal shares and their capacity normalized BTD requirements are set to $\tilde{d}_1 = 10, \tilde{d}_2 = 12, \tilde{d}_3 = 15$ respectively. The Nash equilibrium was solved via the algorithm included in Section 2.4.1. The convergence is reached within 3 rounds of iterations under the parameters $\eta_v = \beta = 0.5, \forall v \in \mathcal{V}$, $\sigma = 0.1, \epsilon = 0.01$.

The results shown in Figure 2.5 exhibit dashed lines corresponding to the theoretical carried load gains in the saturated regime. As can be seen, these coincide with the Nash equilibria of the simulated traffic shaping games for high arrival rates. For lower arrival rates the gains can be much higher, e.g.,

almost a factor of 1.6, for slices with non-uniform mobility patterns. This was to be expected since for lower loads we expect higher statistical multiplexing gains from sharing, and thus relatively higher carried loads to be admitted. For very low loads, as expected, there are no gains since all traffic can be admitted and BTD constraints are met.

Also shown in Figure 2.5(subfigure) is the degree to which the relative loads of slices, and the weighted aggregate traffic on the network $\tilde{\boldsymbol{g}}$ are balanced, as measured by $||\cdot||_2$, as the arrival rates on the network increase. As expected, based on Theorem 2.4.3, as arrivals increase relative loads of slices and the network become more balanced, showing the compromise the traffic shaping game is making, balancing slices relative loads and that of the overall network.

2.6 Extensions and Generalizations

The results in this chapter can be generalized in several ways. In this section we discuss some possibilities.

2.6.1 General User Activity Model

Our framework so far considered dynamic user arrivals where users were active throughout their sojourn in the system. A possible extension is to consider user activity models which alternate between active/inactive states during their sojourn as long as this is independent of their service, e.g., the activity patterns in an ongoing voice call experiencing adequate QoS. For example if all customers on slice v are active with probability p_v, the previously developed results hold true with the additional thinning of slice loads on various base stations by p_v.

Another alternative is to consider network slices that support fixed collections of users which exhibit on-off behavior. This might be the case for slices provisioned to support fixed Internet of Things (IoT) devices/users. We can model each user on slice v as having an on-off process which is independent to all other users and at a given time t, the probability of being active is p_v for all slice v users. Each user is associated with a fixed base station. Assuming that the total number of users on slice v at base station b is denoted by ρ_b^v, the number of active user on slice v at base station b can be modeled by a random variable $N_b^v \sim \text{Binomial}(\rho_b^v, p_v)$. Since the sum of two independent Binomial random variables with the same 'success probability' is again Binomial, the

result in Lemma 2.2.3 still holds true. The results regarding BTD and gains follow with loads thinned by the activity factor p_v. In this setting a slice operator may do network dimensioning by deciding how many devices its slice might support on each base station, i.e., ρ_b^v for each b. A simple admission control policy can also be studied wherein when a user wakes up from the inactive mode, the slice determine if it may access the network (independently of the state of the system) based on an admission probability a_b^v for slice v at base station b. In this setting performance management of all slices in the network can be similarly modeled as a game as before.

2.6.2 Multi-Class Routing

The proposed framework can also be developed for a more general routing model to capture user mobility. In general the mobility pattern of users can be more complex, exhibiting dependencies across base stations, e.g., a user may be driving along a highway. The random routing model studied so far might not be accurately reflecting the movement of such customers including dependency introduced by the underlying traffic infrastructure, or commuting patterns of different customers. However more complex routing behavior that is not state dependent can in general be addressed using a multi-class routing model.

In the multi-class setting, for each slice, say v, we can define a set of user classes $\mathcal{K}_v$, where a class k user will traverse a specific sequence of base stations, denoted by $\mathcal{R}_k = (b_{k,1}, b_{k,2}, \ldots, b_{k,|\mathcal{R}_k|})$ – we will assume for simplicity each

class traverses a given base station at most once. Upon finishing traversing the base stations in $\mathcal{R}_k$, the user can randomly change to another class or leave the system. Paralleling the random routing scenario, we can redefine the routing matrix $\boldsymbol{Q}^v$ with components Q_{ij}^v corresponding to the probability of a customer completing the path for class i becomes a class j customer. Letting $\boldsymbol{\gamma}^v = (\gamma_k^v : k \in \mathcal{K}_v)$ denote the arrival rate to slice v for each class. Following our previous framework, we can define an admission control vector $\bar{\boldsymbol{a}}^v$ for arriving customers in each class. Then $(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)^{-1}\text{diag}(\boldsymbol{\gamma}^v)\bar{\boldsymbol{a}}^v$ is the vector of intensities of customer flow of each class. Thus if we define $\mu_{b,k}^v$ as the mean sojourn times of class k customer at base station b, and a matrix $\boldsymbol{\mu}^v = (\mu_{b,k}^v : \ b \in \mathcal{B}, k \in \mathcal{K}_v)$, we have that $\boldsymbol{\rho}^v = \boldsymbol{\mu}^v(\boldsymbol{I} - (\boldsymbol{Q}^v)^T)^{-1}\text{diag}(\boldsymbol{\gamma}^v)\boldsymbol{a}^v = \bar{M}^v\boldsymbol{a}^v$ is the induced load at each base station after admission control. Here we use the convention that if a base station $b \notin \mathcal{R}_k$, $\mu_{b,k}^v = 0$, thus the flow of customers belonging to class k will not place any load at base station b. Clearly random routing is simply a special case of this more general model wherein each $\mathcal{R}_k$ contains one base station.

Also note that $\bar{\boldsymbol{M}}^v$ is a fat matrix thus it might be the case that multiple $\bar{\boldsymbol{a}}^v$ can result in one $\boldsymbol{\rho}^v$. However, we can still work with $\boldsymbol{\rho}^v$ in order to get a convex problem.

Multi-class routing admission control for slice v under SCPF (MultiAC$_v$): Given other slices' load distribution decisions $\boldsymbol{\rho}^{-v}$, slice

v determines its load distribution policy $\boldsymbol{\rho}^v$ by solving

$$\max_{\boldsymbol{\rho}^v}\{\ \mathbf{1}^T\boldsymbol{\rho}^v \mid \boldsymbol{\rho}^v \in Y^v(\boldsymbol{\rho}^{-v})\ \} \tag{2.55}$$

where $Y^v(\boldsymbol{\rho}^{-v})$ denotes slice v's feasible policies and is given by

$$Y^v(\boldsymbol{\rho}^{-v}) \triangleq \{\ \boldsymbol{\rho}^v \mid \boldsymbol{\rho}^v \in \Xi^v,\ \tilde{\boldsymbol{g}}^T\boldsymbol{\rho}^v \leq s_v(\tilde{d}_v - 1)\ \}, \tag{2.56}$$

where Ξ^v is defined as following set of load distributions:

$$\begin{aligned}\Xi^v = \{\ \boldsymbol{\rho}^v \mid \exists \bar{\boldsymbol{a}}^v \text{ s.t. } \mathbf{0} \preceq \bar{\boldsymbol{a}}^v \preceq \mathbf{1} \text{ and} \\ \boldsymbol{\rho}^v = \bar{\boldsymbol{M}}^v\bar{\boldsymbol{a}}^v\ \}.\end{aligned} \tag{2.57}$$

The set Ξ^v is actually the projection of a cube to a lower dimension hyperplane, thus must be convex. Therefore the resulting admission control problem is a convex one. Each slice can solve such convex problem by any existing algorithm, then find the corresponding admission control policy $\bar{\boldsymbol{a}}^v$ through solving a (possibly underdetermined) linear system.

As discussed above, adding multi-class routing can increase the accuracy of the model in the sense that it can better predict the load distribution after admission control. However it is at the cost of increasing the complexity of the problem: Ξ^v can be hard to determine and mapping from ρ^v back to $\bar{\boldsymbol{a}}^v$ might take extra efforts.

Chapter 3

Network Slicing in Generalized setting - Coupled Resources

3.1 Introduction

3.1.1 Background and Motivation

In the previous chapter we studied the network slicing problem in a setting where users' resource demands are decoupled, i.e., users only require a single resource type. However as the wireless communication technology evolves, next generation networks seek to support a variety of data-intensive services and applications, such as self-driving cars, infotainment, augmented/virtual reality [43], Internet of things [44, 43], and mobile data analytic [45, 46], which, probably require the availability of heterogeneous resources at the network "edge". Thus, our previous setup needs to be generalized to more complex settings where heterogeneous resources are provisioned simultaneously. Still, shared resources are provisioned in slices to different ser-

This chapter was partially included in the following paper. J. Zheng and G. de Veciana, Elastic multi-resource network slicing: Can protection lead to improved performance? in proceeding of *WiOpt'19*, 2019. The author was responsible for developing those analytic results, conducting simulation-based evaluation, and writing the paper.

vices/applications, so that customization could be performed in accordance to service providers'/tenants' requirements.

The ability to support slice-based provisioning is central to enabling service providers to take control of managing performance of their own dynamic and mobile user populations. This also improves the scalability by reducing the complexity of performance management on multi-service platforms. The ability to efficiently share network/compute resources is also key to reducing the cost of deploying such services. By contrast with today's cloud computing platforms, our focus is on provisioning slices of edge resources to meet mobile users/devices requirements. In general, shared edge resources will have smaller overall capacity resulting in reduced statistical multiplexing and making efficiency critical. Perhaps similarly to cloud computing platforms, providers/tenants will want to make long-term provisioning commitments enabling predictable costs and resource availability, yet benefit, when possible, of elastic resource allocations aligned with spatial variations in their mobile workloads but not at the expense of other slices. Thus a particularly desirable feature is to enable slice-level provisioning agreements which achieve inter-slice protection, load-driven elasticity and network efficiency.

These challenges distinguish our work from previous research in areas including engineering, computer science and economics. The standard framework used in communication networks is utility maximization (see e.g., [13] and references therein), which has led to the design of several transport and scheduling mechanisms and criteria, e.g., the widely discussed proportional

fairness. When considering dynamic/stochastic networks, e.g., [20], [19], researchers have studied networks where users are allocated resources based on utility maximization and studied requirements for network stability for 'elastic' user demands, e.g., file transfers. This body of work emphasizes user-level resource allocations, without specifically accounting for interactions among slices. Thus, it does not directly address the requirements of network slicing.

Instead in this chapter, we propose a novel approach, namely, *Share Constrained Slicing (SCS)*, wherein each slice is assigned a share of the overall resources, and in turn, distributes its share among its users. Then the user level resource allocation is determined by maximizing a sharing criterion. When SCS is applied to a setting where each user only demands one resource, for example, slices sharing wireless resources in cellular networks [17, 47, 7], it can be viewed as a Fisher market where agents (slices), which are share (budget) constrained, bid on network resources, see, e.g., [14], and for applications [15, 16, 17]. However, those works do not deal with settings where users require heterogeneous resources, and how to orchestrate slice-level interactions on different resources is not clear yet.

When it comes to sharing on heterogeneous resources, a simple solution is static partitioning of all resources according to a service-level agreement, see, e.g., [11]. It offers each slice a guaranteed allocation of the network resources thus in principle provides ideal protection among slices. However, it falls short from the perspective of providing load-driven elasticity to a slice's users, possibly resulting in either resource under-utilization or over-booking. Other

natural approaches include full sharing [4], where users from all slices are served based on some prioritizing discipline without prior resource reservation. Such schemes may not achieve slice-level protection and are vulnerable to surging user traffics across slices.

Additionally, many resource sharing schemes have been proposed for cluster computing where heterogeneous resources are involved, including Dominant Resource Fairness (DRF)[3], Competitive Equilibrium from Equal Income (CEEI) [48][49][50], Bottleneck Max Fairness (BMF) [51], etc. These allocation schemes are usually based on modelling joint resource demands of individual users, but lack of the notion of slicing, thus it is not clear how to incorporate the need to enable slice-level long-term commitments. In these works, inter-slice protection and elasticity of allocations have not been characterized. Furthermore, most of these works are developed under the assumption that users are sharing a centralized pool of resources. In this chapter we focus on a settings where resources are distributed, and mobile users are restricted to be served by proximal edge resources.

3.1.2 Contributions

The novelty of our proposed approach lies in maintaining slice-level *long-term commitments* defined by a service-level agreement, while enabling user-level resource provisioning which is driven by dynamic user loads. We consider a model where users possibly require heterogeneous resources in different proportions, and the processing rate of a user scales linearly in the amount of

resources it is allocated. Such a model captures tasks/services which speeds up in the allocated resources, which is discussed further in the sequel.

We show that SCS can capture inter- and intra-slice fairness separately. When viewed as a resource sharing criterion, SCS is shown to satisfy a set of axiomatically desirable properties akin to those in [52], and can be interpreted as achieving a tunable trade-off among inter-slice fairness (which can be seen as a proxy of protection), intra-slice fairness, and overall utilization. Fairness is connected to load-driven elasticity through share constrained weight allocation. The merits of SCS are demonstrated in both static and dynamic settings. In static settings, we prove a set of desirable properties of SCS as a sharing criterion, including slice-level *protection* and *envyfreeness*, and we demonstrate the feasibility of using a simpler (dynamically) weighted max-min as a surrogate resource allocation scheme for the cases where the cost of implementing SCS is excessive. In a dynamic settings, we consider the elastic traffic model where each user carries a fixed workload, and leaves the system once the work is processed. We model such system as a stochastic queuing network, and establish its stability condition.

Finally, and perhaps surprisingly, we show via extensive simulations that while SCS provides inter-slice protection, it can also achieve improved average job delay and/or perceived throughput, as compared with multiple variations of traditional (weighted) max-min fair allocations but without share-constrained weight allocation. We provide a heuristic explanation of such improvement that SCS can separate the busy-periods of different slices, thus

reduces inter-slice contention, and validate the explanation through simulations.

3.2 Fairness in Network Slicing

In this section we will briefly introduce the overall framework for resource allocation to network slices, namely, Share Constrained Slicing (SCS) where each slice manages a possibly dynamic set of users. Specifically, we will consider resource allocation driven by the maximization of an objective function geared at achieving a trade-off between overall efficiency and fairness [52].

To begin, we consider the set of active users on each slice to be fixed. Let us denote the set of slices by $\mathcal{V}$, the set of resources by $\mathcal{R}$, each with a capacity normalized to 1. Each slice v supports a set of user classes, denoted by $\mathcal{C}^v$, and the total set of user classes is defined as $\mathcal{C} := \cup_{v\in\mathcal{V}}\mathcal{C}^v$. For simplicity, we let $v(c)$ denote the slice which supports class c. We let $\mathcal{U}_c$ denote the set of users of class c, and the users on slice v is denoted by $\mathcal{U}^v := \cup_{c\in\mathcal{C}^v}\mathcal{U}_c$. Also, the overall set of users is $\mathcal{U} := \cup_{v\in\mathcal{V}}\mathcal{U}^v$. For each user, possibly heterogeneous resources are required to achieve certain processing rate. Let us denote the processing rate seen by user u by λ_u. We also define the resource demand vector of user class c as $\mathbf{d}_c := (d_c^r : r \in \mathcal{R})$, where d_c^r is the fraction of resource r required by user $u \in \mathcal{U}_c$ for a unit processing rate, i.e., to achieve $\lambda_u = 1$, we need to allocate fraction d_c^1 of the total amount of resource 1 to u, d_c^2 of the total amount of resource 2 to u, and so on. If a user class c does not use a given resource r then $d_c^r = 0$. Note that if two slices support users with the same requirements, we will distinguish them by defining two distinct user classes one for each slice. In other words, it is possible to have more than one user

classes with exactly the same $\mathbf{d}_c$. Also, we let $\mathcal{R}_c$ denote the set of resources required by users of class c, and in turn, let the set $\mathcal{C}_r$ denote user classes using resource r. Among $\mathcal{C}_r$, the set of classes on slice v is $\mathcal{C}_r^v := \mathcal{C}_r \cap \mathcal{C}^v$. The number of active users of class c at time t is denoted by a random variable $N_c(t)$, and that on slice v by $N^v(t)$. $N^v(t) = \sum_{c \in \mathcal{C}^v} N_c(t)$. Realizations of these are denoted by lower case variables n_c and n^v, respectively.

This model captures the services/applications where tasks speed up with more allocated resources, e.g., a file download is faster when allocated more communication resources, or computation task that can be parallelized, e.g., typical MapReduce jobs [53], and mobile data analytics when additional compute resources are available [45]. For more complex applications involving different types of stages, the stages conducting massive data processing might be parallelizable, making it possible to accelerate by allocating more resources. For example, in mobile cloud gaming [54], the most time-consuming and resource-consuming stage is usually the cloud rendering where computing cluster renders the frames of the game. The rendering procedure can be accelerated by allocating more GPUs, and thus, can be viewed as a quantized version of our model.

Example: Let us consider an example where there are two autonomous vehicle service operators, say Slice 1 and Slice 2, coexisting in the same area, and supported by two edge computing nodes equipped with fronthaul connectivity and computational resources (e.g., edge GPUs), as shown in Fig. 3.1. Both nodes are connected to the same backhaul node. Different resources at

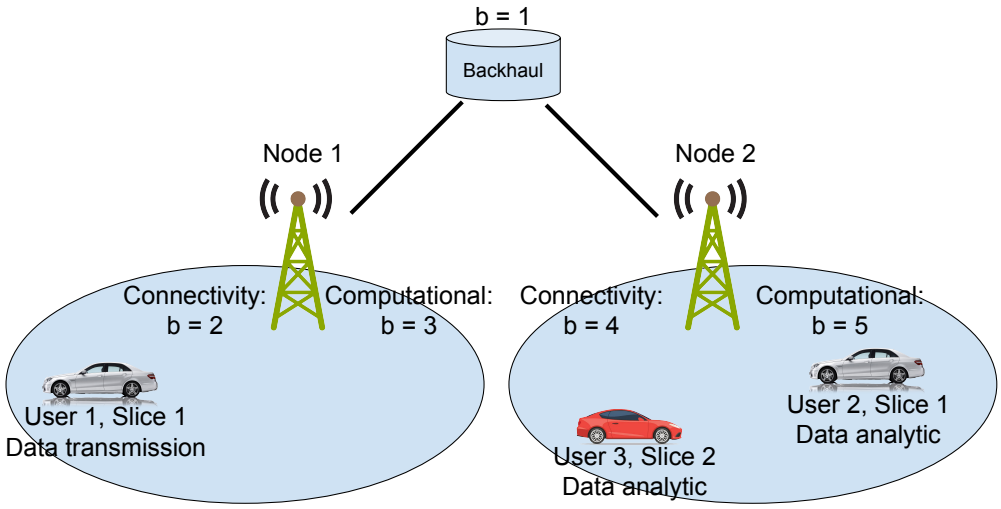


Figure 3.1: Example: network slicing in edge computing with autonomous cars.

different locations are indexed as in the figure. There are 3 vehicles (users) in this area, each of which corresponds to a user class. Users 1 and 2 are on Slice 1, and User 3 is on Slice 2, respectively. Each autonomous vehicle can run either of two applications. User 1 is conducting simple data transmission, with the resource demand vector $\mathbf{d}_1 = (1, 1, 0, 0, 0)$, meaning that User 1's application involves only connectivity resources, and to achieve a unit transmission rate for User 1, the system needs to allocate all the connectivity resources at both Node 1 and the backhaul. Meanwhile, User 2 and 3 are performing mobile data analytics, with demand vectors $\mathbf{d}_2 = \mathbf{d}_3 = (0.6, 0, 0, 1, 1)$, meaning that to achieve a unit processing rate for Users 2 or 3, the system needs to allocate 60% of the backhaul resource, all the fronthaul resource, together with all the computational resource at Node 2. Then, for example, if the resource alloca-

User \ Resource	1	2	3	4	5	Rate
user 1	0.4	0.4	0	0	0	$\lambda_1 = 0.4$
user 2	0.3	0	0	0.5	0.5	$\lambda_2 = 0.5$
user 3	0.3	0	0	0.5	0.5	$\lambda_3 = 0.5$

Table 3.1: Example resource allocation

tion is as given in Table 3.1, the system can achieve user service/processing rates given by $\lambda_1 = 0.4, \lambda_2 = \lambda_3 = 0.5$.

Next, we introduce the concept of *network share*. We assign each slice v a positive share s_v representing the fraction of overall resources to be committed to slice v. The share allocations across slices are denoted by $\mathbf{s} := (s_v : v \in \mathcal{V})$. Without loss of generality we assume $\sum_{v\in\mathcal{V}} s_v = 1$. In turn, each slice distributes its share s_v across its users $u \in \mathcal{U}^v$ according to a *Share-constrained weight allocation* scheme, defined as follows.

Definition 3.2.1. Share-constrained weight allocation (SCWA): A weight allocation across users $\mathbf{w} := (w_u : u \in \mathcal{U})$ is a share-constrained weight allocation if for each slice v,

$$\sum_{u\in\mathcal{U}^v} w_u = s_v. \tag{3.1}$$

If we consider the weight of each class c as $q_c := \sum_{u\in\mathcal{U}_c} w_u$, Eq. (3.1) implies $\sum_{c\in\mathcal{C}^v} q_c = s_v$. As a result, a slice can increase its users' weight by purchasing more shares. Also, note that if the number of users on a slice surges without increasing the associated share, on average each of its users should be given less weight. Two examples of SCWA are

1. **equal intra-class weight allocation**, where user weights are the same within a user class, i.e., $w_u = \frac{q_c}{n_c}$, for $u \in \mathcal{U}_c$ with $\sum_{c \in \mathcal{C}^v} q_c = s_v$; and

2. **equal intra-slice weight allocation**, where user weights are the same within a slice, i.e., $w_u = \frac{s_v}{n^v}$, for $u \in \mathcal{U}^v$. As a result, $q_c = \frac{s_{v(c)} n_c}{n^{v(c)}}$. One can see that equal intra-slice allocation is a further special case of equal intra-class allocation. When each user only demands one resource, such allocation emerges naturally as the social optimal, market and Nash equilibrium when slices exhibit (price taking) strategic behavior in optimizing their own utility, see [18].

In turn, the resources are ultimately committed to users, so a user-level resource allocation criterion is necessary. Let us denote the user rate allocation by $\boldsymbol{\lambda} := (\lambda_u : u \in \mathcal{U})$. In this paper, we assume equal intra-class weight allocation is used, resulting in equal rate allocation within a user class. Thus a class-level allocation criterion can be easily converted to a user-level one. For simplicity, the aggregated rate allocation across user classes is then denoted by $\boldsymbol{\phi} = (\phi_c : c \in \mathcal{C})$, where $\phi_c := n_c \lambda_u, u \in \mathcal{U}_c$, and the weight allocation across user classes by $\mathbf{q} = (q_c : c \in \mathcal{C})$. For each slice v, the weight allocation (across user classes) is $\mathbf{q}^v := (q_c : c \in \mathcal{C}^v)$, and the rate allocation is $\boldsymbol{\phi}^v := (\phi_c : c \in \mathcal{C}^v)$. In view of Eq. (3.1), we define the normalized weight allocation for slice v as $\tilde{\mathbf{q}}^v := (\tilde{q}_c := \frac{q_c}{s_v} : c \in \mathcal{C}^v)$. The rate allocation across slices is $\boldsymbol{\gamma} := (\gamma_v := \sum_{c \in \mathcal{C}^v} \phi_c : v \in \mathcal{V})$. The overall rate across the system is $\lambda := \|\boldsymbol{\lambda}\|_1 = \|\boldsymbol{\phi}\|_1 = \|\boldsymbol{\gamma}\|_1$, where $\|\cdot\|_1$ is the L1-norm. SCS is thus defined as follows.

Definition 3.2.2. $\alpha-$**Share Constrained Slicing (**$\alpha-$**SCS):** Under equal intra-class weight allocation with class weights $\mathbf{q}$, a class-level rate allocation $\boldsymbol{\phi}$ corresponds to $\alpha-$SCS if it is the solution to the following problem

$$\max_{\boldsymbol{\phi}} \ \{U_\alpha(\boldsymbol{\phi};\mathbf{q}) : \sum_{c\in\mathcal{C}_r} d_c^r \phi_c \leq 1, \forall r \in \mathcal{R}\}, \tag{3.2}$$

where $\alpha > 0$ is a pre-defined parameter and

$$U_\alpha(\boldsymbol{\phi};\mathbf{q}) := \begin{cases} e^{\sum_{v\in\mathcal{V}} U_\alpha^v(\boldsymbol{\phi}^v;\mathbf{q}^v)} & \alpha = 1, \\ \sum_{v\in\mathcal{V}} U_\alpha^v(\boldsymbol{\phi}^v;\mathbf{q}^v) & \alpha > 0 \ \text{ and } \ \alpha \neq 1, \end{cases}$$

where $U_\alpha^v(\boldsymbol{\phi}^v;\mathbf{q}^v)$ represents the utility function of slice v and is given by

$$U_\alpha^v(\boldsymbol{\phi}^v;\mathbf{q}^v) := \begin{cases} \sum_{c\in\mathcal{C}^v} q_c \log\left(\frac{\phi_c}{q_c}\right) & \alpha = 1, \\ \sum_{c\in\mathcal{C}^v} q_c \frac{(\phi_c/q_c)^{1-\alpha}}{1-\alpha} & \alpha > 0 \ \text{ and } \ \alpha \neq 1. \end{cases}$$

The criterion underlying SCS is different from class-level (weighted) $\alpha-$fairness proposed in [55] and [20], which is defined as follows.

Definition 3.2.3. Class-level $\alpha-$**fairness:** Under equal intra-class weight allocation, given $\mathbf{q}$, a class-level rate allocation $\boldsymbol{\phi}$ corresponds to (weighted) $\alpha-$fairness if it is the solution to Problem (3.2) with utility function of slice v given by

$$U_\alpha^v(\boldsymbol{\phi}^v;\mathbf{q}^v) := \begin{cases} \sum_{c\in\mathcal{C}^v} q_c \log\left(\phi_c\right) & \alpha = 1, \\ \sum_{c\in\mathcal{C}^v} q_c \frac{(\phi_c)^{1-\alpha}}{1-\alpha} & \alpha > 0 \ \text{ and } \ \alpha \neq 1. \end{cases}$$

As shown in [55], $\alpha-$fairness is equivalent to (weighted) proportional fairness as $\alpha = 1$ and unweighted maxmin fairness as $\alpha \to \infty$, while the asymptotic characterization of $\alpha-$SCS is given as follows.

Corollary 3.2.1. *$\alpha-$SCS is equivalent to (weighted) proportional fairness as $\alpha = 1$, and* weighted *max-min fairness as $\alpha \to \infty$.*

Here under equal intra-class weight allocation, weighted proportional fairness is defined as the solution to the following problem:

$$\max_{\boldsymbol{\phi}} \left\{ \sum_{c \in \mathcal{C}} q_c \log \phi_c : \sum_{c \in \mathcal{C}_r} d_c^r \phi_c \leq 1, \forall r \in \mathcal{R} \right\}, \tag{3.3}$$

and weighted max-min fairness is defined as the solution to the following problem:

$$\max_{\boldsymbol{\phi}} \left\{ \min_{c \in \mathcal{C}} \frac{\phi_c}{q_c} : \sum_{c \in \mathcal{C}_r} d_c^r \phi_c \leq 1, \forall r \in \mathcal{R} \right\}. \tag{3.4}$$

The persistence of weight is important, especially when α increases. Otherwise, the notion of share does not matter when α is large, undermining inter-slice protection. To the best of our knowledge, SCS is the first variation of $\alpha-$fairness incorporating user weighting in a consistent manner.

Proof. When $\alpha = 1$, one can see that the maximum is assumed when $\sum_{c \in \mathcal{C}} q_c \log \left(\frac{\phi_c}{q_c} \right)$ assumes maximum. Due to the concavity, a rate allocation $\boldsymbol{\phi}^* := (\phi_c^* : c \in \mathcal{C})$ is the maximizer if and only if

$$\sum_{c \in \mathcal{C}} \frac{q_c}{\phi_c^*} (\phi_c' - \phi_c^*) \leq 0,$$

for any feasible $\boldsymbol{\phi}'$. Also, for $\alpha-$SCS with weight $\mathbf{q}$, when $\alpha \neq 1$, $\boldsymbol{\phi}^*$ is the maximizer if and only if

$$\sum_{c \in \mathcal{C}} \left(\frac{\phi_c^*}{q_c} \right)^{-\alpha} (\phi_c' - \phi_c^*) \leq 0,$$

for any feasible $\boldsymbol{\phi}'$. One can see that two optimality conditions coincide when $\alpha = 1$.

The asymptotic behavior when $\alpha \to \infty$ is a direct corollary of the Lemma 3 in [55]. □

Let us define function $f_\alpha(\mathbf{x}; \mathbf{y})$ of two positive vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}_+^n$ such that $\|\mathbf{x}\|_1 = \|\mathbf{y}\|_1 = 1$ as

$$f_\alpha(\mathbf{x}; \mathbf{y}) = \begin{cases} e^{-D_{KL}(\mathbf{y}\|\mathbf{x})} & \alpha = 1, \\ \left(\sum_i y_i \left(\frac{x_i}{y_i} \right)^{1-\alpha} \right)^{\frac{1}{\alpha}} & \alpha > 0, \alpha \neq 1, \end{cases} \tag{3.5}$$

where $D_{KL}(\cdot\|\cdot)$ represents the Kulback-Leibler (K-L) divergence. The function $f_\alpha(\mathbf{x}; \mathbf{y})$ can be viewed as a measure of how close a normalized resource allocation $\mathbf{x}$ is to a normalized weight vector $\mathbf{y}$ in that, for example, when $\alpha = 1$, it decreases with the K-L divergence between $\mathbf{x}$ and $\mathbf{y}$, thus assumes maximum when $\mathbf{x} = \mathbf{y}$, meaning that the rate allocation is aligned with the specified weights. Thus $f_\alpha(\mathbf{x}; \mathbf{y})$ can be interpreted as a measure of $\mathbf{y}-$weighted fairness of allocation $\mathbf{x}$.

One can easily show that $f_\alpha(\mathbf{x}; \mathbf{y})$ is continuous. Moreover, for general α and a given $\mathbf{y}$, the following claim can be shown by setting the partial derivative of the associated Lagrangian to 0.

Proposition 3.2.2. *$f_\alpha(\mathbf{x}; \mathbf{y})$ assumes maximum when the rate is aligned with the weight when no constraint is imposed, i.e.,*

$$f_\alpha(\mathbf{y}; \mathbf{y}) = \max_{\mathbf{x}} f_\alpha(\mathbf{x}; \mathbf{y}) \tag{3.6}$$

One can show that for a given α, $\alpha-$SCS criterion can be factorized as follows.

Proposition 3.2.3. *For the $\alpha-$SCS criterion,*

$$U_\alpha(\boldsymbol{\phi};\mathbf{q}) = E_\alpha(\lambda)\left(F_\alpha^{inter}(\boldsymbol{\gamma})F_\alpha^{intra}(\boldsymbol{\phi};\mathbf{q})\right)^\alpha, \tag{3.7}$$

where $E_\alpha(\lambda)$, $F_\alpha^{inter}(\boldsymbol{\gamma})$ and $F_\alpha^{intra}(\boldsymbol{\phi};\mathbf{q})$ can be interpreted as the overall network efficiency, inter-slice and intra-slice fairness, respectively.

In Eq. (3.7), the efficiency is captured by a concave non-decreasing function of λ given by

$$E_\alpha(\lambda) := \begin{cases} \lambda & \alpha = 1, \\ \frac{\lambda^{1-\alpha}}{1-\alpha} & \alpha > 0 \text{ and } \alpha \neq 1. \end{cases}$$

The inter-slice fairness function is given by

$$F_\alpha^{\text{inter}}(\boldsymbol{\gamma}) := f_\alpha(\tilde{\boldsymbol{\gamma}};\mathbf{s}),$$

where $\tilde{\boldsymbol{\gamma}} := (\tilde{\gamma}^v := \gamma^v/\lambda : v \in \mathcal{V})$ is the normalized aggregated rate across slices. Let us define the normalized rate allocation across user classes on slice v as $\tilde{\boldsymbol{\phi}}^v := (\tilde{\phi}_c := \frac{\phi_c}{\gamma^v} : c \in \mathcal{C}^v)$. The intra-slice fairness term is then given by

$$F_\alpha^{\text{intra}}(\boldsymbol{\phi};\mathbf{q}) := \begin{cases} e^{\sum_{v\in\mathcal{V}} t_\alpha^v(\tilde{\boldsymbol{\gamma}};\mathbf{s}) \log f_\alpha(\tilde{\boldsymbol{\phi}}^v;\tilde{\mathbf{q}}^v)} & \alpha = 1, \\ \left(\sum_{v\in\mathcal{V}} t_\alpha^v(\tilde{\boldsymbol{\gamma}};\mathbf{s})(f_\alpha(\tilde{\boldsymbol{\phi}}^v;\tilde{\mathbf{q}}^v))^\alpha\right)^{\frac{1}{\alpha}} & \alpha \neq 1, \end{cases}$$

where $t_\alpha^v(\tilde{\boldsymbol{\gamma}};\mathbf{s})$ can be viewed as the weight for the fairness of each slice v:

$$t_\alpha^v(\tilde{\boldsymbol{\gamma}};\mathbf{s}) := \frac{s_v(\frac{\tilde{\gamma}^v}{s_v})^{1-\alpha}}{\sum_{v'\in\mathcal{V}} s_{v'}(\frac{\tilde{\gamma}^{v'}}{s_{v'}})^{1-\alpha}}. \tag{3.8}$$

One can see that Eq. (3.7) captures a trade-off among overall network efficiency, inter-slice fairness, which can be seen as a proxy of inter-slice protection, and intra-slice fairness. The significance of fairness increases as α increases. When $\alpha \to 0$, $\alpha-$SCS is maximizing the overall rate allocated, regardless of the weights. In order to achieve desirable resource utilization, a sharing criterion should realize *load-driven elasticity*, i.e., the amount of resources provisioned to a user class increases in the number of its users. Under equal intra-slice weight allocation, from Eq. (3.7) one can observe that, due to the fairness terms, the relative resource allocation of a slice tends to be aligned with $\tilde{\mathbf{q}}^v = (\frac{n_c}{n^v} : c \in \mathcal{C}^v)$, i.e., its relative load distribution. Thus the elasticity of $\alpha-$SCS is achieved as a result of weighted fairness. Specifically under SCS and parallel resource assumption, i.e., each user only uses one resource, $|\mathcal{R}_c| = 1, \forall c \in \mathcal{C}$, one can show the following result.

Theorem 3.2.4. *Under equal intra-slice weight allocation, assuming* $|\mathcal{R}_c| = 1, \forall c \in \mathcal{C}$, $\alpha-$*SCS is such that* ϕ_c *is a monotonically increasing function of* n_c, *when* $n_{c'}$ *is fixed for* $c' \neq c$.

Specifically in the setting of Theorem 3.2.4, each resource r will provision its resource across user classes in proportion to $\frac{s_{v(c)} n_c}{n^{v(c)}}$.

Such elasticity is key to achieving a sharing scheme that is aware of the inter-slice protection, while still improves the resource utilization by accommodating dynamic user loads on different slices.

3.3 Properties of SCS: A Utility-Based Perspective

3.3.1 System Model

In this section we will take a closer look at the characterization of SCS slice level rate allocations.

The SCS criterion (Problem (3.2)) is equivalent to the solution to the following problem

$$\max_{\boldsymbol{\phi}} \left\{ \sum_{v \in \mathcal{V}} U_\alpha^v(\boldsymbol{\phi}^v; \mathbf{q}^v) : \sum_{c \in \mathcal{C}_r} d_c^r \phi_c \leq 1, \quad \forall r \in \mathcal{R} \right\}. \tag{3.9}$$

We shall explore two key desirable properties for a sharing criterion, namely, *protection* and *envyfreeness*. In our setting, protection means that no slice is penalized under SCS sharing vs. static partitioning. Envyfreeness means that no slice is motivated to swap its resource allocation with another slice with a smaller share. These two properties together motivate the choice of $\alpha-$SCS sharing, and at least partially purchasing a larger share in order to improve performance.

3.3.2 Protection

Formally, let us characterize protection among slices by how much performance deterioration is possible for a slice when switching from static partitioning to $\alpha-$SCS sharing. Note that under static partitioning, slices are decoupled, so inter-slice protection is achieved possibly at the cost of efficiency. To be specific, the rate allocation for slice v under static partitioning

is given by the following problem.

$$\max_{\boldsymbol{\phi}^v} \left\{ U_\alpha^v(\boldsymbol{\phi}^v; \mathbf{q}^v) : \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c \leq s_v, \quad \forall r \in \mathcal{R} \right\}, \tag{3.10}$$

From now on, for a given α, let us denote the rate allocation for slice v under $\alpha-$SCS by $\boldsymbol{\phi}^{v,S} := (\phi_c^S : c \in \mathcal{C}^v)$, and that under static partitioning by $\boldsymbol{\phi}^{v,P} := (\phi_c^P : c \in \mathcal{C}^v)$. The parameter α is suppressed when there is no ambiguity. The following result demonstrates that $\alpha-$SCS with $\alpha = 1$ achieves inter-slice protection in that any slice achieves a better utility under $\alpha-$SCS sharing.

Theorem 3.3.1. *For a given* $\mathbf{q}$*, when the resource allocation is performed according to* $\alpha-$*SCS, difference in slice* v*'s utility compared to that under static partitioning is upper-bounded by (when* $\alpha \neq 1$*)*

$$U_v^\alpha(\boldsymbol{\phi}^{v,P}; \mathbf{q}^v) - U_v^\alpha(\boldsymbol{\phi}^{v,S}; \mathbf{q}^v) \leq s_v \left(\sum_{c \in \mathcal{C}^v} \tilde{q}_c \Big(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^* \Big)^{\frac{\alpha-1}{\alpha}} - \sum_{c \in \mathcal{C}} q_c \Big(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^* \Big)^{\frac{\alpha-1}{\alpha}} \right),$$

where $\tilde{q}_c := q_c / s_{v(c)}$ *is the normalized weight of class* c*, and* ν_r^* *is the optimal dual variable associated with the capacity constraint at resource* r *in Problem (3.9), also known as the shadow price of resource* r*.*

Remark: The right hand side characterizes how the protection changes with α. $\sum_{c \in \mathcal{C}^v} \tilde{q}_c (\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^*)^{1-\frac{1}{\alpha}}$ can be viewed as the average of the $(1 - \frac{1}{\alpha})-$order moment of 'charged' resource usage of slice v's user, while $\sum_{c \in \mathcal{C}} q_c (\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^*)^{1-\frac{1}{\alpha}}$ is that of the overall users. When $0 < \alpha < 1$, sharing

tends to benefit slices with greater average user usages, at the cost of other slices, while when $\alpha > 1$, slices with smaller average user prices are preferred.

When $\alpha \to 1$, the utility of slice v, $U_v^\alpha(\boldsymbol{\phi}^v; \mathbf{q}^v) = \frac{1}{1-\alpha} \times \sum_{c \in \mathcal{C}^v} q_c \left(\frac{\phi_c}{q_c}\right)^{1-\alpha}$ tends to be non-changing with $\boldsymbol{\phi}$, so the result in Theorem 3.3.1 seems to be trivial. However, due to the factor $\frac{1}{1-\alpha}$, the utility function is not well-defined at $\alpha \to 1$. Thus the exact result for $\alpha = 1$ need to be discussed on its own, as in Theorem 3.3.2.

Also, note that when SCWA constraint Eq. (3.1) is voided, another form of the theorem can be written as

$$U_v^\alpha(\boldsymbol{\phi}^{v,P}; \mathbf{q}^v) - U_v^\alpha(\boldsymbol{\phi}^{v,S}; \mathbf{q}^v) \leq w_v(p_v - s_v \frac{w_0}{w_v} \bar{p}_0),$$

where $w_v := \sum_{c \in \mathcal{C}^v} q_c$ is the total weight of users of slice v, $w_0 := \sum_{c \in \mathcal{C}} q_c$ is the total weight of all users, $p_v := \sum_{c \in \mathcal{C}^v} \tilde{q}_c \left(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^*\right)^{1-\frac{1}{\alpha}}$ is the average $1 - \frac{1}{\alpha}$ power of the prices of slice v's users, and $\bar{p}_0 := \sum_v \frac{w_v}{w_0} p_v$ is the weighted average of p_v across all slices. One could see that, if w_v is not limited, as $w_v \to \infty$, as long as $p_v \neq \bar{p}_0$, the gap can be arbitrarily bad, implying significant utility loss when slice-level sharing is used. In comparison, if we use SCWA, i.e., constrained by Eq. (3.1), the right hand side equals to $s_v(p_v - \bar{p}_0)$. This quantity is small when s_v is small, or slice v's users do not use many 'expensive' resources.

Proof. Under sharing scheme, the rate allocation for each slice v should be the

same as the solution to the following problem:

$$\begin{aligned} \max_{\boldsymbol{\phi}^v} \quad & U_\alpha^v(\boldsymbol{\phi}^v; \mathbf{q}^v) \\ \text{such that} \quad & \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c \leq \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c^S, \quad \forall r \in \mathcal{R}. \end{aligned} \tag{3.11}$$

Problem (3.11) yields solution $\boldsymbol{\phi}^{v,\mathrm{S}}$. In comparison, under static partitioning scheme, the rate allocation for slice v is given by

$$\begin{aligned} \max_{\boldsymbol{\phi}^v} \quad & U_\alpha^v(\boldsymbol{\phi}^v; \mathbf{q}^v) \\ \text{such that} \quad & \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c \leq s_v, \quad \forall r \in \mathcal{R}, \end{aligned} \tag{3.12}$$

which can be regarded as a perturbed version of Problem (3.11), and yields solution $\boldsymbol{\phi}^{v,\mathrm{P}}$. It is a well known result in convex optimization [56] that the change in the optimal objective function value due to perturbation of the constraints can be bounded by:

$$U_v^\alpha(\boldsymbol{\phi}^{v,\mathrm{P}}; \mathbf{q}^v) - U_v^\alpha(\boldsymbol{\phi}^{v,\mathrm{S}}; \mathbf{q}^v) \leq \sum_{r \in \mathcal{R}} \nu_r^* \left(\sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c^S - s_v \right). \tag{3.13}$$

The Lagrangian of Problem (3.11) is

$$L^\alpha(\boldsymbol{\phi}^v; \boldsymbol{\nu}) = -\frac{1}{1-\alpha} \sum_{c \in \mathcal{C}^v} q_c \left(\frac{\phi_c}{q_c} \right)^{1-\alpha} + \sum_{r \in \mathcal{R}} \nu_r \Big(\sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c - \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c^S \Big). \tag{3.14}$$

Setting the partial derivative against $\boldsymbol{\phi}^v$ to 0, we obtain the dual function as:

$$\begin{aligned} g^\alpha(\boldsymbol{\nu}) = & -\frac{1}{1-\alpha} \sum_{c \in \mathcal{C}^v} q_c \left(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r \right)^{1-\frac{1}{\alpha}} \\ & + \sum_{r \in \mathcal{R}} \nu_r \left(\sum_{c \in \mathcal{C}_r^v} d_c^r q_c \left(\sum_{r' \in \mathcal{R}_c} d_c^{r'} \nu_{r'} \right)^{-\frac{1}{\alpha}} - \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c^S \right). \end{aligned}$$

Also,

$$\phi_c^S = q_c \left(\sum_{r' \in \mathcal{R}_c} d_c^{r'} \nu_{r'}^* \right)^{-\frac{1}{\alpha}}. \tag{3.15}$$

By swapping the order of summation, one can show that

$$\sum_{r \in \mathcal{R}} \nu_r \sum_{c \in \mathcal{C}_r^v} d_c^r q_c \left(\sum_{r' \in \mathcal{R}_c} d_c^{r'} \nu_{r'} \right)^{-\frac{1}{\alpha}} = \sum_{c \in \mathcal{C}^v} q_c \left(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r \right)^{1-\frac{1}{\alpha}}.$$

Due to strong convexity of Problem (3.11), we know

$$g^\alpha(\boldsymbol{\nu}^*) = -\frac{1}{1-\alpha} \sum_{c \in \mathcal{C}^v} q_c \left(\frac{\phi_c^S}{q_c} \right)^{1-\alpha}.$$

Plugging in Eq. (3.15) we have

$$\sum_{c \in \mathcal{C}^v} q_c \left(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^* \right)^{1-\frac{1}{\alpha}} - \sum_{r \in \mathcal{R}} \nu_r^* \sum_{c \in \mathcal{C}_r^v} d_c^r \phi_c^S = 0. \tag{3.16}$$

Note that if a resource is binding, the sum of resource allocated should equal to 1. Otherwise it has 0 shadow price. Summing above across $v \in \mathcal{V}$, we have

$$\sum_{c \in \mathcal{C}} q_c \left(\sum_{r \in \mathcal{R}_c} d_c^r \nu_r^* \right)^{1-\frac{1}{\alpha}} - \sum_{r \in \mathcal{R}} \nu_r^* = 0. \tag{3.17}$$

Plugging in the right hand side of Eq. (3.13) to substitute $\sum_{r \in \mathcal{R}} \nu_r^*$, and also plugging in Eq. (3.15), the theorem is proved. □

Following is the result specifically for the case when $\alpha = 1$.

Theorem 3.3.2. *For a given* $\mathbf{q}$*, when the resource allocation is performed according to* $1-$*SCS, slice* v*'s utility exceeds that under static partitioning (Problem (3.10)), i.e.,*

$$U_1^v(\boldsymbol{\phi}^{v,P}; \mathbf{q}^v) \leq U_1^v(\boldsymbol{\phi}^{v,S}; \mathbf{q}^v). \tag{3.18}$$

Remark: It is a straightforward observation that under $\alpha-$SCS, the global utility $\sum_{v\in\mathcal{V}} U_1^v(\boldsymbol{\phi}^v;\mathbf{q}^v)$ is improved since it can be viewed as relaxing the system constraints. However, Theorem 3.3.2 asserts that this holds uniformly on a per slice basis.

Proof. Similar to the argument for general α, we have that the gap between sharing and static partitioning satisfies Eq. (3.13). Also, by solving the first order condition, one can obtain that $\phi_c^S = \frac{q_c}{\sum_{r\in\mathcal{R}_c} d_c^r \nu_r^*}$. By plugging in this expression and swapping the order of summation we have

$$U_1^v(\boldsymbol{\phi}^{v,P};\mathbf{q}^v) - U_1^v(\boldsymbol{\phi}^{v,S};\mathbf{q}^v) \leq s_v(1-\sum_{r\in\mathcal{R}}\nu_r^*), \tag{3.19}$$

where ν_r^* is the shadow price of resource r under SCS, or the dual variables associated with the capacity constraints.

Then if we have $\sum_{r\in\mathcal{R}}\nu_r^* = 1$, the proof is complete. For $\alpha = 1$, the Lagrangian is given by

$$L^1(\boldsymbol{\phi}^v;\boldsymbol{\nu}) = -\sum_{c\in\mathcal{C}^v} q_c \log\phi_c + \sum_{r\in\mathcal{R}}\nu_r(\sum_{c\in\mathcal{C}_r^v} d_c^r\phi_c - \sum_{c\in\mathcal{C}_r^v} d_c^r\phi_c^S).$$

By setting the derivative against $\boldsymbol{\phi}^v$ to 0, we have the dual function as

$$g^1(\boldsymbol{\nu}) = -\sum_{c\in\mathcal{C}^v} q_c \log\frac{q_c}{\sum_{r\in\mathcal{R}_c} d_c^r\nu_r} + s_v - \sum_{r\in\mathcal{R}}\nu_r\sum_{c\in\mathcal{C}_r^v} d_c^r\phi_c^S,$$

and

$$\phi_c^S = \frac{q_c}{\sum_{r\in\mathcal{R}_c} d_c^r\nu_r^*}. \tag{3.20}$$

By strong duality, maximal dual should be minimal primal function. And optimal dual is achieved at the shadow price $\boldsymbol{\nu}^*$. Thus,

$$g^1(\boldsymbol{\nu}^*) = -\sum_{c\in\mathcal{C}^v} q_c \log \phi_c^S,$$

which gives us

$$s_v - \sum_{r\in\mathcal{R}} \nu_r^* \sum_{c\in\mathcal{C}_r^v} d_c^r \phi_c^S = 0. \tag{3.21}$$

Summing above across $v \in \mathcal{V}$ we have $1 - \sum_{r\in\mathcal{R}} \nu_r^* = 0$. Because if a resource is binding, the sum of rate allocated should be equal to 1. Otherwise it has 0 shadow price. Plugging above result into Eq. (3.19), the theorem is proved. □

3.3.3 Envyfreeness

Formally, envyfreeness is defined under the assumption that, for two slices v and v', if they swap their allocated resources, slice v's associated utility will not be improved if $s_{v'} \leq s_v$. Before swapping, the rate allocation for slice v is given by $\boldsymbol{\phi}^{v,S}$, while after swapping with slice v', its rate allocation is determined by solving following problem:

$$\max_{\boldsymbol{\phi}^v} \ \{U_\alpha^v(\boldsymbol{\phi}^v; \mathbf{q}^v) : \sum_{c\in\mathcal{C}_r^v} d_c^r \phi_c \leq \sum_{c\in\mathcal{C}_r^{v'}} d_c^r \phi_c^S, \ \forall r \in \mathcal{R}\}.$$

Note that $\sum_{c\in\mathcal{C}_r^{v'}} d_c^r \phi_c^S$ corresponds to the fraction of resource r provisioned to slice v'. Let us denote the solution to such problem for slice v as $\boldsymbol{\phi}^{v\leftrightarrow v'}$. Then we have the following result.

Theorem 3.3.3. *The difference between the utility obtained by slice v under $\alpha-$SCS with SCWA, and that under static partitioning within the resource*

provisioned to another slice v' is upper-bounded by the following inequality:

$$U_\alpha^v(\boldsymbol{\phi}^{v\leftrightarrow v'};\mathbf{q}^v) - U_\alpha^v(\boldsymbol{\phi}^{v,S};\mathbf{q}^v) \leq \sum_{c\in\mathcal{C}^{v'}} q_c \left(\sum_{r\in\mathcal{R}_c} d_c^r \nu_r^*\right)^{\frac{\alpha-1}{\alpha}} - \sum_{c\in\mathcal{C}^v} q_c \left(\sum_{r\in\mathcal{R}_c} d_c^r \nu_r^*\right)^{\frac{\alpha-1}{\alpha}}.$$

Remark: As a special case, when $\alpha = 1$, the right hand side of the inequality becomes $s_{v'} - s_v$, and thus a slice has no incentive to swap its allocation with another with a less or equal share, which implies SCS is envyfree. Envyfreeness implies that $\alpha-$SCS achieves desirable resource utilization in that the right portion of resource is provisioned to the right slice.

Proof. Still by the sensitivity of convex optimization problem [56], we have

$$U_\alpha^v(\boldsymbol{\phi}^{v\leftrightarrow v'};\mathbf{q}^v) - U_\alpha^v(\boldsymbol{\phi}^{v,S};\mathbf{q}^v) \leq \sum_{r\in\mathcal{R}} \nu_r^* \left(\sum_{c\in\mathcal{C}_r^{v'}} d_c^r \phi_c^S - \sum_{c\in\mathcal{C}_r^v} d_c^r \phi_c^S\right). \quad (3.22)$$

Then by substituting Eq. (3.16) the theorem is proved. □

3.3.4 Using $\infty-$SCS As a Surrogate for $1-$SCS

From previous discussions, one can see that it is of particular interest to use $1-$SCS as the fairness criterion, for it achieves strict protection and envyfreeness. When $\alpha = 1$, $\alpha-$SCS becomes weighted proportional fairness, whose solution usually involves iterative methods, and the complexity increases rapidly with the number of user classes as well as the accuracy requirement, see, e.g., [57], making it hard to implement in large-scale. In comparison, weighted max-min is relatively easy to implement in distributed

manner, see [3] for example. Specifically a progressive water-filling algorithm [57] has $O(|\mathcal{C}| \max_{c\in\mathcal{C}} |\mathcal{R}_c|)$ complexity. Thus, in our work we will discuss the feasibility of using $\infty-$SCS, which is equivalent to a (dynamically) weighted maxmin, as a surrogate to $1-$SCS. If the resulted utility function is not far from the optimum of $1-$SCS criterion, we shall assert $\infty-$SCS achieves similar performance as $1-$SCS.

For simplicity, we consider the original form of weighted-log utility, given by

$$\Psi(\boldsymbol{\phi}; \mathbf{q}) := \sum_{c\in\mathcal{C}} q_c \log \phi_c. \tag{3.23}$$

Then for the overall utility achieved, we have following theorem.

Theorem 3.3.4. *For a given weight allocation* $\mathbf{q}$*, if* $d_c^r \geq 1, \forall r \in \mathcal{R}, c \in \mathcal{C}$*, we have*

$$\Psi(\boldsymbol{\phi}^{*,1}; \mathbf{q}) - \Psi(\boldsymbol{\phi}^{*,\infty}; \mathbf{q}) \leq \sum_{c\in\mathcal{C}} q_c D_c - 1, \tag{3.24}$$

where $\boldsymbol{\phi}^{*,\alpha} := (\phi_c^{*,\alpha} : c \in \mathcal{C})$ *is the optimal rate allocation under* $\alpha-$*SCS, and* $D_c := \sum_{r\in\mathcal{R}_c} d_c^r$.

Remark: First note that the condition $d_c^r \geq 1$ can be easily satisfied by rescaling the unit of rate without loss of generality. Also by rescaling, one can show that such bound vanishes when each user class is associated with only one resource, i.e., $|\mathcal{R}_c| = 1, \forall c \in \mathcal{C}$, and d_c^r are the same, e.g., $d_c^r = 1$. Such bound implies that, the suboptimality due to using a surrogate solution to achieve weighted proportional fairness depends on the diversity in

the users' requirements on resources. Also, this gap of suboptimality cannot be arbitrarily bad because under SCWA, we have $\sum_c q_c = 1$, thus the right hand side of Eq. (3.24) is at most $\max_c D_c - 1$.

Proof. Note that when $\alpha \to \infty$, SCS approaches weighted maxmin, which can be solved by a progressive water-filling algorithm. Let us denote the resource where class c is bottlenecked under weighted maxmin by $r(c)$, and in turn, the set of users being bottlenecked at resource r by $\tilde{\mathcal{C}}_r$. Let us define ν_r^* as the shadow price for resource r when $\alpha = 1$. According to the definition we have

$$\begin{aligned}
\Psi(\boldsymbol{\phi}^{*,1}; \mathbf{q}) - \Psi(\boldsymbol{\phi}^{*,\infty}; \mathbf{q}) &= \sum_c q_c (\log \phi_c^{*,1} - \log \phi_c^{*,\infty}) \\
&\leq \sum_c q_c \left(\log \frac{q_c}{\sum_{r' \in \mathcal{R}_c} d_c^{r'} \nu_{r'}^*} - \log \left(\frac{q_c}{\sum_{c' \in \mathcal{C}_{r(c)}} d_{c'}^{r(c)} q_{c'}} \right) \right) \\
&= \sum_{r \in \mathcal{R}} \sum_{c \in \tilde{\mathcal{C}}_r} q_c \log \left(\frac{\sum_{c' \in \mathcal{C}_r} d_{c'}^r q_{c'}}{\sum_{r' \in \mathcal{R}_c} d_c^{r'} \nu_{r'}^*} \right).
\end{aligned}$$

The first inequality follows from the form of solution of sharing problem when $\alpha = 1$, and the fact that $\phi_c^{*,\infty} \geq \frac{q_c}{\sum_{c' \in \mathcal{C}_{r(c)}} d_{c'}^{r(c)} q_{c'}}$, since the worst rate user u could obtain is when there is no other users get saturated before it at its

bottleneck resource. Because $\log x \leq x - 1$ we have

$$\begin{aligned}
&\Psi(\boldsymbol{\phi}^{*,1}; \mathbf{q}) - \Psi(\boldsymbol{\lambda}^{*,\infty}; \mathbf{q}) \\
&\leq \sum_{r \in \mathcal{R}} \sum_{c \in \tilde{\mathcal{C}}_r} q_c \frac{\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'}}{\sum_{r' \in \mathcal{R}_c} d^{r'}_c \nu^*_{r'}} - 1 \\
&= \sum_{r \in \mathcal{R}} \left(\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'} \right) \sum_{c \in \tilde{\mathcal{C}}_r} \frac{q_c}{\sum_{r' \in \mathcal{R}_c} d^{r'}_c \nu^*_{r'}} - 1 \\
&\leq \sum_{r \in \mathcal{R}} \left(\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'} \right) \sum_{c \in \mathcal{C}_r} \frac{q_c}{\sum_{r' \in \mathcal{R}_c} d^{r'}_c \nu^*_{r'}} - 1 \\
&\leq \sum_{r \in \mathcal{R}} \left(\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'} \right) \sum_{c \in \mathcal{C}_r} \frac{d^r_c q_c}{\sum_{r' \in \mathcal{R}_c} d^{r'}_c \nu^*_{r'}} - 1 \\
&\leq \sum_{r \in \mathcal{R}} \left(\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'} \right) - 1.
\end{aligned}$$

The penultimate inequality holds true because $d^r_c \geq 1, \forall c \in \mathcal{C}_r$. The last inequality comes from the capacity constraint, by plugging in $\phi^{*,1}_c = \frac{q_c}{\sum_{r \in \mathcal{R}_c} d^r_c \nu^*_r}$ into $\sum_{c \in \mathcal{C}_r} d^r_c \phi^{*,1}_c \leq 1$, we have $\sum_{c \in \mathcal{C}_r} \frac{d^r_c q_c}{\sum_{r' \in \mathcal{R}_c} d^{r'}_c \nu^*_{r'}} \leq 1$. Then by swapping the order of summation, we have

$$\sum_{r \in \mathcal{R}} \left(\sum_{c' \in \mathcal{C}_r} d^r_{c'} q_{c'} \right) = \sum_{c \in \mathcal{C}} q_c \sum_{r \in \mathcal{R}_c} d^r_c = \sum_{c \in \mathcal{C}} q_c D_c.$$

□

3.4 Elastic Traffic Model

3.4.1 System Model

In this section we switch gears to study a scenario where the user traffic is elastic, i.e., each user carries a certain amount of work and leaves the system once it is finished. Specifically, for a class-c user, we assume that its service requirement is drawn from an exponential distribution with mean $\frac{1}{\mu_c}$ independently, and its arrival follows a Poisson process with intensity ν_c. Then the traffic intensity associated with user class c is given by $\rho_c = \frac{\nu_c}{\mu_c}$.

Let us first consider a given time instant, when the size of $\mathcal{U}_c$ and $\mathcal{U}^v$ are given by n_c and n^v respectively. Also, for simplicity we assume equal intra-slice weight allocation, thus $q_c = \frac{s_{v(c)} n_c}{n^{v(c)}}$. Substituting q_c into Problem (3.2), the $\alpha-$SCS criterion can be rewritten as follows.

$$\begin{aligned} \max_{\boldsymbol{\phi}} \quad & \sum_{c \in \mathcal{C}} \left(\frac{s_{v(c)} n_c}{n^{v(c)}} \right)^{\alpha} \frac{(\phi_c)^{1-\alpha}}{1-\alpha} \\ \text{such that} \quad & \sum_{c \in \mathcal{C}_r} \phi_c d_c^r \leq 1, \ \forall r \in \mathcal{R}. \end{aligned} \tag{3.25}$$

3.4.2 Stability

Problem (3.25) characterizes the rate allocation across classes when the numbers of users in the network are fixed. However, it is natural to study the evolution of the system when user distributions are random processes. Note that while [20] studied the stability condition for $\alpha-$fairness when weights are introduced, their weights do not depend on the dynamic distribution of users in the network. By using the fluid system theory established in [58],[59] and

[60], one can show that SCS stablizes the system as long as no resource is overloaded.

Theorem 3.4.1. *Assume that under equal intra-slice weight allocation, the rate allocation is given by Problem (3.25). Then, when the following effective load conditions are satisfied:*

$$\sum_{c \in \mathcal{C}_r} \rho_c d_c^r < 1, \ \forall r \in \mathcal{R}, \tag{3.26}$$

the network is stable.

Remark: Theorem 3.4.1 is significant in that the system might become transient under specific sharing criterion even when Eq. (3.26) is satisfied, e.g., Example 1 in [20] when strict priorities are designated in favor of the system throughput. Moreover, Example 2 in the same literature demonstrates that even no strict priority is designated, instability is still possible under Eq. (3.26). Those examples implies the importance of SCS sharing and associated weight allocation schemes.

The result in [20] is under the assumption that each user has a fixed weight. Thus the *overall* resources committed to a slice increases with the number of its active users, possibly compromising inter-slice protection. Theorem 3.4.1 shows that even when inter-slice protection is maintained, SCS can still stablize the system through efficient utilization.

Proof. This can be proved by studying the "fluid system" associated with the service discipline proposed. Briefly, the "fluid system" associated with

a queuing system is its asymptotic version when the transition frequency is very high and the change of the queue length in one transition is infinitesimal. Such limiting is approached by rescaling the time axis. The stability of the original queuing system can then be examined by studying the associated "fluid system", see, for example, [59], [60], and [58].

According to [59] and [60], if one can show that such fluid system gets empty eventually, the associated original queuing system is positive recurrent. In view of this result, the outline of the proof is as follows. Firstly we establish two functions $K(t)$ and $H(t)$ such that $K(t) \geq H(t) \geq 0$, where $H(t)$ only takes 0 value when all the fluid limits equal to 0. Then we find a lower bound on the negative drift rate of $K(t)$ so that we can conclude that $K(t) \to 0$ eventually. Therefore $H(t)$, together with all the fluid limits tend to 0 eventually.

Let us define the vector of users' distribution as $\mathbf{N}(t) = (N_c(t) : c \in \mathcal{C})$. Consider the set of "fluid limits" defined by:

$$\mathbf{x}(t) = \lim_{\omega \to \infty} \frac{\mathbf{N}(\omega t)}{\omega}, \text{ with } \sum_{c \in \mathcal{C}} N_c(0) = \omega, \tag{3.27}$$

where $\mathbf{x}(t) := (x_c(t) : c \in \mathcal{C})$ is the vector of the fluid limit for each class. If such limit exists, we have $\sum_{c \in \mathcal{C}} x_c(0) = 1$. According to the Lemma 4.2 in [59], from Strong Law of Large Number one can derive that, $\mathbf{x}(t)$ is deterministic and the dynamic of such fluid limits system is actually determined by the rate allocation problem associated with the fluid limits. That is, $\mathbf{x}(t)$ follows the differential equations:

$$\frac{d}{dt} x_c(t) = \nu_c - \mu_c \tilde{\phi}_c(t), \text{ when } x_c(t) > 0, \tag{3.28}$$

where $\tilde{\phi}_c(t)$ is the aggregated rate allocated to the fluid limit of class-c, which should be given by the following problem:

$$\begin{aligned} \max_{\tilde{\boldsymbol{\phi}}:=(\tilde{\phi}_c:c\in\mathcal{C})} \quad & \sum_{c\in\mathcal{C}} \left(\frac{s_{v(c)}x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)} \right)^{\alpha} \frac{\tilde{\phi}_c^{1-\alpha}(t)}{1-\alpha} \\ \text{such that} \quad & \sum_{c\in\mathcal{C}_r} \tilde{\phi}_c(t) d_c^r \leq 1, \ \forall r \in \mathcal{R}. \end{aligned} \tag{3.29}$$

Let us assume that $\tilde{\boldsymbol{\phi}}(t)$ achieves the maximum of Problem (3.29). Then the concavity of the objective function, together with the first-order optimality condition gives us

$$G'(\boldsymbol{\zeta}) \cdot (\boldsymbol{\zeta} - \boldsymbol{\Lambda}) \leq 0,$$

where $G(\cdot)$ is the objective function of Problem (3.29), for any feasible rate allocation vector $\boldsymbol{\zeta}$. Also note that, if the capacity constraints Eq. (3.26) are satisfied by $\boldsymbol{\rho}$, there exists $\epsilon > 0$ such that $(1+\epsilon)\boldsymbol{\rho}$ also satisfies Eq. (3.26). Plugging in $(1+\epsilon)\boldsymbol{\rho}$ as $\boldsymbol{\zeta}$ to the above inequality we have:

$$\sum_{c\in\mathcal{C}} \left(\frac{s_{v(c)}x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)} \right)^{\alpha} \rho_c^{-\alpha}(\rho_c - \tilde{\phi}_c(t)) \leq -\epsilon \sum_{c\in\mathcal{C}} \left(\frac{s_{v(c)}x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)} \right)^{\alpha} \rho_c^{1-\alpha}. \tag{3.30}$$

If we define function $K(t)$ as

$$\begin{aligned} K(t) := \ & \sum_{v\in\mathcal{V}} (s_v)^{\alpha} \sum_{c\in\mathcal{C}^v} \int_0^t \left(\frac{x_c(\tau)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(\tau)} \right)^{\alpha} \frac{(\rho_c - \tilde{\phi}_c(\tau))}{(\rho_c)^{\alpha}} d\tau \\ & + \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \|\mathbf{x}^v(0)\|_{\alpha+1}, \end{aligned} \tag{3.31}$$

where $\bar{\mu} = \max_c \mu_c$, and $\bar{\rho} = \max_c \rho_c$ are the maximal processing rate and effective load across user types, respectively, and we define the fluid limit

vector of slice v at time t as $\mathbf{x}^v(t) := (x_c(t) : c \in \mathcal{C}^v)$, with its L$k-$norm denoted by $\|\mathbf{x}^v(t)\|_k$. We have that Eq. (3.30) is equivalent to

$$\frac{d}{dt}K(t) \leq -\epsilon \sum_{c\in\mathcal{C}} \left(\frac{s_{v(c)}x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} \rho_c^{1-\alpha}. \tag{3.32}$$

The right hand side of the above inequality can be bounded by:

$$\begin{aligned}
&\sum_{c\in\mathcal{C}} \left(\frac{s_{v(c)}x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} \rho_c^{1-\alpha} \\
&\quad \geq s_{min}^{\alpha}\rho_{bound}^{1-\alpha} \sum_v \sum_{c\in\mathcal{C}^v} \left(\frac{x_c(t)}{\sum_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} \\
&\quad \geq s_{min}^{\alpha}\rho_{bound}^{1-\alpha} \min\{1, (\max_v |\mathcal{C}^v|)^{1-\alpha}\},
\end{aligned}$$

where $s_{min} = \min_v s_v$, ρ_{bound} takes $\bar{\rho}$ when $\alpha > 1$ and takes $\min_c \rho_c$ when $0 < \alpha < 1$. The inequality is due to that for each active slice (a slice is said to be active if $\sum_{c\in\mathcal{C}^v} x_c(t) > 0$), we have two possible cases:

1. When $0 < \alpha \leq 1$, we have

$$\sum_{c\in\mathcal{C}^v} \left(\frac{x_c(t)}{\sum\limits_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} \geq \left(\sum_{c\in\mathcal{C}^v} \frac{x_c(t)}{\sum\limits_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} = 1,$$

   due to the concavity of power-α.

2. When $\alpha > 1$, we have

$$\sum_{c\in\mathcal{C}^v} \left(\frac{x_c(t)}{\sum\limits_{c'\in\mathcal{C}^v} x_{c'}(t)}\right)^{\alpha} = \frac{\sum\limits_{c\in\mathcal{C}^v} x_c^{\alpha}(t)}{\left(\sum\limits_{c\in\mathcal{C}^v} x_c(t)\right)^{\alpha}} \geq |\mathcal{C}^v|^{1-\alpha}.$$

   The inequality is due to that $\|\mathbf{x}^v(t)\|_{\alpha}|\mathcal{C}^v|^{1-\frac{1}{\alpha}} \geq \|\mathbf{x}^v(t)\|_1$ when $\alpha > 1$, see [61].

Thus we found a lower bound for each v. By noting that there should be at least one active user type before the fluid system gets emptied, we can get the last factor by taking the minimum across all slices.

Thus, we have

$$\begin{aligned}\frac{d}{dt}K(t) &\leq -\epsilon s_{min}^{\alpha}\rho_{bound}^{1-\alpha}\min\{1,(\max_{v}|\mathcal{C}^{v}|)^{1-\alpha}\}\\ K(t) &\leq K(0)-\epsilon s_{min}^{\alpha}\rho_{bound}^{1-\alpha}\min\{1,(\max_{v}|\mathcal{C}^{v}|)^{1-\alpha}\}t. \qquad (3.33)\end{aligned}$$

In order to find a lower bound of $K(t)$, we observe that for each slice $v \in \mathcal{V}$ we have

$$\begin{aligned}&\sum_{c\in\mathcal{C}_v}\int_0^t\left(\frac{x_c(\tau)}{\sum_{c'\in\mathcal{C}^v}x_{c'}(\tau)}\right)^{\alpha}\rho_c^{-\alpha}(\rho_c-\tilde{\phi}_c(\tau))d\tau\\ \geq\ &\frac{1}{\bar{\mu}\bar{\rho}^{\alpha}}\sum_{c\in\mathcal{C}^v}\int_0^t\left(\frac{x_c(\tau)}{\sum_{c'\in\mathcal{C}^v}x_{c'}(\tau)}\right)^{\alpha}dx_c(\tau),\end{aligned}$$

and

$$
\begin{aligned}
& \sum_{c\in\mathcal{C}^v}\int_0^t \left(\frac{x_c(\tau)}{\sum_{c\in\mathcal{C}^v} x_c(\tau)}\right)^{\alpha} dx_c(\tau) \\
& \underline{\underline{y_c(t):=(x_c(t))^{\alpha+1}}} \frac{1}{\alpha+1}\sum_{c\in\mathcal{C}^v}\int_{y_c(0)}^{y_c(t)} \left(\frac{1}{\sum\limits_{c'\in\mathcal{C}^v}(y_{c'}(\tau))^{\frac{1}{\alpha+1}}}\right)^{\alpha} dy_c(\tau) \\
= \quad & \frac{1}{\alpha+1}\sum_{c\in\mathcal{C}^v}\int_{y_c(0)}^{y_c(t)} \left(\left(\sum_{c'\in\mathcal{C}^v}(y_{c'}(\tau))^{\frac{1}{\alpha+1}}\right)^{\alpha+1}\right)^{-\frac{\alpha}{\alpha+1}} dy_c(\tau) \\
= \quad & \frac{1}{\alpha+1}\sum_{c\in\mathcal{C}^v}\int_{y_c(0)}^{y_c(t)} \left(\|\mathbf{y}^v(\tau)\|_{\frac{1}{\alpha+1}}\right)^{-\frac{\alpha}{\alpha+1}} dy_c(\tau) \\
\geq \quad & \frac{1}{\alpha+1}\sum_{c\in\mathcal{C}^v}\int_{y_c(0)}^{y_c(t)} \left(|\mathcal{C}^v|^{\alpha}\|\mathbf{y}^v(\tau)\|_1\right)^{-\frac{\alpha}{\alpha+1}} dy_c(\tau) \\
= \quad & \frac{|\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}}}{\alpha+1}\int_{\|\mathbf{y}^v(0)\|_1}^{\|\mathbf{y}^v(t)\|_1} \left(\|\mathbf{y}^v(\tau)\|_1\right)^{-\frac{\alpha}{\alpha+1}} d\left(\|\mathbf{y}^v(\tau)\|_1\right) \\
= \quad & |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}}\left(\left(\|\mathbf{y}^v(t)\|_1\right)^{\frac{1}{\alpha+1}} - \left(\|\mathbf{y}^v(0)\|_1\right)^{\frac{1}{\alpha+1}}\right) \\
= \quad & |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}}\left(\|\mathbf{x}^v(t)\|_{\alpha+1} - \|\mathbf{x}^v(0)\|_{\alpha+1}\right),
\end{aligned}
$$

where the inequality comes from the relation between $L1-$norm and $L(\frac{1}{\alpha+1})-$norm, and the following equality is by moving the summation into the integral. Plug-

ging the above inequality into the definition of $K(t)$, we have

$$\begin{aligned} K(t) &\geq \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} \sum_{c\in\mathcal{C}^v} \int_0^t \left(\frac{x_c(\tau)}{\sum\limits_{c'\in\mathcal{C}^v} x_{c'}(\tau)} \right)^{\alpha} dx_c(\tau) \\ &\quad + \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \|\mathbf{x}^v(0)\|_{\alpha+1} \\ &\geq \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \left(\|\mathbf{x}^v(t)\|_{\alpha+1} - \|\mathbf{x}^v(0)\|_{\alpha+1} \right) \\ &\quad + \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \|\mathbf{x}^v(0)\|_{\alpha+1} \\ &= \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \|\mathbf{x}^v(t)\|_{\alpha+1}. \end{aligned}$$

Let us define

$$H(t) := \frac{1}{\bar{\mu}\bar{\rho}^{\alpha}} \sum_{v\in\mathcal{V}} (s_v)^{\alpha} |\mathcal{C}^v|^{-\frac{\alpha^2}{\alpha+1}} \|\mathbf{x}^v(t)\|_{\alpha+1}. \tag{3.34}$$

Thus, we can conclude $K(t) \geq H(t) \geq 0$, where the non-negativity of $H(t)$ is straightforward, and $H(t) = 0$ only when $x_c(t) = 0, \forall c \in \mathcal{C}$. Therefore, if we take

$$T = \frac{K(0)}{\epsilon s_{min}^{\alpha} \rho_{bound}^{1-\alpha} \min\{1, (\max_v |\mathcal{C}^v|)^{1-\alpha}\}},$$

$K(t) = H(t) \equiv 0$ when $t \geq T$, implying $x_c(t) \equiv 0$ eventually for all $c \in \mathcal{C}$. Thus the system is positive recurrent. □

3.5 Performance Evaluation

One might think by introducing inter-slice protection, SCS effectively imposes additional constraints to the service discipline, thus is compromised in users' performance. However, this needs not to be true, as we will demonstrate via extensive simulations in this section. We compare the performance of SCS versus several benchmarks, including:

1. Dominant Resource Fairness (DRF) [3], which is a variation of weighted maxmin fairness where users' weights are associated with their resource demands. Here to incorporate network slicing, we use its variation where a user's weight is also associated with equal intra-slice weight allocation, i.e., $w_u = \frac{s_v}{N_v} \cdot \delta_u, u \in \mathcal{U}^v$, where δ_u is the dominant share of user u and is given by $\delta_u := \frac{1}{\max_{r \in \mathcal{R}} d_c^r}, u \in \mathcal{U}_c$.

2. (Discriminatory) Processor Sharing (DPS)[4, 5]. To apply to the multi-resource case, we implement DPS as a variation of maxmin fairness where user u's weight is $w_u = s_v, u \in \mathcal{U}^v$, without the notion of per-slice share constraint and inter-slice protection.

Note that because SCS might be hard to scalably compute for general α, we propose the use of $\infty-$SCS, as a surrogate resource allocation scheme.

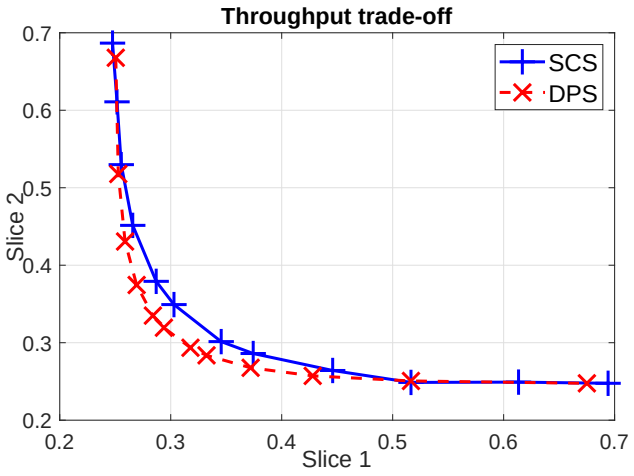
In our simulations, we focus on two performance metrics: mean delay and mean throughput. The delay is defined as the sojourn time of each user, i.e., how long it takes for a user to complete service. The throughput is defined

as the workload divided by the sojourn time of each user. The performance of different sharing schemes were compared in a range of settings, from a simple single resource setting, to more complex cases where different services/tasks are coupled together through shared resources.

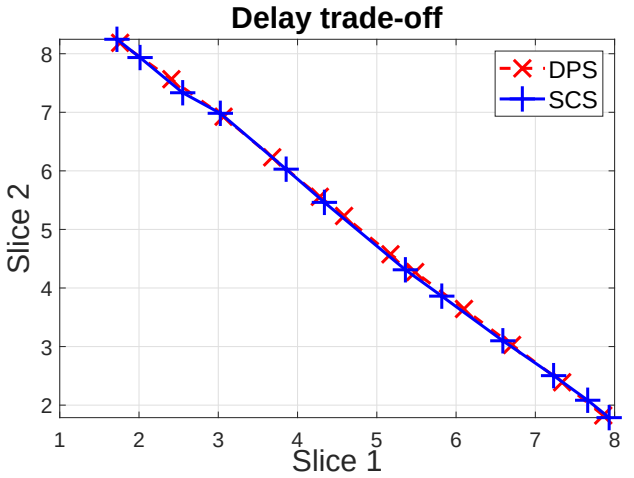
3.5.1 Single-Resource Case

Since for more complicated network setup, the system performance (for example, processing rate) is often determined by resource allocations at certain 'bottleneck' resources, we first consider single-resource setting. Note that, under such circumstances, SCS coincides with Generalized Processor Sharing (GPS) [62] as well as DRF because all classes of users $c \in \mathcal{C}$ are associated with the same resource, and have the same demand.

To begin with, we consider a simple scenario where $|\mathcal{V}| = 2$, and each slice only supports one user class, so in this setting, a user class corresponds to a slice. Two slices shares one resource, referred to as Resource 1 with capacity 1, and $d_1^1 = d_2^1 = 1$. Their traffic models are assumed to be symmetric, with mean arrival rates $\nu_1 = \nu_2 = 0.45$ and mean workloads $\frac{1}{\mu_1} = \frac{1}{\mu_2} = 1$. Their shares, however, are tuned to achieve different performance trade-offs. The share of Slice 1, s_1, ranges from 0.01 to 0.99, while $s_2 = 1 - s_1$. The achieved mean user perceived delay and throughput are illustrated in Fig. 3.2. One can see that while the average delays are marginally better under $\infty-$SCS, $\infty-$SCS clearly outperforms DPS on the average throughput. For example, when two slices have the same share $s_1 = s_2 = 0.5$, SCS increases the throughput of



(a)



(b)

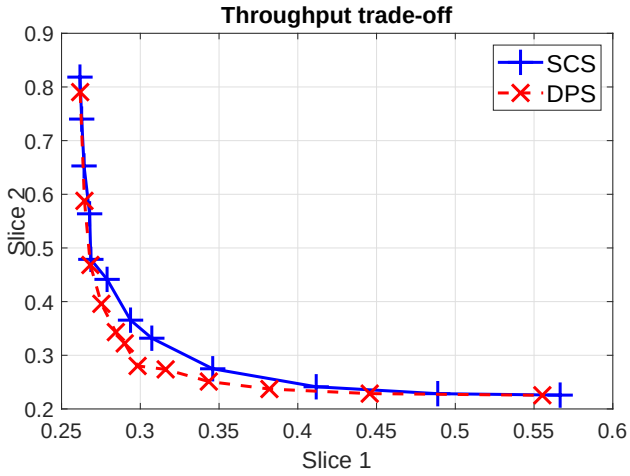
Figure 3.2: Performance trade-offs of single-resource case under symmetric traffic.

users on both slice by ∼10%.

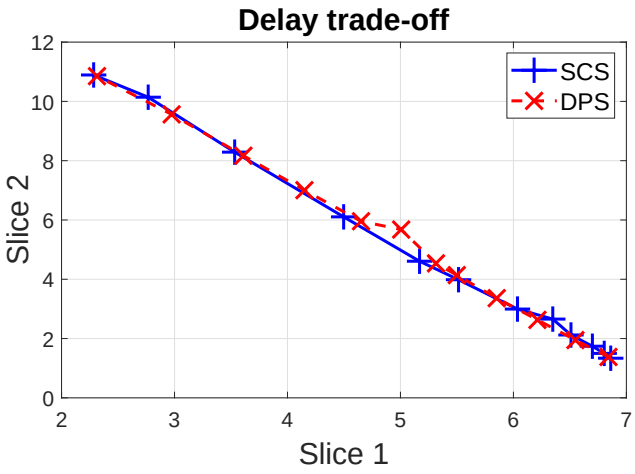
This phenomenon was widely observed under different traffic assumptions. For example, when the traffics are asymmetric, with mean arrival rates

$\nu_1 = 0.6, \nu_2 = 0.3$ and mean workload $\frac{1}{\mu_1} = \frac{1}{\mu_2} = 1$, the results are illustrated in Fig. 3.3. Also, for symmetric traffics with arrival rates of 0.45 and the workloads are set to a constant 1, the results are shown in Fig. 3.4. In general, while the mean delay achieved by SCS is marginally better than DPS, the mean throughput achieved is improved significantly.

To explain the somewhat surprising result, we conjectured that due to the inter-slice protection built into SCS, under stochastic traffic, the slice with fewer customers tends to see higher processing rate than other sharing criterion, as a result the customers leave the system faster. Overall, SCS tends to separate the busy periods of slices, so that the level of inter-slice contention is reduced. We validated our conjecture by measuring the busy period under the symmetric traffic pattern, where the arrival rates of both slices are the same, and are tuned from 0.05 to 0.45, with $s_1 = s_2 = 0.5$. Other parameters are the same as in the setting in Fig. 3.2. We plot the fraction of times when there is only one busy slice and both slices are busy, vs. the effective traffic intensity $\rho = \frac{\nu_1}{\mu_1} + \frac{\nu_2}{\mu_2}$ in Fig. 3.5. One can see that, for both SCS and DPS, the time fraction when both slices are busy increases with ρ, and that when only one slice is busy first increases when ρ is low due to underutilization, but decreases when ρ is high because the inter-slice contention becomes inevitable. However the time fraction when both are busy is always smaller under SCS than that under DPS.



(a)

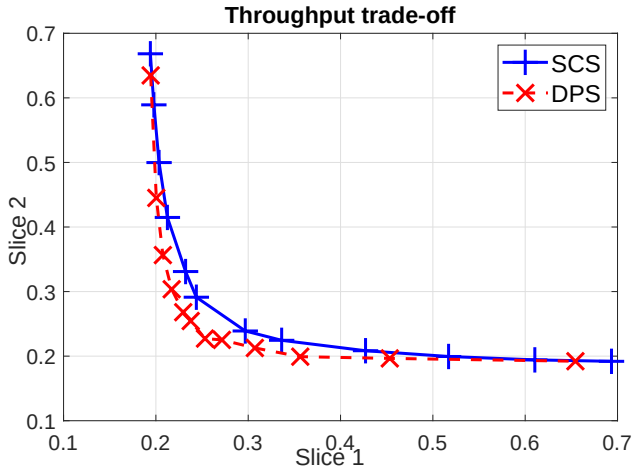


(b)

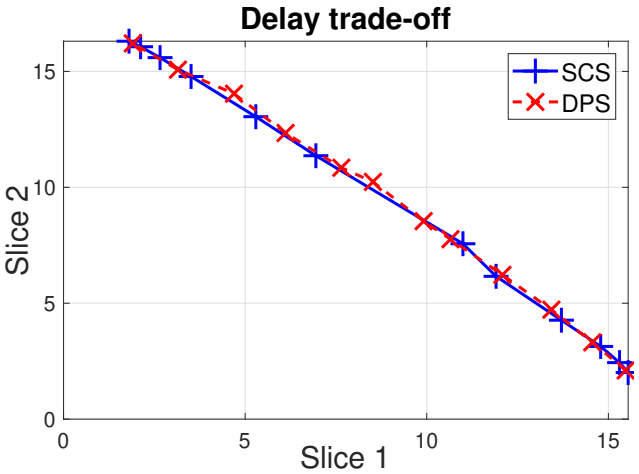
Figure 3.3: Delay and throughput trade-offs of single-resource case under asymmetric traffic.

3.5.2 Multi-Resource Cases

We also test the performance of SCS under a more complex setting where a simple cellular networks with both fronthaul and backhaul resources



(a)



(b)

Figure 3.4: Delay and throughput trade-offs of single-resource case under symmetric M/D/1 traffic model.

are simulated.

Let us consider a setting with 6 fronthaul resources, 3 backhaul resources, and a cloud computing resource. This system supports two slices,

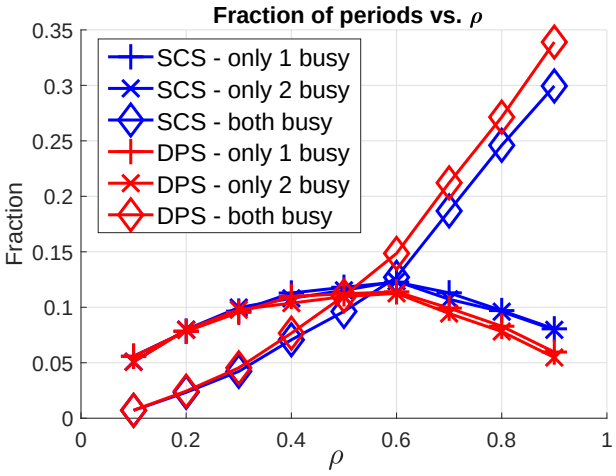


Figure 3.5: Busy period of slice 1 and 2 vs. load intensity.

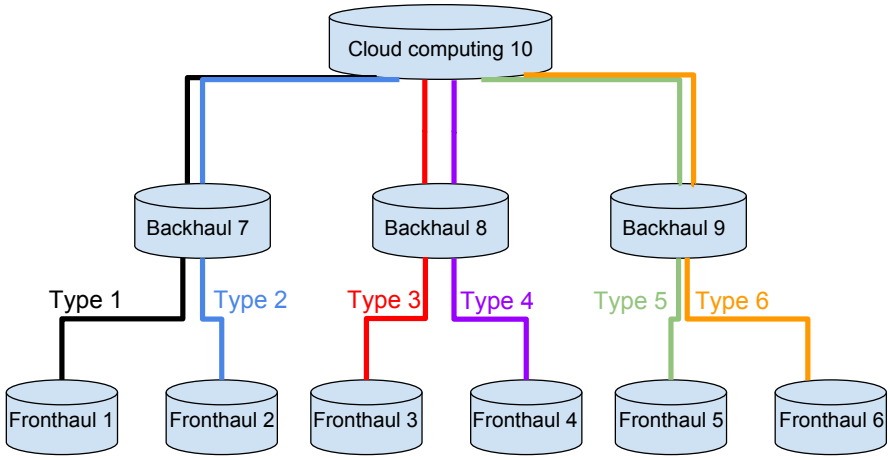


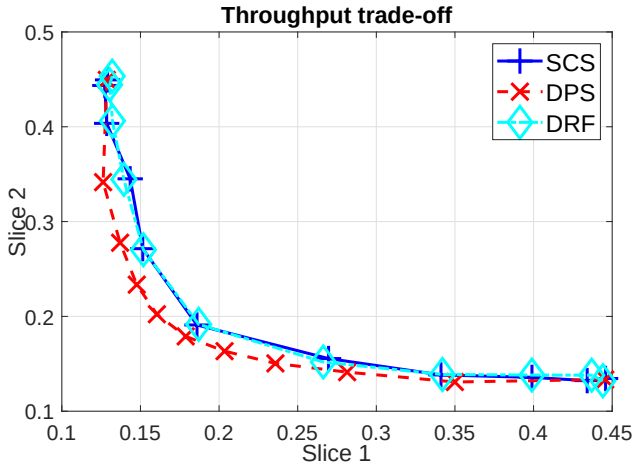
Figure 3.6: Association between user classes and resources.

each containing 3 user classes. Slice 1 includes Classes 1,2 and 3, while Slice 2 includes Classes 4, 5 and 6. The association between user classes and resources is demonstrated in Fig. 3.6, and the demand vectors, as well as the arrival rates and mean workloads, are given in Table 3.2. Slice 1's share is ranged from 0.1 to 0.9, while $s_2 = 1 - s_1$. The achieved performance trade-

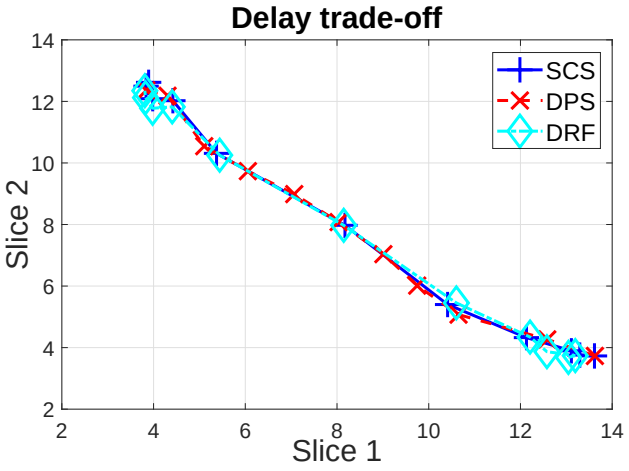
User class	Demand vector	Mean workload	Arrival rate
Class 1	$(\frac{5}{6}, 0, 0, 0, 0, 0, 0.5, 0, 0, 0.217)$	1	0.7
Class 2	$(0, \frac{5}{6}, 0, 0, 0, 0, 0.5, 0, 0, 0.217)$	1	0.7
Class 3	$(0, 0, 1, 0, 0, 0, 0, 0.625, 0, 0.217)$	1	0.7
Class 4	$(0, 0, 0, 1, 0, 0, 0, 0.625, 0, 0.217)$	1	0.7
Class 5	$(0, 0, 0, 0, 1, 0, 0, 0, 0.625, 0.217)$	1	0.7
Class 6	$(0, 0, 0, 0, 0, 1, 0, 0, 0.625, 0.217)$	1	0.7

Table 3.2: Example resource allocation in simulation

offs under different sharing criteria are illustrated in Fig. 3.7. One can see that both SCS and DRF outperform DPS in throughput, with similar mean delays under all 3 criteria. Similar results are observed in a range of settings with different traffic patterns and resource demands. Moreover, in Fig. 3.8, we adjust the weighting schemes used in DRF by voiding SCWA. Instead, $w_u = s_v \delta_u, u \in \mathcal{U}^v$, and the resources are provisioned according to DPS with weight w_u. The results show that without SCWA, DRF is similar to DPS in both throughput and delay. Therefore, we can conclude that SCWA is the root cause of the desirable performance, and SCS can even improve the system performance while providing inter-slice protection.

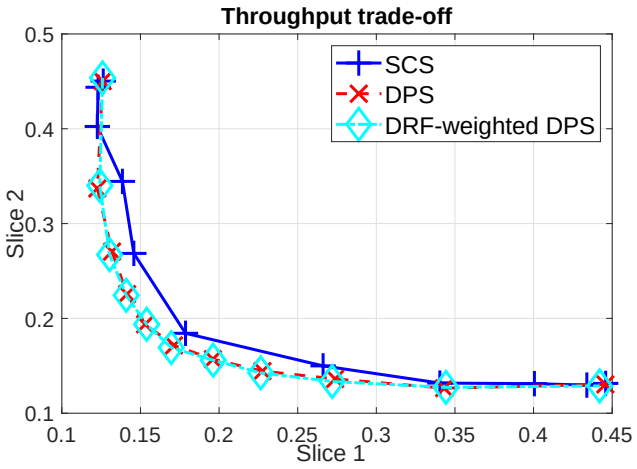


(a)

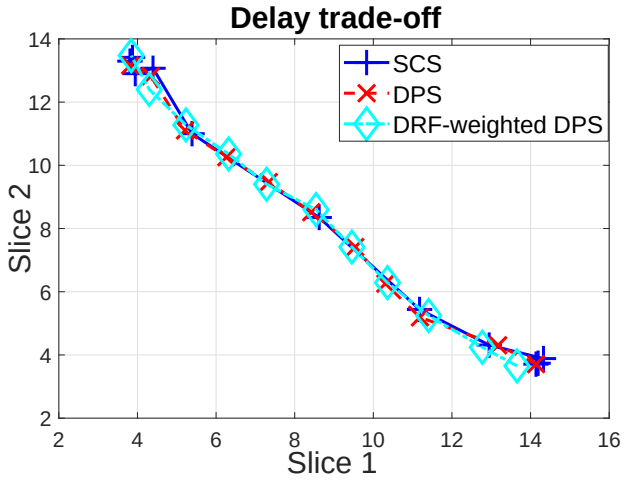


(b)

Figure 3.7: Performance trade-offs of multi-resource case.



(a)



(b)

Figure 3.8: Performance trade-offs with DRF-weighted DPS.

Chapter 4

Conclusion and Future Work on Network Slicing

This thesis discussed a sharing-based resource allocation scheme for network slices, which accommodates the dynamic user distributions and distributed resources characteristic of today's mobile services and/or applications. The approach specifies shares of distributed resource pools to be allocated to network slices via a slice-level agreement. Each slice can then distribute its share to its users, and in turn the user level resource allocations are based on the users' subshares.

There is still a long way to go before an efficient and practical implementation of a distributed resource sharing mechanism is established for network slicing context. Some aspects that will need further attention are discussed below.

Comparison with other network slicing mechanisms. This thesis studied share-based resource allocation mechanisms. There are, of course, other alternatives. For example, a mechanism could allocate resources to users on

This chapter is partially based on the results developed in previous publications. Please see Chapters 2 and 3 for details.

different slices in a FCFS manner, then apply admission control according to quotas of different slices, or charge slices at different prices according to the level of congestion. We believe that share-based slicing mechanisms such as those described in this thesis provide a better interface towards making longer term service level agreements, which provide inter-slice protection while enabling efficient sharing of distributed resources. However it is still of interest to discuss under what conditions the share-based resource allocation is superior to other possibilities.

Adaptive share dimensioning. This thesis discussed share dimensioning problem and pricing strategy associated to SCPF for fixed network loads. However specifying shares across slices is a broad topic per se. When service requirements and/or user traffic are changing, share dimensioning could be adapted. Developing a systematic approach to adapting slice shares to changes in the underlying service traffic profiles or tenant demands would be of interest.

Alternative share specification approach. Besides defining a single share parameter for the overall network resources for each slice, slice shares could be specified in a more fine-grained manner. Slices may wish to request different shares of different types of resources pools and/or different regions, due to imbalanced user loads. Thus, effectively, multiple resource pools should be defined, which are coupled through users' service requirements. Also, it would be of interest to devise schemes where each slice is guaranteed specific shares of network resources (when needed), yet has the option to reallocate these when traffic on other slices permits.

Data-driven performance management. In our study of per-slice performance management, knowledge of user traffic patterns was assumed to be available. However in practice, load distributions and dynamics need to be estimated based on collecting user data. Thus, it is of interest to explore the use of machine learning techniques to model traffic and manage user performance on slices.

Mixed traffic patterns. In this thesis, one implicit but important underlying assumption was that the users were either all inelastic or all elastic. In practice, one would expect to have both types of traffic sharing network resources. For example, some slices might support rate-adaptive applications such as video streaming and live chatting while others support elastic services like file transfers and/or web browsing. Alternatively, a single slice could have both elastic traffic and inelastic traffic. Therefore, it would be of interest to study how to allocate resources among slices supporting different traffic types, or, even address a more fundamental question as whether to define slices supporting mixed traffic or to split them up to allow better performance management.

Mapping slice-based virtual resources to actual resources. SCPF, as well as SCS specify a policy for how network slices might dynamically share virtual resources. This can be directly applied when distributed resources are homogeneous and can be shared in a manner as simple as TDMA. However, when multiple resource types are involved, one must develop flexible mappings of virtual resource allocation to physical resources. Thus different interfaces need

to be defined to interact with different underlying technologies. For example, a mobile data analysis application may require edge computing resources as well as connectivity resources. Thus a scheduler will need to orchestrate the provisioning of different types of resources in the time domain.

Part II

Human-Machine Interactive Processes

Chapter 5

Modeling and Optimization of Human-Machine Interactions

5.1 Introduction

5.1.1 Background and Motivations

Computing and information systems are increasingly prevalent in our daily lives, forming a variety of human-in-the-loop systems. Many such systems are interactive in the sense that, humans and machines take decisions/actions in response to each other, forming a sequence driven by unknown dynamics associated with human behavior. For instance, one can view web searches as an interactive process, where humans' search history, attention, and eventual decisions reflect an interaction with the machine's sequencing, placement and timing of advertisements. The industry refers to such interactive processes as 'convergence paths' and is increasingly interested in optimizing their outcomes [63]. Such problems involving interactions are usually studied under the context of Markov decision processes (MDP) and its variants, see, e.g., [64, 65, 66]. However, the actual problem associated with interactive processes presents several challenges which remain unsolved, including the following.

Complexity of inferring interactive human behavior. In this chapter we will focus on structured human-machine interactions where one has modeled

both human and machine behaviour/choices over time, and the setting arises repeatedly either by the same person or by a large population. The outcomes of such interactions can depend subtly on the history thus one can expect exponential complexity to be a challenge–unless the underlying processes have a 'nice' structure. Such assumptions are essential for widely studied problems including MDP [64, 65, 66], reinforcement learning [67, 68], and multi-armed bandit problem [69], where human decision-making processes are assumed to be independent across time, or have one-step Markov property. However, those assumptions are questionable according to studies on human cognition, see, e.g., [70].

One recent work considering long-term dependency is deep Q-learning [71], where authors used a complex neural network to capture the potential value of a state-action pair, where the state may incorporate complex historical information. However, because most interactive processes are transient, as both human and machine accumulate a history of decisions over time, one might expect the data requirements of carrying out deep Q-learning is quite high.

Biases in collecting data in interaction processes. Inferring a model for human behavior in the context of human-machine interaction process is also challenging because to collect data one must choose a machine policy to 'interact' with humans. This may in turn lead to 'biased' inferences of human behavior. In particular, a machine policy that focuses on 'rewarding' actions may preclude exploration/observation of other interaction modalities.

Similar considerations have been explored in partially observable MDP [72] to improve the efficiency of the solution, and in multi-armed bandit problem [69, 73] to achieve better exploration-exploitation trade-off. However, data-driven models and inferences should respect the *causal* nature of human-machine interactions, but how to model/promote the randomness of a *causal* model remains unknown.

Robustness and exploration in optimizing machine interactions. A general data-driven framework for modeling and optimizing human-machine interaction processes might be viewed as involving two concerns. On one hand, engaging humans in interaction to collect data to infer models of human behavior, and on the other, using models of human behavior to choose machine policies optimizing interaction 'rewards,' i.e., the effectiveness of the sequence of machine actions in nudging human towards desirable outcomes. To that end, it is desirable to choose machine policies which are not overly sensitive to sampling noise in data collection and/or variability in human behavior. Also, of interest are policies that are not overly deterministic/predictable, as in some settings, such policies may be poor in keeping humans 'engaged', see, e.g., [74], and poor in eliciting rich human-machine data sets.

5.1.2 Contributions

In this chapter, we propose a data-driven framework to jointly solve the estimation and optimization problems associated with human-machine interaction processes. We adopt an inference technique based on a constrained

maximum entropy principle for interactive processes, see [75, 76]. This allows one to incorporate prior knowledge of the (possibly) relevant features of human behavior, via moment constraints associated with interaction functions. We consider optimizing machine policies based on an interaction reward function with an entropy-based regularization term. This aims to find machine policies which maximize rewards, are robust to estimation noise, and maintain a degree of exploration when interacting with humans. Our proposed Alternating Reward-Entropy Ascent (AREA) Algorithm, alternates between data-collecting, estimation of human behavior, and the optimization of machine policy, with a view on reaching a consistent fixed point. We provide a characterization of various properties of AREA. In particular, for decomposable and/or path-based feature and reward functions, we devise a computationally efficient approach to estimation and optimization steps. The approach takes advantage of defining a stopping time over the interaction and the conditional Markov property of the estimated human model, to significantly reduce space and time complexity. We provide a theoretical characterization of the AREA algorithm in terms of its convergence, along with simple preliminary evaluation results based on synthetic data obtained from a noisy nonlinear model for human decision-making.

5.2 Problem Formation

We shall consider a structured discrete time human-machine *interaction process* over a period of time $1, 2, \ldots, T$, which can be viewed as a pair of sequences of random variables, $(H_1, \ldots, H_T)$ corresponding to human actions/responses if any, and $(M_1, \ldots, M_T)$ denoting those of the machine. We shall assume the random variables H_t, M_t capture discrete human and machine actions at time t, and, without loss of generality, that for all t, $H_t \in \mathcal{H}$, where $\mathcal{H}$ denotes the human's action space, and $M_t \in \mathcal{M}$, where $\mathcal{M}$ corresponds to the machine's action space[1]. Throughout this paper we assume that $|\mathcal{H}|$ and $|\mathcal{M}|$ are finite. To simplify notation, we let $H^t = (H_1, \ldots, H_t)$ for $t = 1, \ldots, T$, and similarly define M^t. When $t = 0$, H^t or M^t contains no elements. We assume that the human and the machine take turns, such that the machine's action at time $t+1$, i.e., M_{t+1} depends on H^t, M^t while that of the human at time $t+1$, i.e., H_{t+1} depends on H^t, M^{t+1}. The joint distribution of (H^T, M^T) captures the interplay between the human and machine. Depending on the setting, the human refers to a particular individual or a population, where the behavior can be captured via a stable distributional model.

We shall assume that when a human and machine interact, the machine's policy is known and captured by a collection of conditional distributions Q, for succinctness denoted by $Q(m_t|h^{t-1}, m^{t-1}) := p_{M_t|H^{t-1},M^{t-1}}(m_t|h^{t-1}, m^{t-1})$ for $t = 1, \ldots T$. Similarly human behavior is denoted by conditional distribu-

[1]Note both the human and/or machine could choose to do nothing in their turn. This can be included in our model by including null action in both $\mathcal{M}$ and $\mathcal{H}$.

tions P given by $P(h_t|h^{t-1}, m^t) := p_{H_t|H^{t-1},M^t}(h_t|h^{t-1}, m^t)$ for $t = 1, \dots T$. It is easy to show that joint distribution of (H^T, M^T), denoted by PQ, resulting from a human model P interacting with a machine policy Q, can be decomposed as $PQ(h^T, m^T) = P(h^T||m^T)Q(m^T||h^T)$, where

$$P(h^T||m^T) := \prod_{t=1}^{T} P(h_t|h^{t-1}, m^t) \quad \text{and} \quad Q(m^T||h^T) := \prod_{t=1}^{T} Q(m_t|h^{t-1}, m^{t-1}), \tag{5.1}$$

correspond to the *causally conditioned distributions* of the human and the machine, i.e., products of sequentially conditioned distributions. We will assume that data of human-machine interactions can be collected by fixing a machine policy, and keeping track of the realizations of such interactions.

We let $P^*(h^T\|m^T)$ denote the *true* human behavior and $\hat{P}(h^T\|m^T)$ an estimated model thereof. We let $PQ(A)$ denote the probability of an event A measurable w.r.t. (H^T, M^T) and we let $E_{PQ}[f(H^T, M^T)]$ denote the expectation of a function $f(h^T, m^T) : \mathcal{H}^T \times \mathcal{M}^T \to \mathbb{R}$ under the joint distribution PQ. When we collect interaction data of the human with a machine policy $Q(m^T\|h^T)$ we denote expected value under the associated empirical distribution by $\hat{E}_{P^*Q}$ where in the ideal case (no noise) we have $\hat{E}_{P^*Q}[f(H^T, M^T)] = E_{P^*Q}[f(H^T, M^T)]$.

Those notations are summarized in Table 5.1.

5.2.1 Data-Driven Human Model Estimation

A brute force approach to modeling human behaviour would be to directly estimate the conditional distributions $\{P^*(h_t|h^{t-1}, m^t), t = 1, \dots, T\}$

Table 5.1: Key notation used in Chapter 5.

Sequence of human actions	$H^t = \{H_1, H_2, \cdots, H_t\},\ 0 \leq t \leq T$
Specific realization of human action sequence	$h^t = \{h_1, h_2, \cdots, h_t\},\ 0 \leq t \leq T$
Sequence of machine actions	$M^t = \{M_1, M_2, \cdots, M_t\},\ 0 \leq t \leq T$
Specific realization of machine action sequence	$m^t = \{m_1, m_2, \cdots, m_t\},\ 0 \leq t \leq T$
Joint PMF of M^T, H^T	$p_{H^T,M^T}(h^T, m^T)$
Causally conditional distribution of human actions given machine actions	$p_{H^T\|M^T}(h^T\|m^T)$ or $P(h^T\|m^T)$
Causally conditional distribution of machine actions given human actions	$p_{M^T\|H^T}(m^T\|h^T)$ or $Q(m^T\|h^T)$
Joint PMF when the human model is $P(h^T\|m^T)$ and machine model is $Q(m^T\|h^T)$	$PQ(h^T, m^T)$
Probability of event A when the human model is $P(h^T\|m^T)$ and machine model is $Q(m^T\|h^T)$.	$PQ(A)$
Expectation of function of interactions $f(H^T, M^T)$ w.r.t. the joint PMF given by $PQ(h^T, m^T)$	$\mathbb{E}_{PQ}[f(H^T, M^T)]$
The actual human behavior model	$P^*(h^T\|m^T)$ or P^*
The estimated human behavior model if the machine model is Q	$\hat{P}(h^T\|m^T) = h(Q)$ or $\hat{P} = h(Q)$
The machine model if the estimated human model is P	$\hat{Q}(m^T\|h^T) = m(P)$ or $\hat{Q} = m(P)$

based on the collected data which is clearly not scalable. Instead, in this paper we embrace the extension of constrained maximum entropy estimation to interactive processes developed in [76, 77].

In this setting, one defines a set of feature functions ideally known to capture relevant characteristics of human behavior which become equality and inequality constraints in the estimation process. The choice of such features would be motivated by known frameworks for understanding *human behavior* in dynamic environments, e.g, the effort accuracy [78], exploration-exploitation [79], soft constraints [80], and specific character of the human-machine interaction. The equality constraints are based on matching the moments of a set of feature functions $\mathcal{F}$ denoted by $\mathbf{f}(h^T, m^T) := \{f^i(h^T, m^T),\ i \in \mathcal{F}\}$, and their moments based the empirical distribution when interacting with a given machine policy Q, which are denoted by $\mathbf{c}_f := \hat{E}_{P^*Q}[\mathbf{f}(H^T, M^T)]$. Below we will neglect sampling errors by assuming that $\hat{E}_{P^*Q}[\mathbf{f}(H^T, M^T)] = E_{P^*Q}[\mathbf{f}(H^T, M^T)]$. The set of inequality constraints are denoted by $\mathbf{g}(h^T, m^T) := \{g^i(h^T, m^T),\ i \in \mathcal{G}\}$, where $\mathcal{G}$ is another set of feature functions whose moments are constrained not to exceed pre-specified thresholds $\mathbf{c}_g = \{c_g^i,\ i \in \mathcal{G}\}$.

Formally, for a given machine policy $Q(m^T \| h^T)$, we are interested in models for human behaviour $P(h^T \| m^T)$ satisfying the following constraints

$$\mathcal{P}^Q_{\mathcal{F},\mathcal{G}} = \{P(h^T \| m^T) \mid E_{PQ}[\mathbf{f}(H^T, M^T)] = \mathbf{c}_f, \text{and } E_{PQ}[\mathbf{g}(H^T, M^T)] \geq \mathbf{c}_g\}. \tag{5.2}$$

The maximum entropy estimation principle chooses the model for human behaviour in $\mathcal{P}^Q_{\mathcal{F},\mathcal{G}}$ with maximum entropy. In the case of interactive

processes, since the machine policy $Q(m^T \| h^T)$ is known we shall maximize the entropy of the causally conditioned distributions of the human behavior model. In particular, the causally conditioned entropy of human behaviour model $P(h^T \| m^T)$ given machine policy in use is $Q(m^T \| h^T)$, is given by

$$\mathbb{H}_{PQ}(H^T \| M^T) := E_{PQ}\left[-\log\left(P(H^T \| M^T)\right)\right] = \sum_{t=1}^{T} \mathbb{H}_{PQ}(H_t | H^{t-1}, M^t). \tag{5.3}$$

In the sequel we consider optimizing functionals of the causally conditioned distributions for the human (and the machine). Doing so means optimizing over a set of conditioned distributions $\{P(h_t|h^{t-1}, m^t) \,|\, t = 1, \ldots, T\}$, which for simplicity we *also* denote by $P(h^T \| m^T)$. It can be shown that these collections of distributions belong to a convex polytope denoted by $\mathcal{C}_H$ (resp. $\mathcal{C}_M$). Indeed, according to [75], $P(h^T \| m^T) \in \mathcal{C}_H$ is equivalent to the requirement that $P(h^T \| m^T)$ can be factorized into a product of conditional distributions as in (5.1). Similar result holds true for $Q(m^T \| h^T)$. This generalizes the notion of optimizing over a set of distributions with a given support, e.g., over the simplex. In the sequel for the sake of simplicity, we will omit the constraints $P(h^T \| m^T) \in \mathcal{C}_H$ and $Q(m^T \| h^T) \in \mathcal{C}_M$ when they appear in optimization problems–it is assumed to be understood that one is optimizing over causally conditioned distributions that must be properly normalized. The overall human estimation problem can thus be expressed as follows.

Definition 5.2.1. (Human estimation problem) Given a known machine policy $Q(m^T \| h^T)$ and a set of moments $\mathbf{c}_f$ associated with human-machine in-

teraction for equality constraints, the constrained maximum entropy estimate model for human behavior, say $\hat{P}(h^T \| m^T)$ is the solution to the following problem:

$$\max_{P(h^T \| m^T)} \{ \mathbb{H}_{PQ}(H^T \| M^T) \mid P(h^T \| m^T) \in \mathcal{P}^Q_{\mathcal{F},\mathcal{G}} \}. \tag{5.4}$$

Note that since this problem is convex, the solution $\hat{P}(h^T \| m^T)$ is unique. However, it depends on underlying machine policy Q both through the cost function and the constraints.

5.2.2 Machine Optimization

We assume one has defined a *reward function* $r(h^T, m^T)$ over human-machine interactions. This function might reflect both desirable human outcomes/decisions as well as machine costs for taking certain sequences of actions. Given an estimated model for the human behaviour, $\hat{P}(h^T \| m^T)$, one can in turn consider choosing a reward maximizing machine policy, i.e.,

$$\max_{Q(m^T \| h^T)} \quad E_{\hat{P}Q}[r(H^T, M^T)].$$

A direct optimization of the reward as above would result in machine policies that take deterministic actions associated with the 'best' choices. Such policies are likely to be vulnerable to the error in the estimated human behaviour model due to the sampling noise. This has also been observed in the context of reinforcement learning, see, e.g., [81, 82]. Such machine policies may also be limited in the degree to which 'explore' interaction with the human, and thus subsequently the obtained interaction data may lead to poor

estimates of human behavior and sub-optimal results. Further, we also posit that deterministic machine policies have poor characteristics from a human interaction perspective, e.g., might also be boring/too predictable, leading to poor engagement [74], and/or in certain settings may be unfair. For example, in an advertising setting, one might want to incorporate randomness in placing advertisements to ensure fairness and/or encourage competition.

To address these concerns we propose adding a 'regularizing' entropy term to the reward function. Thus given an estimated model for human behavior $\hat{P}$, the machine's policy is obtained as the solution to the following problem.

Definition 5.2.2. (Machine policy optimization problem) Given an estimated model for human behavior $\hat{P}(h^T \| m^T)$, the reward maximizing machine policy is given by the solution to

$$\max_{Q(m^T \| h^T)} \quad \mathbb{H}_{\hat{P}Q}(M^T \| H^T) + \gamma E_{\hat{P}Q}[r(H^T, M^T)], \tag{5.5}$$

where $\gamma > 0$ controls the degree to which one weighs entropy versus reward in the machine policies. We shall realize that this formulation is in fact similar to human estimation problem introduced earlier.

5.2.3 Closing the Loop: Alternating Reward-Entropy Ascent (AREA) Algorithm

Note that the optimized machine policy obtained via (5.5) depends on a estimated model for human behavior, which in turn was estimated by

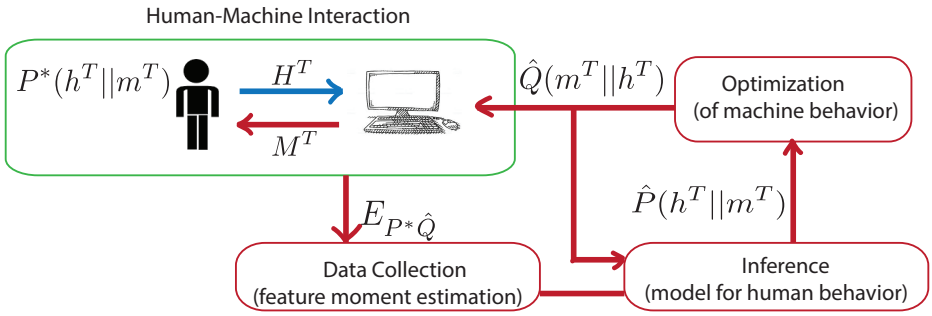


Figure 5.1: Overview of framework for the optimization of human-machine interactions.

solving (5.4) based on data obtained from human-machine interactions using the previously selected machine policy. The two machine policies need not to be the same, possibly making the estimation and optimization steps inconsistent. To resolve this, we propose *Alternating Reward-Entropy Ascent* (AREA) algorithm exhibited in Figure 5.1. We begin with a default machine policy (for example, the machine might choose actions at random), denoted by $\hat{Q}^{(0)}(m^T\|h^T)$. Under this machine policy we collect data/realizations of human machine interactions. Then from the data, we can estimate the feature moments, which, in turn, enable estimation of a model for human behavior $\hat{P}^{(0)}$ through our *inference* phase, i.e., (5.4). Based on the estimated model of human behavior we generate a new machine policy through the machine *optimization* phase, where the optimization is based on $\hat{P}^{(0)}$, obtaining the next machine policy $\hat{Q}^{(1)}$. This alternating process generates a sequence of causally conditioned distributions given by $\hat{Q}^{(0)} \to \hat{P}^{(0)} \to \hat{Q}^{(1)} \to \hat{P}^{(1)} \to \dots$, which we refer to as AREA iterations.

5.3 Related Work

Markov decision processes and reinforcement learning: The optimization of human-machine interactions can in principle be modelled as a Markov Decision Process (MDP), where the human behavior can be viewed as driven by a transition kernel among a set of states, and the machine behavior corresponds to a sequence of actions taken in response to the human's behavior. The underlying assumptions are that there exists a state space for the human and an action space such that the distribution of future states depends only on the current state (say of the human) and chosen action (say of the machine). In such a setting, one can define a reward function and consider optimizing the associated machine policy, see e.g., [64, 65, 66]. When the transition kernel is unknown, but assumed Markov, the resulting problem is known as reinforcement learning, see e.g., surveys [67, 68]. Both model-based and model-free reinforcement learning approaches (and methods that combine both approaches) have been studied in the literature. Model based methods combine estimation of the environment and optimization of machine actions, while model-free methods aim to directly optimize the machine without first estimating a model the environment. For example, Q-learning aims to directly estimate the value of state-action pair, denoted by $Q(s, a)$, where the s is the current state and a is the candidate action. The Q-function can be used to select the optimal sequence of machine actions [67, 68]. The traditional framework of reinforcement learning relies heavily on the assumption that the underlying environment is Markov. However as deep learning technologies emerge,

deep Q-learning [71] have been devised to approximately solve this problem. Indeed to overcome the difficulties brought by non-Markov environments, an option is to first enlarge the state space of the underlying environment significantly, for example, to include all possible history of the system. Then use a deep neural network to encode the Q-function, and fit the neural network to the observed data. This approach also has its challenges in terms of demanding data requirements and might not be applicable to some use cases.

In our framework, when the reward function is decomposable over time i.e. $r(h^T, m^T) = \sum_{t=1}^{T} r(h_t, m_t)$, and the estimated human model $\hat{P}(h^T||m^T)$ is one-step Markov, the machine optimization program reduces to a traditional MDP setting, with a possible time-inhomogeneous transition kernel and the reward function is regularized by an entropy term to promote exploring different actions.Some recent literature suggests that model based methods may be preferred to model free methods in terms of sample complexity [81, 82]. In the special case of a Markov model, our approach may be considered as a variation of model-based reinforcement learning, where the model is learned by maximizing causal entropy subject to moment constraints, and the machine behavior is regularized using the causal entropy of the machine process. As discussed, this analogy no longer holds for the general case.

We are aware of only a few cases where (relative) entropy regularization has been combined with Markov decision processes and related models. [83] consider a generalization of the Markov decision process where, instead of impacting the process through some actions, the agent can directly manipulate

the transition matrix of the system state. However, such manipulation would incur some cost which is proportional to the relative entropy between the transition probability after manipulation, and the transition matrix of a 'passive' process which models the 'natural' system evolution. In [72], the authors propose an entropy-regularized cost function to approximately solve a partially observable Markov decision process (POMDP) model efficiently. Due to the absence of knowledge of the exact system state (i.e. partial observation), the agent must estimate it through the reward it receives and a noisy observation of its current state. Therefore, there is a trade-off between gaining more profit based on current belief – which requires focusing on the most profitable action, and improving the quality of estimation – which requires exploring different actions. The authors of [72] used the expectation of entropy in the agent's belief state as a proxy of how well it explored different actions. The main challenge associated with MDP is that the human's behavior transition kernels may have long term dependencies – and an extremely large state space may be required state to remain in the Markov setting.

Bandit problem: The state-of-the-art approach to solving the problems with such sequential and interactive context also includes multi-armed bandit problem and its variants [69], which are widely discussed and used in use cases including computational advertising. In such context, the search engine uses the user feature including gender, age and searching history as the context, to pick up an ad, which is modeled as the arm, after each user's query, such that the user will have a good chance of clicking through the ad. The most

representative method is the ILOVETOCONBANDITS algorithm proposed in [73], where it is assumed that the reward received for each attempt depends on some observable random 'context'. However the approach depends heavily on the i.i.d. assumption on the environment, in order to improve the quality of the estimation by accumulating samples. Therefore when the user does not make independent decisions or has a long-term memory, the performance of such contextual bandit based solutions will not be acceptable.

The most general way to model such problems in a multi-armed bandit way is continuum armed bandit, for example, [84], where the arms to pick can be a vector of real numbers instead of discrete index. We can directly model the machine's policy $Q(m^T \| h^T)$ as arms. However when the support of the arm is big, the convergence of the algorithm is slow, and also it requires a prior knowledge of the number of iterations we need, thus cannot be implemented in a fully online manner.

5.4 Solution to AREA's Optimization Problems

The Lagrangians for the optimization problems (5.4) and (5.5) have similar forms. We shall begin our discussion of the solution approach, based on [76], for the human estimation problem and subsequently that of the machine optimization, pointing out some key results and notation that will be critical for our development in the sequel.

5.4.1 Solution to Human Estimation Problem

It has been shown in [76] that the human estimation problem is concave in $P(h^T \| m^T)$ given $Q(m^T \| h^T)$, and the solution can be found by its dual.

Theorem 5.4.1. *[76] The dual form of the human estimation problem (5.4) is given by:*

$$\min_{\substack{\boldsymbol{\lambda}=(\boldsymbol{\lambda_f},\boldsymbol{\lambda_g}),\\ \boldsymbol{\lambda}_g \leq 0}} \sum_{m_1} Q(m_1) \log Z_{\boldsymbol{\lambda}}(m_1) - \boldsymbol{\lambda}_f^T \mathbf{c}_f - \boldsymbol{\lambda}_g^T \mathbf{c}_g \tag{5.6}$$

where

$$Z_{\boldsymbol{\lambda}}(h^t, m^{t+1}) = \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|h^t, m^{t+1}), \quad Z_{\boldsymbol{\lambda}}(m_1) = \sum_{h_1} Z_{\boldsymbol{\lambda}}(h_1|m_1) \tag{5.7}$$

and

$$Z_{\boldsymbol{\lambda}}(h_t|h^{t-1}, m^t) = \begin{cases} e^{\sum_{m_{t+1}} Q(m_{t+1}|h^t,m^t) \log Z_{\boldsymbol{\lambda}}(h^t,m^{t+1})} & t < T \\ e^{\boldsymbol{\lambda}_f^T \mathbf{f}_1(h^T,m^T) + \boldsymbol{\lambda}_g^T \mathbf{f}_2(h^T,m^T)} & t = T \end{cases}, \tag{5.8}$$

The associated human model for dual variables $\boldsymbol{\lambda}$ is given by $P_{\boldsymbol{\lambda}}(h_t|h^{t-1}, m^t) = \frac{Z_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t)}{Z_{\boldsymbol{\lambda}}(h^{t-1},m^t)}$.

The optimal dual $\boldsymbol{\lambda}^*$ can be found by subgradient-based method. Theorem 4 in [76] shows the strong duality of Problem (5.4). Therefore, the optimal dual is the optimal primal solution. Thus, if we find a $\boldsymbol{\lambda}^*$ minimizing (5.6), the estimated human model $\hat{P}$ is given by $P_{\boldsymbol{\lambda}^*}$. The dual problem can be solved via a subgradient-based algorithm. In particular if we use an adaptive learning rate $\eta^{(n)} \in \mathbb{R}^+$, the dual variable should be updated by

$$\begin{aligned} \boldsymbol{\lambda}_f^{(n+1)} &\leftarrow \boldsymbol{\lambda}_f^{(n)} - \eta^{(n)}(E_{P_{\boldsymbol{\lambda}}Q}[\mathbf{f}(H^T, M^T)] - \mathbf{c}_f), \\ \boldsymbol{\lambda}_g^{(n+1)} &\leftarrow \max\{0, \boldsymbol{\lambda}_g^{(n)} - \eta^{(n)}(E_{P_{\boldsymbol{\lambda}}Q}[\mathbf{g}(H^T, M^T)] - \mathbf{c}_g)\}, \end{aligned} \tag{5.9}$$

where $\mathbf{c}_f = E_{P^*Q}[\mathbf{f}(H^T, M^T)]$ are the moments of the feature functions associated with the equality constraints obtained from the human-machine interaction data in the inference step, and the gradients are computed using the recursive form defined in Theorem 5.4.1. Then the sequence $\{\boldsymbol{\lambda}^{(n)}\}$ converges to the optimal dual $\boldsymbol{\lambda}^*$.

In the sequel it will be useful to denote the solution to the human estimation problem by $h^*(Q, \mathbf{c}_f, \mathbf{c}_g)$ to make clear its dependence on Q the machine policy, $\mathbf{c}_f$ the feature moments estimated from human-machine interactions, and the constants $\mathbf{c}_g$.

The solution given in Theorem 5.4.1 has several interpretations, two of which are given in following two theorems.

Theorem 5.4.2. *[76] Using statistics from the true distribution without sampling error, maximizing the causal entropy subject to feature constraints in human estimation problem is equivalent to maximizing the log causal likelihood of the true distribution over the family of* causal Gibbs distributions*.*

$$\max_{\boldsymbol{\lambda}} \quad E_{P^*Q}[\log P_{\boldsymbol{\lambda}}(H^T \| M^T)] \tag{5.10}$$

Theorem 5.4.3. *[76] The human estimation problem is equivalent to minimizing the* worst case causal log-loss *when the true human behavior is chosen adversarially.*

$$\begin{aligned} \inf_{P(h^T \| m^T)} \sup_{P^*(h^T \| m^T)} \quad & E_{P^*Q}[-\log P(H^T \| M^T)] \\ \textit{such that:} \quad & E_{PQ}[\mathbf{f}(H^T, M^T)] = E_{P^*Q}[\mathbf{f}(H^T, M^T)] \end{aligned} \tag{5.11}$$

5.4.2 Solution to Machine Optimization Problem

It should be clear at this point that the the objective function in (5.5) is similar to the Lagrangian of Problem (5.4) with a fixed 'dual variable' γ. Thus the following result is fairly straightforward.

Theorem 5.4.4. *For a given model of human behavior $\hat{P}(h^T\|m^T)$ the solution to the machine optimization problem (5.5), $\hat{Q}(m^T\|h^T)$ is given as follows. Let*

$$Y_\gamma(m_t|h^{t-1},m^{t-1}) = \begin{cases} e^{\sum_{h_t}\hat{P}(h_t|h^{t-1},m^t)\log Y_\gamma(h^t,m^t)} & t<T \\ e^{\gamma\sum_{h_T}\hat{P}(h_T|h^{T-1},m^T)r(h^T,m^T)} & t=T \end{cases}, \tag{5.12}$$

where $Y_\gamma(h^t,m^t) = \sum_{m_{t+1}} Y_\gamma(m_{t+1}|h^t,m^t)$, $Y_\gamma = \sum_{m_1} Y_\gamma(m_1)$. Then the optimal machine policy is $\hat{Q}(m_t|h^{t-1},m^{t-1}) = \frac{Y_\gamma(m_t|h^{t-1},m^{t-1})}{Y_\gamma(h^{t-1},m^{t-1})}$ and $\hat{Q}(m_1) = \frac{Y_\gamma(m_1)}{Y_\gamma}$.

Proof. The machine optimization Problem (5.5) can be shown to be concave in Q thus one can directly solve it via first-order optimality conditions. Considering the variables to be $\{Q(m_t|h^{t-1},m^{t-1}), t=1,2,\dots T, h^{t-1}\in\mathcal{H}^{t-1}, m^t\in\mathcal{M}^t\}$, the Lagrangian associated with Problem (5.5) can be written as:

$$\begin{aligned}\Lambda(Q,\beta) =& \mathbb{H}_{\hat{P}Q}(M^T\|H^T) + \gamma E_{\hat{P}Q}[r(H^T,M^T)] \\ &- \sum_{\substack{1\le t\le T \\ h^{t-1}\in\mathcal{H}^{t-1} \\ m^{t-1}\in\mathcal{M}^{t-1}}} \beta(h^{t-1},m^{t-1})\left(1-\sum_{m_t} Q(m_t|h^{t-1},m^{t-1})\right),\end{aligned}$$

where $\beta(h^{t-1},m^{t-1})$ for $t=1,\dots,T$ denote dual variables associated with the respective normalization constraints $\sum_{m_t} Q(m_t|h^{t-1},m^{t-1}) = 1$. Now differ-

entiating the Lagrangian we have

$$\nabla_{Q(m_t|h^{t-1},m^{t-1})}\Lambda(Q,\beta) =$$
$$\beta(h^{t-1},m^{t-1}) + \hat{P}Q(h^{t-1},m^{t-1})\left(-\log Q(m_t|h^{t-1},m^{t-1}) - 1\right.$$
$$\left. + \mathbb{H}_{\hat{P}Q}\left(M^T\|H^T|h^{t-1},m^t\right) + \gamma E_{\hat{P}Q}\left[r(H^T,M^T)|h^{t-1},m^t\right]\right),$$

where $\mathbb{H}_{\hat{P}Q}(M^T\|H^T|h^{t-1},m^t)$ is the further conditioned, causally conditioned entropy, defined as:

$$\mathbb{H}_{\hat{P}Q}\left(M^T\|H^T|h^{t-1},m^t\right) := E_{\hat{P}Q}\left[-\log Q(M^T\|H^T) \mid H^{t-1}=h^{t-1}, M^{t-1}=m^{t-1}\right].$$

Now plugging Y_γ defined recursively in Theorem 5.4.4, and setting $\beta(h^{t-1},m^{t-1}) = \hat{P}Q(h^{t-1},m^{t-1}) + \log Y_\gamma(h^{t-1},m^{t-1})\hat{P}Q(h^{t-1},m^{t-1})$, we can show that $\nabla_{Q(m_t|h^{t-1},m^{t-1})}\Lambda(Q,\beta) = 0$. Thus the optimal solution is achieved. □

In the sequel it will be useful to represent the result stated in Theorem 5.4.4 as follows. In particular the auxiliary function $\mathbf{Y}_\gamma := \{Y_\gamma(m_t|h^{t-1},m^{t-1}), \forall 1 \le t \le T\}$ generated by the procedure given in Theorem 5.4.4 depends on the human model and so is denoted by $\mathbf{Y}_\gamma = m(\hat{P})$. The associated optimal machine policy $\hat{Q}$ is in turn a function of $\mathbf{Y}_\gamma$ denoted by $\hat{Q} = m^*(\mathbf{Y}_\gamma)$.

5.5 Complexity of AREA Algorithm

As can be seen, the dual problem of human estimation problem is over a vector $\boldsymbol{\lambda}$ of dimension $|\mathcal{F}| + |\mathcal{G}|$. The authors of [76] shows that we can find the optimal dual variables by a recursion only involves computing the expectation of feature functions, respect to joint distribution $P_{\boldsymbol{\lambda}}Q$, where $P_{\boldsymbol{\lambda}}$ is the human distributional model associated with $\boldsymbol{\lambda}$. However when updating the dual variables, computing those expectations are intractable in the most general setting. Specifically, if we define the space complexity as the number of variables that need to be stored, and the time complexity as the number of basic math operations (e.g. addition, multiplication and exponential function evaluation) required to carry out the update, we can see that because the number of conditioning sequences in (5.8) grows exponentially in T, thus if we need to put all conditional PMFs into the memory and then compute the joint PMF accordingly, both space and time complexities required are exponential in T. Fortunately, when the feature functions have specific forms, the complexity of computing such updates can be reduced. Specifically, we will discuss cases where one iteration of AREA algorithm described in Section 5.2.3 has polynomial complexity in T.

Definition 5.5.1. A feature function $f(h^T, m^T)$ is said to be decomposable if it can be written as $f(h^T, m^T) = \sum_{t=1}^{T} f_t(h_t, m_t)$.

Definition 5.5.2. A function $f(h^T, m^T)$ is said to be path-based if it is proportional to the indicator function of a specific realization of the human-machine interaction, say $(\bar{h}^T, \bar{m}^T)$, i.e., $f(h^T, m^T) = c\mathbf{1}_{\{(h^T, m^T) = (\bar{h}^T, \bar{m}^T)\}}$.

Note that it is always desirable to include the reward function in the equality feature set $\mathcal{F}$ to ensure that the estimated human model matches the true human behavior in terms of the associated mean rewards. Then we have the following result.

Theorem 5.5.1. *Suppose the reward function $r(h^T, m^T)$ can be written as a sum of a decomposable function and a set $\mathcal{R}_p$ of path-based functions, and the remaining feature functions are either decomposable or path-based, i.e., $\mathcal{F} = \mathcal{F}_p \cup \mathcal{F}_d \cup \{r(h^T, m^T)\}$ and $\mathcal{G} = \mathcal{G}_p \cup \mathcal{G}_d$, where $\mathcal{F}_p$ and $\mathcal{G}_p$ denote path-based equality/inequality features, and $\mathcal{F}_d$ and $\mathcal{G}_d$ decomposable equality/inequality features, respectively. Suppose further that the initial machine's policy $\hat{Q}^{(0)}$ is uniformly random. Then the space complexity of each dual update of human estimation problem is $O\left((|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)T|\mathcal{H}||\mathcal{M}|\right)$, and the time complexity of each dual update is $O(T(|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)\max(T, |\mathcal{H}||\mathcal{M}|))$. The time and space complexity of the machine optimization problem are both $O((|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)T|\mathcal{H}||\mathcal{M}|)$.*

Remark: We envisage that the inclusion of path-based and decomposable feature and reward functions might allows a fairly rich framework to capture relevant interaction characteristics. In particular path-based features are capable of modeling detailed long-term memory in human-machine interactions while decomposable features can model short-term dependencies. As shown in Theorem 5.5.1, for such settings, the solution to (5.4) and (5.5) require steps with only polynomial space and time complexity.

Proof. Before proving Theorem 5.5.1, let us first consider a simpler case where only decomposable features are included in Problem (5.4). The following corollary to Theorem 5.4.1 characterizes a case where the complexity of the solution is polynomial in T.

Lemma 5.5.2. *Suppose the machine's policy is given by a (possibly time-inhomogeneous) one-step Markov process, i.e.,* $Q(m_t|h^{t-1}, m^{t-1}) = Q(m_t|h_{t-1}, m_{t-1})$, $\forall t, m^{t-1}, h^{t-1}$, *and all feature functions are decomposable, i.e.,* $\mathbf{f}(h^T, m^T) = \sum_{t=1}^{T} \mathbf{f}_t(h_t, m_t)$, *and* $\mathbf{g}(h^T, m^T) = \sum_{t=1}^{T} \mathbf{g}_t(h_t, m_t)$. *Then the solution to the human estimation problem is given, by the following procedure over a given dual* $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_f, \boldsymbol{\lambda}_g)$*:*

$$Z_{\boldsymbol{\lambda}}(h_t|m_t) = \begin{cases} e^{(\boldsymbol{\lambda}_f)^T \mathbf{f}_t(h_t,m_t) + (\boldsymbol{\lambda}_g)^T \mathbf{g}_t(h_t,m_t) + \sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t) \log Z_{\boldsymbol{\lambda}}(m_{t+1})} & t < T \\ e^{(\boldsymbol{\lambda}_f)^T \mathbf{f}_T(h_T,m_T) + (\boldsymbol{\lambda}_g)^T \mathbf{g}_T(h_T,m_T)} & t = T \end{cases},$$

where

$$Z_{\boldsymbol{\lambda}}(m_t) = \sum_{h_t} Z_{\boldsymbol{\lambda}}(h_t|m_t), \text{ and } P_{\boldsymbol{\lambda}}(h_t|m_t) = \frac{Z_{\boldsymbol{\lambda}}(h_t|m_t)}{Z_{\boldsymbol{\lambda}}(m_t)}.$$

Moreover, both the space and time complexity of establishing the distributional model is $O(T|\mathcal{H}||\mathcal{M}|)$. *The complexity of carrying out each dual update is* $O\left(T(|\mathcal{F}| + |\mathcal{G}|)|\mathcal{H}|^2|\mathcal{M}|^2\right)$

Proof. We'll prove that under such assumption, $Z_{\boldsymbol{\lambda}}(h_t|h^{t-1}, m^t)$ in Theorem 5.4.1 is given by:

$$\begin{aligned} Z_{\boldsymbol{\lambda}}(h_t|h^{t-1}, m^t) = {} & e^{(\boldsymbol{\lambda}_f)^T \sum_{\tau=1}^{t-1} \mathbf{f}_\tau(h_\tau,m_\tau) + (\boldsymbol{\lambda}_g)^T \sum_{\tau=1}^{t-1} \mathbf{g}_\tau(h_\tau,m_\tau)} \\ & \times e^{(\boldsymbol{\lambda}_f)^T \mathbf{f}_t(h_t,m_t) + (\boldsymbol{\lambda}_g)^T \mathbf{g}_t(h_t,m_t) + \sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t) \log Z_{\boldsymbol{\lambda}}(m_{t+1})}, \end{aligned}$$

where $Z_{\boldsymbol{\lambda}}(m_{t+1})$ is given as in Lemma 5.5.2.

The above equation implies that:

$$P_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t) = \frac{Z_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t)}{Z_{\boldsymbol{\lambda}}(h^{t-1},m^t)} = \frac{Z_{\boldsymbol{\lambda}}(h_t|m_t)}{Z_{\boldsymbol{\lambda}}(m_t)} = P_{\boldsymbol{\lambda}}(h_t|m_t),$$

and the Markov property follows.

For $t = T$, the identity holds true trivially. Now suppose it is true for $t+1$. Then according to Theorem 5.4.1, for $t < T$

$$\begin{aligned}
& Z_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t) \\
= \; & e^{\sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t)\log Z_{\boldsymbol{\lambda}}(h^t,m^{t+1})} \\
= \; & e^{\sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t)\log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|m_{t+1}) e^{(\boldsymbol{\lambda}_f)^T \sum_{\tau=1}^{t} \mathbf{f}_\tau(h_\tau,m_\tau)+(\boldsymbol{\lambda}_g)^T \sum_{\tau=1}^{t} \mathbf{g}_\tau(h_\tau,m_\tau)}} \\
= \; & e^{\sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t)\log e^{(\boldsymbol{\lambda}_f)^T \sum_{\tau=1}^{t} \mathbf{f}_\tau(h_\tau,m_\tau)+(\boldsymbol{\lambda}_g)^T \sum_{\tau=1}^{t} \mathbf{g}_\tau(h_\tau,m_\tau)}} \\
& \times e^{\sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t)\log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|m_{t+1})} \\
= \; & e^{(\boldsymbol{\lambda}_f)^T \sum_{\tau=1}^{t-1} \mathbf{f}_\tau(h_\tau,m_\tau)+(\boldsymbol{\lambda}_g)^T \sum_{\tau=1}^{t-1} \mathbf{g}_\tau(h_\tau,m_\tau)} \\
& \times e^{(\boldsymbol{\lambda}_f)^T \mathbf{f}_t(h_t,m_t)+(\boldsymbol{\lambda}_g)^T \mathbf{g}_t(h_t,m_t)+\sum_{m_{t+1}} Q(m_{t+1}|h_t,m_t)\log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|m_{t+1})}.
\end{aligned}$$

Then when we compute the ratio $\frac{Z_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t)}{Z_{\boldsymbol{\lambda}}(h^{t-1},m^t)}$ the term $e^{(\boldsymbol{\lambda}_f)^T \sum_{\tau=1}^{t-1} \mathbf{f}_\tau(h_\tau,m_\tau)+(\boldsymbol{\lambda}_g)^T \sum_{\tau=1}^{t-1} \mathbf{g}_\tau(h_\tau,m_\tau)}$ cancels out.

For the complexity, it's easy to see that in total we need to compute $T|\mathcal{H}||\mathcal{M}|$ probabilities. Thus the space complexity is $O(T|\mathcal{H}||\mathcal{M}|)$. If the vector multiplication is viewed as a basic operation, then computing each $Z_{\boldsymbol{\lambda}}(h_t|m_t)$ involves the sum of at most three vector inner product, and evaluating of its exponentiation. Therefore, the time complexity involved in establishing the distributional model is also $O(T|\mathcal{H}||\mathcal{M}|)$.

When computing the expectation of the feature functions, note that since all feature functions are decomposable, for all i:

$$E_{P_{\boldsymbol{\lambda}}Q}[f^i(H^T, M^T)] = \sum_{t=1}^{T} E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)],$$

and

$$E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)] = \sum_{m_t \in \mathcal{M}} P_{\boldsymbol{\lambda}}Q(m_t) \sum_{h_t \in \mathcal{H}} P_{\boldsymbol{\lambda}}(h_t|m_t) f_t(h_t, m_t).$$

Suppose we already obtained $P_{\boldsymbol{\lambda}}Q(m_{t-1})$, then

$$P_{\boldsymbol{\lambda}}Q(m_t) = \sum_{m_{t-1} \in \mathcal{M}} P_{\boldsymbol{\lambda}}Q(m_{t-1}) \sum_{h_{t-1} \in \mathcal{H}} P_{\boldsymbol{\lambda}}(h_{t-1}|m_{t-1}) Q(m_t|h_{t-1}, m_{t-1}).$$

Note that the marginal distribution of m_1: $P_{\boldsymbol{\lambda}}Q(m_1) = Q(m_1)$, which is already available. Thus we can compute $E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)]$ from $t = 1$ to $t = T$ and store $P_{\boldsymbol{\lambda}}Q(m_t), \forall 1 < t < T, m_t \in \mathcal{M}$. Then it is straightforward that computing $E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)]$ involves $|\mathcal{H}|^2|\mathcal{M}|^2$ operations, and computing $E_{P_{\boldsymbol{\lambda}}Q}[f^i(H^T, M^T)]$ is of complexity $T|\mathcal{H}|^2|\mathcal{M}|^2$. Each dual update involves evaluation of $E_{P_{\boldsymbol{\lambda}}Q}[f^i(H^T, M^T)],\ \forall\ i \in \mathcal{F}$, and $E_{P_{\boldsymbol{\lambda}}Q}[g^i(H^T, M^T)],\ \forall\ i \in \mathcal{G}$, thus has the complexity of $O(T(F+G)|\mathcal{H}|^2|\mathcal{M}|^2)$. □

Now let us assume that the equality and inequality constraint sets can be each partitioned into two subsets: $\mathcal{F} = \mathcal{F}_p \cup \mathcal{F}_d$, and $\mathcal{G} = \mathcal{G}_p \cup \mathcal{G}_d$, where $\mathcal{F}_d$ and $\mathcal{G}_d$ correspond to the decomposable features and $\mathcal{F}_p$ and $\mathcal{G}_p$ correspond to the path-based features. Moreover, the path-based features are:

$$f^i(h^T, m^T) = c_i \mathbf{1}_{\{(h^T, m^T) = (\bar{h}^{i,T}, \bar{m}^{i,T})\}},\ i \in \mathcal{F}_p, \text{ and}$$
$$g^i(h^T, m^T) = c_i \mathbf{1}_{\{(h^T, m^T) = (\bar{h}^{i,T}, \bar{m}^{i,T})\}},\ i \in \mathcal{G}_p,$$

while decomposable features are:

$$f^i(h^T, m^T) = \sum_{t=1}^{T} f_t^i(h_t, m_t),\ i \in \mathcal{F}_d,\ \text{and}\ \ g^i(h^T, m^T) = \sum_{t=1}^{T} g_t^i(h_t, m_t),\ i \in \mathcal{G}_d.$$

Also, the reward function is given by its path-based part $r^{i,p}(h^T, m^T) = c_i \mathbf{1}_{\{(h^T, m^T) = (\bar{h}^{i,T}, \bar{m}^{i,T})\}},\ i \in \mathcal{R}_p,$, together with a decomposable part $r^d(h^T, m^T) = \sum_{t=1}^{T} r_t^d(h_t, m_t)$, giving

$$r(h^T, m^T) = \sum_{i \in \mathcal{R}_p} r^{i,p}(h^T, m^T) + \sum_{t=1}^{T} r_t^d(h_t, m_t).$$

First let us consider the human estimation problem. Note that if the conditioning sequence is not a prefix of any path-based feature function (including functions in $\mathcal{R}_p$), the backward recursion in Theorem 5.4.1 is equivalent to the case where we only have decomposable feature functions.

Without loss of generality, consider decomposable feature functions given by:

$$\mathbf{f}^d(h^T, m^T) = \sum_{t=1}^{T} \mathbf{f}_t^d(h_\tau, m_\tau),$$

and

$$\mathbf{g}^d(h^T, m^T) = \sum_{t=1}^{T} \mathbf{g}_t^d(h_\tau, m_\tau).$$

We shall let $\boldsymbol{\lambda}_f^d$ be the dual variable corresponding to the decomposable equality constraints, $\boldsymbol{\lambda}_g^d$ that corresponding to the decomposable inequality constraints, and λ_r that corresponding to the reward function. Note that when we establish the distributional model, functions in $\mathcal{R}_p$ together with $r^d(h^T, m^T)$ can be regarded as individual 'feature' functions, which share the same dual

variable λ_r. It follows from Lemma 5.5.2 that if $(h^t, m^t) \neq (\bar{h}^{i,t}, \bar{m}^{i,t})$, $\forall i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p$, we have

$$Z_{\boldsymbol{\lambda}}(h_t|h^{t-1}, m^t) = Z_{\boldsymbol{\lambda}}(h_t|m_t) e^{(\boldsymbol{\lambda}_f^d)^T \sum_{\tau=1}^{t-1} \mathbf{f}_\tau^d(h_\tau, m_\tau) + (\boldsymbol{\lambda}_g^d)^T \sum_{\tau=1}^{t-1} \mathbf{g}_\tau^d(h_\tau, m_\tau) + \lambda_r \sum_{\tau=1}^{t-1} r_\tau^d(h_\tau, m_\tau)},$$

where $Z_{\boldsymbol{\lambda}}(h_t|m_t)$ is given by the recursion specified in Lemma 5.5.2, with $r^d(h^T, m^T)$ as a feature function. Let us denote the set of machine actions at time t that stay on at least one path-based feature function's support by $\mathcal{M}_t^p(h^{t-1}, m^{t-1}) = \{m_t | \exists i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p \ s.t. \ h^{t-1} = \bar{h}^{i,t-1}, m^t = \bar{m}^{i,t}\}$ and $\mathcal{H}_t^p(h^{t-1}, m^t) = \{h_t | \exists i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p \ s.t. \ h^t = \bar{h}^{i,t}, m^t = \bar{m}^{i,t}\}$. For $\bar{h}^{i,t}, \bar{m}^{i,t}$, the backward recursion in Theorem 5.4.1 becomes following:

$$\begin{aligned}
& Z_{\boldsymbol{\lambda}}(\bar{h}_{i,t} | \bar{h}^{i,t-1}, \bar{m}^{i,t}) \\
= \ & e^{\sum_{m_{t+1}} Q(m_{t+1}|\bar{h}^{i,t}, \bar{m}^{i,t}) \log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1}))} \\
= \ & \exp\Bigg(\underbrace{\sum_{m_{t+1} \in \mathcal{M}_{t+1}^p(\bar{h}^{i,t}, \bar{m}^{i,t})} Q(m_{t+1}|\bar{h}^{i,t}, \bar{m}^{i,t}) \log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}\left(h_{t+1}|\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1})\right)}_{:=A} \\
& + \underbrace{\sum_{m_{t+1} \notin \mathcal{M}_{t+1}^p(\bar{h}^{i,t}, \bar{m}^{i,t})} Q(m_{t+1}|\bar{h}^{i,t}, \bar{m}^{i,t}) \log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}\left(h_{t+1}|\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1})\right)}_{:=B} \Bigg) \\
= \ & \exp(A + B). \qquad (5.13)
\end{aligned}$$

From the result of Lemma 5.5.2,

$$A = \sum_{\substack{m_{t+1} \\ \in \mathcal{M}^p_{t+1}(\bar{h}^{i,t}, \bar{m}^{i,t})}} Q(m_{t+1}|\bar{h}^{i,t}, \bar{m}^{i,t}) \log \Bigg(\sum_{\substack{h_{t+1} \\ \in \mathcal{H}^p_{t+1}(\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1}))}} Z_{\boldsymbol{\lambda}}(h_{t+1}|\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1}))$$
$$+ \sum_{h_{t+1} \notin \mathcal{H}^p_t(\bar{h}^{i,t}, (\bar{m}^{i,t}, m_{t+1}))} Z_{\boldsymbol{\lambda}}(h_{t+1}|m_{t+1}) \exp \Bigg((\boldsymbol{\lambda}^d_f)^T \sum_{\tau=1}^{t} \mathbf{f}^d_\tau(\bar{h}_{i,\tau}, \bar{m}_{i,\tau})$$
$$+ (\boldsymbol{\lambda}^d_g)^T \sum_{\tau=1}^{t} \mathbf{g}^d_\tau(\bar{h}_{i,\tau}, \bar{m}_{i,\tau}) + \lambda_r \sum_{\tau=1}^{t} r^d_\tau(\bar{h}^i_\tau, \bar{m}^i_\tau) \Bigg) \Bigg).$$

$$B = \sum_{m_{t+1} \notin \mathcal{M}^p_{t+1}(\bar{h}^{i,t}, \bar{m}^{i,t})} Q(m_{t+1}|\bar{h}^{i,t}, \bar{m}^{i,t})$$
$$\times \log \sum_{h_{t+1}} Z_{\boldsymbol{\lambda}}(h_{t+1}|m_{t+1}) e^{(\boldsymbol{\lambda}^d_f)^T \sum_{\tau=1}^{t} \mathbf{f}^d_\tau(\bar{h}_{i,\tau}, \bar{m}_{i,\tau}) + (\boldsymbol{\lambda}^d_g)^T \sum_{\tau=1}^{t} \mathbf{g}^d_\tau(\bar{h}_{i,\tau}, \bar{m}_{i,\tau}) + \lambda_r \sum_{\tau=1}^{t} r^d_\tau(\bar{h}_{i,\tau}, \bar{m}_{i,\tau})}.$$

Note that B solely depends on the result of the case where there are only decomposable features. The additional complexity introduced is in the computation of A which is determined by the number of nonzero path-based features after current step. The key insight is that we only need to track A for a prefix where there is at least one nonzero path-based feature function, and the set of possible choices of such prefixes forms a tree where the number of leaf nodes is at most $|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|$. Then at each t, we need to compute A for at most $|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|$ conditioning prefixes. Thus the complexity of obtaining the whole distributional model is $O((|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)T|\mathcal{H}||\mathcal{M}|)$.

When computing the mean of a feature function $E_{P_{\boldsymbol{\lambda}}Q}[f^i(H^T, M^T)]$, we have two different cases:

1. If $f^i(h^T, m^T)$ is a path-based feature with support $\bar{h}^{i,T}, \bar{m}^{i,T}$. Then $E_{P_{\boldsymbol{\lambda}}Q}[f^i(H^T, M^T)] = P_{\boldsymbol{\lambda}}Q(\bar{h}^T, \bar{m}^T) = c_i \prod_{t=1}^{T} Q(\bar{m}_t^i|\bar{h}^{i,t-1}, \bar{m}^{i,t-1})P(\bar{h}_t^i|\bar{h}^{i,t-1}, \bar{m}^{i,t})$. This requires at most T multiplications.

2. If $f^i(h^T, m^T)$ is a decomposable feature then the associated moment can be written as

$$E_{P_{\boldsymbol{\lambda}}Q}[f(H^T, M^T)] = \sum_{t=1}^{T} E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)].$$

Let us define a stopping time T_D w.r.t. (H^T, M^T) such that $T_D := \min \{t \mid 1 \leq t \leq T, (H^t, M^t) \neq (\bar{h}^{i,t}, \bar{m}^{i,t}),\ \forall i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p\}$. That is, T_D is the first time when the realization of interaction deviates from supports of all path-based feature functions. Then based on the value of T_D, we can partition $E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)]$ as follows:

$$\begin{aligned} E_{P_{\boldsymbol{\lambda}}Q}[f_t(H_t, M_t)] &= P_{\boldsymbol{\lambda}}Q(T_D \leq t)E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)|T_D \leq t] \\ &+P_{\boldsymbol{\lambda}}Q(T_D > t)E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)|T_D > t]. \end{aligned} \tag{5.14}$$

After deviating from all supports, i.e., when $T_D \leq t$, the distribution is the same as the case where only decomposable features functions have been included. Thus $E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)|T_D \leq t]$ can be easily obtained within $O(T|\mathcal{H}||\mathcal{M}|)$ computations, by taking advantage of the one-step Markov property. Also, the distribution of the stopping time T_D can be

computed as:

$$
\begin{aligned}
P_{\boldsymbol{\lambda}}Q(T_D \leq t) &= 1 - P_{\boldsymbol{\lambda}}Q(T_D > t) \\
&= 1 - \sum_{i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p} P_{\boldsymbol{\lambda}}Q(H^t = \bar{h}^{i,t}, M^t = \bar{m}^{i,t}) \\
&= 1 - \sum_{i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p} \prod_{\tau=1}^{t} P_{\boldsymbol{\lambda}}(\bar{h}_{i,\tau} | \bar{h}^{i,\tau-1}, \bar{m}^{i,\tau}) Q(\bar{m}_{i,\tau} | \bar{h}^{i,\tau-1}, \bar{m}^{i,\tau-1}).
\end{aligned}
$$

At most it requires $T(|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)$ computations. Same computation complexity is expected when computing $P_{\boldsymbol{\lambda}}Q(T_D > t)$. For $E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)|T_D > t]$, we have:

$$
\begin{aligned}
&E_{P_{\boldsymbol{\lambda}}Q}[f_t^i(H_t, M_t)|T_D > t] \\
&\quad = \sum_{i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p} f_t^i(\bar{h}_{i,t}, \bar{m}_{i,t}) P_{\boldsymbol{\lambda}}Q(H_t = \bar{h}_{i,t}, M_t = \bar{m}_{i,t} | T_D > t) \\
&\quad = \sum_{i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p} f_t^i(\bar{h}_{i,t}, \bar{m}_{i,t}) \frac{P_{\boldsymbol{\lambda}}Q(H^t = \bar{h}_i^t, M^t = \bar{m}_i^t)}{P_{\boldsymbol{\lambda}}Q(T_D > t)}.
\end{aligned}
$$

It's easy to observe that it requires $O(T(|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|))$ computations, too. Then the computation complexity to compute the sum is $O(T^2(|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|))$.

Exactly the same complexity is obtained when computing functions in $\mathcal{G}$, $\mathcal{R}_p$ and $r^d(h^T, m^T)$. Then the time complexity of one dual update will be given by the maximum of the two cases, as well as the the time to establish the distributional model, thus is given by $O(T(|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|) \max(T, |\mathcal{H}||\mathcal{M}|))$.

For the machine optimization problem, we do not need to carry out the dual update, since γ is fixed throughout the iterations. Therefore we only

need to establish the distributional model $\hat{Q}$. By viewing the path-based part of the reward function as the path-based 'feature' in the machine optimization problem, we can easily conclude that both the space and time complexity in obtaining the machine's policy $\hat{Q}$ is $O((|\mathcal{F}_p| + |\mathcal{G}_p| + |\mathcal{R}_p|)T|\mathcal{H}||\mathcal{M}|)$. Moreover, as long as the initial machine policy is such that after deviating from the union of the supports of all path-based feature functions, it is one-step Markov, i.e. $\hat{Q}^{(0)}(m_t|m^{t-1}, h^{t-1}) = \hat{Q}^{(0)}(m_t|m_{t-1}, h_{t-1})$ when $(m^{t-1}, h^{t-1}) \neq (\bar{m}_i^{t-1}, \bar{h}_i^{t-1}),\ \forall i \in \mathcal{F}_p \cup \mathcal{G}_p \cup \mathcal{R}_p$, all the assumptions introduced in Theorem 5.5.1 are satisfied throughout the AREA iterations. A uniform random $\hat{Q}^{(0)}$ is a special case satisfying that condition.

Therefore the complexity of AREA algorithm is polynomial in T as long as the number of dual updates is limited in the human estimation problem. □

5.6 AREA Convergence

As discussed in previous sections, the AREA Algorithm is aimed at achieving high rewards through consistency in the estimated human model and optimized machine policy. In this section, we characterize AREA's convergence properties.

The convergence of the algorithm can be guaranteed in two extremal cases. Clearly if the set of feature functions $\mathcal{F}$ and $\mathcal{G}$ is rich enough that the true human behaviour is recovered as the solution to (5.4), then AREA converges. Or, if the feature set is sufficient to guarantee that the actual human behaviour along the 'paths' that are impactive to reward is perfectly captured by the estimated human model, then AREA also converges in one iteration. Following two theorems address those two extremal cases, respectively.

Convergence of AREA under sufficient statistics: An implication of Theorem 5.4.1 is that under our maximum entropy framework estimated human models will be of the form given in the theorem for a given value of $\boldsymbol{\lambda}$. We refer to such distributions as causally conditioned Gibbs distributions formally defined as follows:

Definition 5.6.1. Given the set of constraints $\mathcal{F}$, $\mathcal{G}$ and underlying machine policy $Q(m^T \| h^T)$, we define the associated *causally conditioned Gibbs distri-*

butions as

$$\mathcal{P}_g(Q,\mathcal{F},\mathcal{G}) := \{\ P(h^T\|m^T) \mid \exists \boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{F}|} \times \mathbb{R}_-^{|\mathcal{G}|} \\ s.t.\ P(h_t|h^{t-1},m^t) = P_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t) \text{ for } t = 1,\ldots.T\ \}, \tag{5.15}$$

where $P_{\boldsymbol{\lambda}}(h_t|h^{t-1},m^t)$ is as given in Theorem 5.4.1. That is, each element in $\mathcal{P}_g(Q,\mathcal{F},\mathcal{G})$ is a causally conditioned distribution of the form given in Theorem 5.4.1 for $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_f, \boldsymbol{\lambda}_g)$.

Remark: According to *Hammersley-Clifford Theorem* in [85], if the human behavior $P^*(h^T\|m^T)$ has full support, i.e., there is no $(h^T, m^T) \in \mathcal{H}^T \times \mathcal{M}^T$ such that $P^*(h^T\|m^T) = 0$, and machine's policy $Q(m^T\|h^T)$ also has full support, then there exists a pair of finite sets of constraint $\mathcal{F}^*$ and $\mathcal{G}^*$ such that the true human behavior is in the associated causal Gibbs distribution set, such that, $P^*(h^T\|m^T) \in \mathcal{P}_g(Q,\mathcal{F}^*,\mathcal{G}^*)$.

Then if the features in human estimation problem are rich enough, the following theorem captures the convergence of AREA.

Theorem 5.6.1. *If the feature sets $\mathcal{F}$ and $\mathcal{G}$ and initial machine policy $\hat{Q}^{(0)}$ are such that,*

$$P^*(h^T\|m^T) \in \mathcal{P}_g(\hat{Q}^{(0)},\mathcal{F},\mathcal{G}) \cap \mathcal{P}_g(m^*(m(P^*)),\mathcal{F},\mathcal{G})$$

AREA algorithm converges after the first iteration.

Proof. When $P^*(h^T\|m^T) \in \mathcal{P}_g(\hat{Q}^{(0)},\mathcal{F},\mathcal{G})$, then $P^*(h^T\|m^T)$ can be parameterized for some $\boldsymbol{\lambda}^* := (\boldsymbol{\lambda}_f^*, \boldsymbol{\lambda}_g^*)$ and will be the solution to the human estimation

problem, based on the data produced under machine policy $\hat{Q}^{(0)}$. Now, given $\hat{P}^{(0)} = P^*$, the machine optimization problem generates $\hat{Q}^{(1)} = m^*(m(P^*))$. However since we have that $P^*(h^T \| m^T) \in \mathcal{P}_g(m^*(m(P^*)), \mathcal{F}, \mathcal{G})$, again we can ensure that $\hat{P}^{(1)} = P^*$. By induction it is easy to see that $\hat{P}^{(n)} = P^*,\ \forall n \geq 0$ and $\hat{Q}^{(n)} = m^*(m(P^*)),\ \forall n \geq 1$. Thus AREA iterations will converge after the first iteration. □

Convergence of AREA when human behavior is perfectly estimated along 'rewarding' path:

Theorem 5.6.2. *Suppose the reward function is a path-based, i.e.,* $r(h^T, m^T) = \mathbf{1}_{\{(h^T, m^T) = (\bar{h}^{r,T}, \bar{m}^{r,T})\}}$ *and included in the feature set* $\mathcal{F}$ *and the initial machine's policy* $\hat{Q}^{(0)}$ *has full support. Consider a modified version of human estimation problem which includes the following additional features. For each path-based feature, i.e.,* $i \in \mathcal{F}_p$*, we include* $T-1$ *auxiliary features* $\mathcal{F}_p^i$ *as follows:*

$$\mathcal{F}_p^i = \{f^{i,t}(h^T, m^T) \mid f^{i,t}(h^T, m^T) = \mathbf{1}_{\{(h^t, m^t) = (\bar{h}^{i,t}, \bar{m}^{i,t})\}},\ \textit{for } t = 1, \ldots, T\},$$

ensuring matching of full-length and prefixes for the path based features. For the modified set of equality features $\mathcal{F} = \mathcal{F}_d \cup \left(\bigcup_{i \in \mathcal{F}_p} \mathcal{F}_p^i \right)$*. and an arbitrary set of inequality features* $\mathcal{G}$ *AREA converges in one iteration.*

Proof. At the nth iteration, when matching the moment of path-based features, we have:

$$\hat{P}^{(n)} \hat{Q}^{(n-1)}(\bar{h}^{i,T}, \bar{m}^{i,T}) = P^* \hat{Q}^{(n-1)}(\bar{h}^{i,T}, \bar{m}^{i,T}),\ \forall\ i \in \mathcal{F}_{1,p}.$$

After cancelling out $Q^{(n-1)}(\bar{m}^{i,T}\|\bar{h}^{i,T})$ on both sides we have

$$\hat{P}^{(n)}(\bar{h}^{i,T}\|\bar{m}^{i,T}) = P^*(\bar{h}^{i,T}\|\bar{m}^{i,T}).$$

If the feature set in problem (5.4) also includes $\mathcal{F}_p^i$, for all $i \in \mathcal{F}_p$, by a similar argument we have that for all $i \in \mathcal{F}_p$ and $t = 1, 2, \ldots, T$,

$$\prod_{\tau=1}^{t} P^{(n)}(\bar{h}_{i,t}|\bar{h}^{i,t-1}, \bar{m}^{i,t}) = \prod_{\tau=1}^{t} P^*(\bar{h}_{i,t}|\bar{h}^{i,t-1}, \bar{m}^{i,t}).$$

Thus $\hat{P}^{(n)}(\bar{h}_{i,t}|\bar{h}^{i,t-1}, \bar{m}^{i,t}) = P^*(\bar{h}_{i,t}|\bar{h}^{i,t-1}, \bar{m}^{i,t})$ for all $i \in \mathcal{F}_p$ and $1 \leq t \leq T$.

When the reward function is a path-based function, a straightforward observation from Theorem 5.4.1 is that, the resulting machine policy $\hat{Q}(m_t|h^{t-1}, m^{t-1})$ is uniformly random if $(h^{t-1}, m^{t-1}) \neq (\bar{h}^{r,t-1}, \bar{m}^{r,t-1})$. The machine policy along the support of the reward function is induced by:

$$\begin{aligned} Y_\gamma(\bar{m}_{r,t}|\bar{h}^{r,t-1}, \bar{m}^{r,t-1}) &= e^{\sum_{h_t} \hat{P}^{(n-1)}(h_t|\bar{h}^{r,t-1}, \bar{m}^{r,t}) \log Y_\gamma((\bar{h}^{r,t-1}, h_t), \bar{m}^{r,t})} \\ &= e^{(1-P^*(\bar{h}_{r,t}|\bar{h}^{r,t-1}, \bar{m}^{r,t})) \log Y_\gamma^t + P^*(\bar{h}_{r,t}|\bar{h}^{r,t-1}, \bar{m}^{r,t}) \log Y_\gamma(\bar{h}^{r,t}, \bar{m}^{r,t})}, \end{aligned} \quad (5.16)$$

where $Y_\gamma^t := Y_\gamma(h^t, m^t)$ for $(h^t, m^t) \neq (\bar{h}^{r,t}, \bar{m}^{r,t})$. The sequence of interactions can be suppressed because from Theorem 5.4.1 we can conclude that, after leaving the 'profitable' path all Y_γ will be the same, independent of corresponding $P(h^T\|m^T)$. From Eq. (5.16) we can prove by induction that $Y_\gamma(\bar{m}_{r,t}|\bar{h}^{r,t-1}, \bar{m}^{r,t-1})$ does not change after the first iteration. Thus the resulted machine's policy $\{\hat{Q}^{(n)}\}$ converges after the first iteration. □

For more general cases, the convergence of AREA algorithm is subtle. Note that the human estimation problem (5.4) depends on the machine policy

$Q(m^T \| h^T)$ used. Thus given $Q(m^T \| h^T)$ at the current iteration one can determine the associated model for human behavior $h^*(Q, \mathbf{c}_f, \mathbf{c}_g)$ which may in turn change the optimal machine policy. This makes the analysis of convergence difficult. In order to facilitate the convergence, we propose introducing an additional inequality constraint to the human estimation problem (5.4).

During the nth iteration, given the previously obtained $\hat{P}^{(n-1)}$ and $\hat{Q}^{(n)}$ we shall include the following *step-dependent* inequality constraint in $\mathcal{G}$. Let $g^{0,(n)}(h^T, m^T) = -\log \hat{Q}^{(n)}(m^T \| h^T) + \gamma r(h^T, m^T)$, and let $c_g^{0,(n)} = E_{\hat{P}^{(n-1)}\hat{Q}^{(n)}}[g^{0,(n)}(H^T, M^T)]$, then on AREA iteration n we require that $E_{P\hat{Q}^{(n)}}[g^{0,(n)}(H^T, M^T)] \geq c_g^{0,(n)}$.

Let us define a sequence $\{L^{(n)}\}$ of entropy regularized expected rewards across iterations, i.e., $L^{(n)} := E_{\hat{P}^{(n)}\hat{Q}^{(n)}}[g^{0,(n)}(H^T, M^T)]$. Then we have the following result.

Theorem 5.6.3. *Consider the AREA algorithm optimizing a human-machine interactive process with a fixed sets of equality/inequality constraints $\mathcal{F}$ and $\mathcal{G}$. Suppose $\mathcal{G}$ is modified to $\mathcal{G}^{(n)}$ by adding the additional step-dependent inequality constraint $E_{P\hat{Q}^{(n)}}[g^{0,(n)}(H^T, M^T)] \geq c_g^{0,(n)}$. Then the modified AREA iterations generate a bounded nondecreasing sequence $\{L^{(n)}\}$, which must converge.*

Remark: Note that when the conditions in Theorem 5.5.1 holds true, then $\hat{Q}^{(n)}$ takes independent actions once the path deviates from the support of all path-based feature functions. Thus the introduced step-dependent feature function can be written as: $g^{0,(n)}(h^T, m^T) = -\sum_{t=1}^{T} \log \hat{Q}_t^{(n)}(m_t) +$

$\sum_{i\in\mathcal{F}_p\cup\mathcal{G}_p}\left(\sum_{t=1}^{T}\log\hat{Q}_t^{(n)}(\bar{m}_t^i)-\log\hat{Q}^{(n)}(\bar{m}^{i,T}\|\bar{h}^{i,T})\right)\mathbf{1}_{\{h^T=\bar{h}^{i,T},m^T=\bar{m}^{i,T}\}}$, which is still a weighted sum of path-based functions and decomposable functions. This in turn means that the added constraint is such that iteration steps will still have the polynomial complexity shown in Theorem 5.5.1.

$L^{(n)}$ can be regarded as a measure of the performance of the associated machine policy $\hat{Q}^{(n)}$. Indeed if $\hat{Q}^{(n)}$ were a fixed point of AREA recursion, then the optimal objective function of (5.5) would have converged to $L^{(n)}$. Also note that by further assuming that the feature and reward functions are decomposable, we can characterize the performance for the converging sequence $\{L^{(n)}\}$–see Section 5.7.

The solution at nth human estimation step can be written as $\hat{P}^{(n)}=h^*(\hat{Q}^{(n)},\mathbf{c}_f(\hat{Q}^{(n)}),\mathbf{c}_g(\hat{Q}^{(n)},\hat{P}^{(n-1)}))$. Indeed $\mathbf{c}_f(\hat{Q}^{(n)})=E_{P^*\hat{Q}^{(n)}}[\mathbf{f}(H^T,M^T)]$ depends on the true human behavior P^*, the feature set $\mathcal{F}$ and also the machine policy in use $\hat{Q}^{(n)}$. However, throughout AREA iterations, P^* and $\mathcal{F}$ are fixed. Thus for simplicity we write $\mathbf{c}_f$ as a function of $\hat{Q}^{(n)}$. Similarly, we write $\mathbf{c}_g$ as a function of $\hat{Q}^{(n)}$ and $\hat{P}^{(n-1)}$, where the only dependency on $\hat{P}^{(n-1)}$ is through the step dependent feature $c_g^{0,(n)}$ we have introduced. Moreover, a direct result of Lemma 2 in [76] showed that $\mathbf{c}_g(\hat{Q}^{(n)},\hat{P}^{(n-1)})$ is actually a function of $Y_\gamma^{(n)}$, which is the Y_γ associated with $\hat{Q}^{(n)}$ as defined in Theorem 5.4.4.

Lemma 5.6.4. *During the nth iteration of AREA, let us denote the Y_γ in the machine optimization problem by $Y_\gamma^{(n)}$. Then*

$$c_g^{0,(n)}=E_{\hat{P}^{(n-1)}\hat{Q}^{(n)}}\left[-\log\hat{Q}^{(n)}(M^T\|H^T)+\gamma r(H^T,M^T)\right]=\log\sum_{m_1\in\mathcal{M}}Y_\gamma^{(n)}(m_1)$$

Proof. This is just a special case of Lemma 2 in [76]. By plugging the recursive form defined in Theorem 5.4.4 we can prove it is true. □

Therefore $\hat{P}^{(n)}$ is actually a function of $Y_\gamma^{(n)}$, because $\hat{Q}^{(n)}$ is naturally a function of $Y_\gamma^{(n)}$ by Theorem 5.4.4, and $\mathbf{c}_g$ is independent of $\hat{P}^{(n-1)}$ given $Y_\gamma^{(n)}$ by Lemma 5.6.4:

$$\hat{P}^{(n)} = h^*(\hat{Q}^{(n)}, \mathbf{c}_f(\hat{Q}^{(n)}), \mathbf{c}_g(\hat{Q}^{(n)}, \hat{P}^{(n-1)})) = h^*(m^*(Y_\gamma^{(n)}), \mathbf{c}_f(m^*(Y_\gamma^{(n)})), \mathbf{c}_g(Y_\gamma^{(n)})).$$

In order to show convergence it will be easier to study it in terms of the underlying variables $Y_\gamma^{(n)}$. In the sequel when there is no ambiguity we will denote it by $\hat{P}^{(n)} = h^*(Y_\gamma^{(n)})$.

Let us define the following function of Y_γ:

$$L(Y_\gamma) := \mathbb{H}_{h^*(Y_\gamma)m^*(Y_\gamma)}(M^T \| H^T) + \gamma E_{h^*(Y_\gamma)m^*(Y_\gamma)}[r(H^T, M^T)]. \tag{5.17}$$

Note that $L^{(n)} = L(Y_\gamma^{(n)})$. Now we are ready to prove Theorem 5.6.3.

Proof. In order to show the convergence of $\{L(Y_\gamma^{(n)})\}$, we define the following functions of Y_γ:

1. $c(Y_\gamma|Y'_\gamma)$ is the objective function of the machine's optimization problem, where Y_γ and Y'_γ are as defined in Theorem 5.4.4 and are associated with $Q(m^T\|h^T)$ and previous machine's policy $Q'(m^T\|h^T)$,

$$c(Y_\gamma|Y'_\gamma) := \mathbb{H}_{h(Y'_\gamma)m(Y_\gamma)}(M^T \| H^T) + \gamma E_{h(Y'_\gamma)m(Y)}[r(H^T, M^T)]. \tag{5.18}$$

2. $L(Y_\gamma)$ is defined as

$$\begin{aligned} L(Y_\gamma) &:= c(Y_\gamma|Y_\gamma) \\ &= \mathbb{H}_{h(Y_\gamma)m(Y_\gamma)}(M^T\|H^T) + \gamma E_{h(Y_\gamma)m(Y_\gamma)}[r(H^T, M^T)], \end{aligned} \quad (5.19)$$

Consider the variables associated at nth iteration to be $Y_\gamma^{(n)}$. During the AREA algorithm there are two possible cases: (1) $\hat{Q}^{(n+1)} = \hat{Q}^{(n)}$, and (2) $\hat{Q}^{(n+1)} \neq \hat{Q}^{(n)}$. In case (1) it's straightforward that $\hat{Q}^{(m)}$ will be the same as $\hat{Q}^{(n)}$, for all $m \geq n$. In case (2) we can show the convergence by proving the strict monotonicity of $\{L(\hat{Y}_\gamma^{(n)})\}$ as follows.

$$\begin{aligned} L(Y_\gamma^{(n+1)}) &\geq c(Y_\gamma^{(n+1)}|Y_\gamma^{(n)}) && (5.20) \\ &\geq c(Y_\gamma^{(n)}|Y_\gamma^{(n)}) && (5.21) \\ &= L(Y_\gamma^{(n)}) && (5.22) \end{aligned}$$

Here Eq. (5.21) follows from the optimality of the solution to the machine's optimization Eq. (5.5), and Eq. (5.22) follows by the definition of $L(Y_\gamma^{(n)})$. Thus we only need to show Eq. (5.20). Based on the definitions of the associated quantities, we have for all $Y_\gamma^{(n+1)}$:

$$\begin{aligned} &L(Y_\gamma^{(n+1)}) - c(Y_\gamma^{(n+1)}|Y_\gamma^{(n)}) \\ &\quad = E_{h(Y_\gamma^{(n+1)})m(Y_\gamma^{(n+1)})}\left[-\log Q^{(n+1)}(M^T\|H^T) + \gamma r(H^T, M^T)\right] - g_0(Y_\gamma^{(n+1)}) \\ &\quad \geq 0. \end{aligned}$$

The inequality holds true because in the human estimation problem, we introduced constraint $E_{P\hat{Q}^{(n)}}[g^{0,(n)}(H^T, M^T)] \geq c_g^{0,(n)}$. Also, due to the boundedness of both $\mathbb{H}_{h(Y_\gamma)m(Y_\gamma)}(M^T\|H^T)$ and the expected reward function, $L(Y_\gamma)$

is also upper bounded. Therefore, the sequence generated by AREA recursion $\{L(Y_\gamma^{(n)})\}$ converges monotonically. □

An interesting observation we can make is that, $\{L(Y_\gamma^{(n)})\}$ converges to a value associated with a fixed point of AREA iterations.

Theorem 5.6.5. *$\{L(Y_\gamma^{(n)})\}$ converges to L^∞, and there exists a Y_γ^∞ such that $L(Y_\gamma^\infty) = L^\infty$, and Y_γ^∞ is a fixed point of AREA iterations, i.e., $m^*(m(h^*(Y_\gamma^\infty))) = m^*(Y_\gamma^\infty)$.*

Proof. Now if we let the AREA algorithm stops once we observe $\hat{Q}^{(n+1)} = \hat{Q}^{(n)}$, otherwise proceed to the next iteration, then throughout the iterations of AREA (except for the last step when we stop), machine optimization problem is strongly concave, thus obtain a unique maximum at any $n+1$st step, which is $\hat{Q}^{(n+1)} \neq \hat{Q}^{(n)}$. Therefore, Eq. (5.21) holds true strictly. Then $L(Y_\gamma^{(n+1)}) > L(Y_\gamma^{(n)})$ in case (2). We can follow the result in [86], by defining the solution set as the set of Y_γ such that $m^*(m(h^*(Y_\gamma))) = m^*(Y_\gamma)$, i.e., the set of fixed point of AREA iterations, Corollary 1-1 in [86] shows that one of the following statement is true:

1. The iteration stops in finite steps. Then we know it corresponds to the case where we have for some n, $\hat{Q}^{(n+1)} = \hat{Q}^{(n)}$. Thus $\forall\ m > n$, $\hat{Q}^{(m)} = \hat{Q}^{(n)}$, implying $\{\hat{Q}^{(n)}\}$ converges.

2. The iteration does not stop. Then according to Corollary 1-1 in [86], any convergent subsequence of $\{Y_\gamma^{(n)}\}$, say $\{\hat{Y}_\gamma^{(k)} : k \in \mathcal{K}_j \subseteq \mathbb{Z}^+\}$ converges

to an accumulation point $\hat{Y}_\gamma^{(\infty),j}$ as $k \to \infty$, such that $\hat{Y}_\gamma^{(\infty),j}$ is within the solution set.

Therefore, due to the convergence of $\{L(Y_\gamma^{(n)})\}$, all the accumulation points of $\{Y_\gamma^{(n)}\}$ have the same value of $L(Y_\gamma)$ function, and are fixed points of AREA iterations. □

5.7 One Important Special Case: Decomposable Features

In this section we discuss AREA under a special family of features. Specifically, we will derive performance guarantees for the case where the solution has a special structure.

From now on we shall make the following assumption.

Assumption 4. *Reward function $r(h^T, m^T)$ is also used as a feature function in the estimation phase. Also $\forall i \in \mathcal{F}$, $f_i(h^T, m^T)$ is decomposable, including the reward function $r(h^T, m^T)$.*

Then following statements are true.

Lemma 5.7.1. *Under Assumption 4, the solution to the machine's optimization phase has no dependency across time t. That is, at the nth iteration:*

$$\hat{Q}^{(n)}(m_t|h^{t-1}, m^{t-1}) = \hat{Q}^{(n)}(m_t).$$

Moreover,

$$\hat{Q}^{(n)}(m_t) \propto e^{\gamma E_{\hat{P}^{(n-1)}\hat{Q}^{(n)}}[r(H_t, m_t)]}.$$

Note that under such assumptions, $E_{\hat{P}^{(n-1)}\hat{Q}^{(n)}}[r(H_t, m_t)]$ only depends on $\hat{P}^{(n-1)}$.

Proof. This can be proved in a manner similar to Lemma 5.5.2. Specifically, we can show that when Assumption 4 is true,

$$Y_\gamma(m_t|m^{t-1}, h^{t-1}) = \left(\prod_{\tau=t+1}^{T} Y_{\gamma,\tau}\right) e^{\gamma \sum_{\tau=1}^{t-1} r_\tau(h_\tau, m_\tau)} e^{\gamma \sum_{h_t} \hat{P}^{(n-1)}(h_t|m_t) r_t(h_t, m_t)}, \tag{5.23}$$

where $Y_{\gamma,t} := \sum_{m_t} e^{\gamma \sum_{h_t} \hat{P}^{(n-1)}(h_t|m_t) r_t(h_t, m_t)}$.

The Markov property of $\hat{P}^{(n-1)}$ follows from Lemma 5.5.2 and this identity holds true trivially when $t = T$, and can be proved by induction for other cases. □

Suppose our task is to find a machine's policy associated with a Y_γ to maximize $L(Y_\gamma)$ defined in Eq. (5.17). In general, such an objective function is not well-defined in Q because Y_γ is not a function of Q. However, when Assumption 4 takes effect, the causally conditional entropy is not dependent on $\hat{P}^{(n-1)}$:

$$\begin{aligned}\mathbb{H}_{\hat{P}^{(n-1)}Q}(M^T \| H^T) &= E_{\hat{P}^{(n-1)}Q}[-\log Q(M^T \| H^T)] \\ &= E_{\hat{P}^{(n-1)}Q}[-\log \prod_{t=1}^{T} Q_t(M_t)] = \sum_{t=1}^{T} E_{\hat{P}^{(n-1)}Q}[-\log Q_t(M_t)],\end{aligned}$$

where $E_{\hat{P}^{(n-1)}Q}[-\log Q_t(M_t)]$ actually does not depend on $\hat{P}^{(n-1)}$, and we always have
$E_{h(Y_\gamma)Q}[r(H^T, M^T)] = E_{P^*Q}[r(H^T, M^T)]$. In the sequel when Assumption 4

is true we will use the notation $\mathbb{H}_Q(M^T\|H^T)$ where P is suppressed. Then $L(Y_\gamma)$ is actually a function of Q, where $Q = m(Y_\gamma)$ as it can be written as

$$L(Y_\gamma) = L(Q) := \mathbb{H}_Q(M^T\|H^T) + \gamma E_{P^*Q}[r(H^T, M^T)].$$

And still we are able to show the strict monotonicity of $\{L(\hat{Q}^{(n)})\}$.

Moreover, we can show such objective function is indeed concave.

Theorem 5.7.2. *When Assumption 4 is true, $L(Q)$ is strongly concave with parameter $|\mathcal{M}|^T$ in $Q(m^T\|h^T)$.*

Proof. It's easy to observe that $E_{P^*Q}[r(H^T, M^T)]$ is affine. We already know that the causally conditional entropy term is strongly concave in Q when $\hat{P}^{(n-1)}$ is fixed. Now we know that when Assumption 4 is true, the causally conditional entropy term is independent of $\hat{P}^{(n-1)}$. Then it is a strong concave function in Q. □

Theorem 5.7.3. *When Assumption 4 is true, $\{L(\hat{Q}^{(n)})\}$, that is, $\{L^{(n)}\}$ converges to some limit L^∞. If Q^* is the global maximizer of*

$$\max_{Q(m^T\|h^T)} L(Q), \tag{5.24}$$

then

$$L(Q^*) - L^\infty \leq \gamma^2|\mathcal{M}|^{2T} r_{max}. \tag{5.25}$$

Proof. First, according to Theorem 5.6.5, L^∞ must be $L(Y_\gamma^\infty)$ where Y_γ^∞ is a fixed point.

The only difference between Eq.(5.5) and Eq.(5.24) is that in Eq.(5.24), the mean reward is induced by $h(Q)$ which is a function of Q and in Eq.(5.5), that is induced by $\hat{P}$ which is fixed. The gradient of $L(Q)$ is given by:

$$\frac{\partial L(Q)}{\partial Q(m^T\|h^T)} = \frac{\partial \mathbb{H}_Q(M^T\|H^T)}{\partial Q(m^T\|h^T)} + \gamma \left(\frac{\boldsymbol{\partial} E_{PQ}[r(H^T, M^T)]}{\boldsymbol{\partial} P(h^T\|m^T)}\bigg|_{P=h(Q)} \cdot \frac{\boldsymbol{\partial} h(Q)}{\partial Q(m^T\|h^T)} + \frac{\partial E_{PQ}[r(H^T, M^T)]}{\partial Q(m^T\|h^T)}\bigg|_{P=h(Q)} \right),$$

Here we suppress the human model $\hat{P}$ in the entropy term because the entropy is independent of the human model.

And for the Eq.(5.5) at a fixed point, we have:

$$\frac{\partial \mathbb{H}_Q(M^T\|H^T)}{\partial Q(m^T\|h^T)} + \gamma \left. \frac{\partial E_{PQ}[r(H^T, M^T)]}{\partial Q(m^T\|h^T)} \right|_{P=h(Q)} = 0. \tag{5.26}$$

Thus at the fixed point, i.e. when $Q = Q^\infty$,

$$\frac{\partial L(Q)}{\partial Q(m^T\|h^T)} = \gamma \left. \frac{\boldsymbol{\partial} E_{PQ}[r(H^T, M^T)]}{\boldsymbol{\partial} P(h^T\|m^T)} \right|_{P=h(Q)} \cdot \frac{\boldsymbol{\partial} h(Q)}{\partial Q(m^T\|h^T)}.$$

Also, from the moment-matching constraint and Assumption 4 we know $E_{h(Q)Q}[r(H^T, M^T)] = E_{P^*Q}[r(H^T, M^T)]$. Thus we have

$$\begin{aligned} &\frac{\boldsymbol{\partial} E_{PQ}[r(H^T, M^T)]}{\boldsymbol{\partial} P(h^T\|m^T)}\bigg|_{P=h(Q)} \cdot \frac{\boldsymbol{\partial} h(Q)}{\partial Q(m^T\|h^T)} \\ &= \frac{\partial E_{PQ}[r(H^T, M^T)]}{\partial Q(m^T\|h^T)}\bigg|_{P=P^*} - \frac{\partial E_{PQ}[r(H^T, M^T)]}{\partial Q(m^T\|h^T)}\bigg|_{P=h(Q)} \\ &= \left(P^*(h^T\|m^T) - P_Q(h^T\|m^T)\right) r(h^T, m^T), \end{aligned}$$

which is also the gradient of $L(Q)$ at the fixed point.

Then according to the strong concavity, we have

$$L(Q^*) - L(Q^\infty) \leq \frac{|\mathcal{M}|^T}{2} \left(\gamma \sum_{h^T, m^T} (P^*(h^T \| m^T) - P_Q(h^T \| m^T)) r(h^T, m^T) \right)^2$$
$$\leq \gamma^2 |\mathcal{M}|^{2T} r_{max}$$

□

5.8 Evaluation

In this section, we conduct a preliminary numerical evaluation of AREA using synthetic human-machine interaction data based on the Leaky Competing Accumulator (LCA) model, see [87]. This non-linear noisy model is known to capture common human decision-making processes driven by external stimuli.

5.8.1 Numerical Evaluation Set-up

5.8.1.1 Leaky, Competing Accumulator

In the simulation set-up we use a discrete-time version of the original continuous-time version devised in [87]. The Leaky, Competing Accumulator model consists of a set of accumulators $X_t(h)$ for $h \in \mathcal{H}$ at time t, representing the tendency of picking h. The evolution of $X_t(h)$ is driven by following parameters: (1) A self decay coefficient α, capturing the forgetting effect of human memory; (2) An inhibitory coefficient β, capturing the negative impact of the belief in one option to others; (3) Intensity/strength of the external stimuli, ρ, modeling the amount of increment an external stimulus can bring to the associated accumulator. (4) Power of noise σ^2, modeling the randomness in human decisions. At each time t, the recursion of accumulators is given by

$$\forall h \in \mathcal{H},\ \ X_{t+1}(h) = \max\left(0, X_t(h) - \alpha X_t(h) - \beta \sum_{h' \neq h} X_t(h') + \rho \mathbf{1}_{\{S_t = h\}} + \sigma N_{t,h}\right), \tag{5.27}$$

where S_t stands for the external stimulus at time t, and $N_{t,h}$ is an i.i.d. Gaussian noise. Then the human will pick the action with the highest value of

$X_t(h)$ at t.

In our setting we use $\alpha = 0.1, \beta = 0.2, \rho = 0.4, \sigma^2 = 0.09$, and the accumulators are all initialized at 0, so at the very beginning human pick responses uniformly randomly.

5.8.1.2 Q-Learning

The detailed update rule for Q function is as follows. After picking m_t and observing human's response h_t at time t, we do

$$Q((h_{t-\tau}, \ldots, h_{t-1}, m_{t-\tau}, \ldots, m_{t-1}), t, m_t) \leftarrow$$
$$(1-\alpha)Q((h_{t-\tau}, \ldots, h_{t-1}, m_{t-\tau}, \ldots, m_{t-1}), t, m_t) + \alpha\Big(r_t(h_t, m_t)$$
$$+\delta \max_{m \in \mathcal{M}} Q((h_{t-\tau+1}, \ldots, h_t, m_{t-\tau+1}, \ldots, m_t), t, m)\Big).$$

The α in Q-learning is the learning rate and δ is the discount factor to balance the weight between current and future reward. In our evaluation, we set $\alpha = 0.1$ and $\delta = 0.8$, which are values commonly used.

In our simulations, the Q-learning picks its action according softmax of the associated Q function, which means when it observes the latest interactions $(h_{t-\tau+1}, \ldots, h_t, m_{t-\tau+1}, \ldots, m_t)$ and $t+1$, it picks a response $m \in \mathcal{M}$ with probability

$$\frac{e^{cQ((h_{t-\tau+1},\ldots,h_t,m_{t-\tau+1},\ldots,m_t),t+1,m_{t+1})}}{\sum_{m'_{t+1}} e^{cQ((h_{t-\tau+1},\ldots,h_t,m_{t-\tau+1},\ldots,m_t),t+1,m'_{t+1})}}.$$

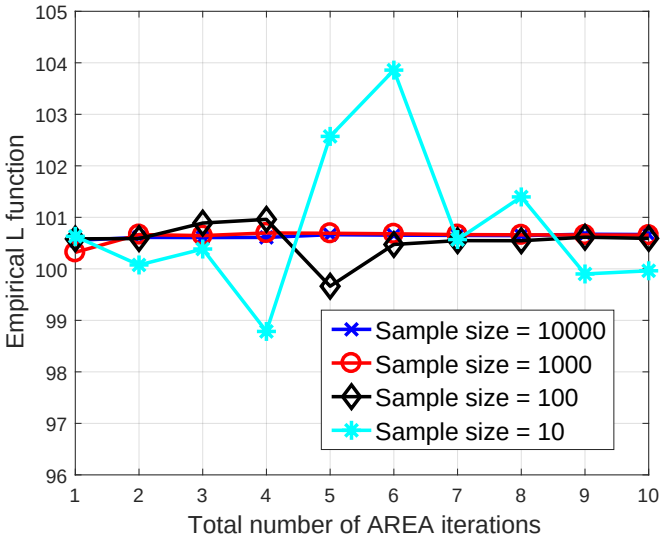
In our simulation we pick $c = 10$ so that Q-learning achieves a comparable average reward as AREA after first 100 samples.

5.8.2 Robustness Against Sampling Noise

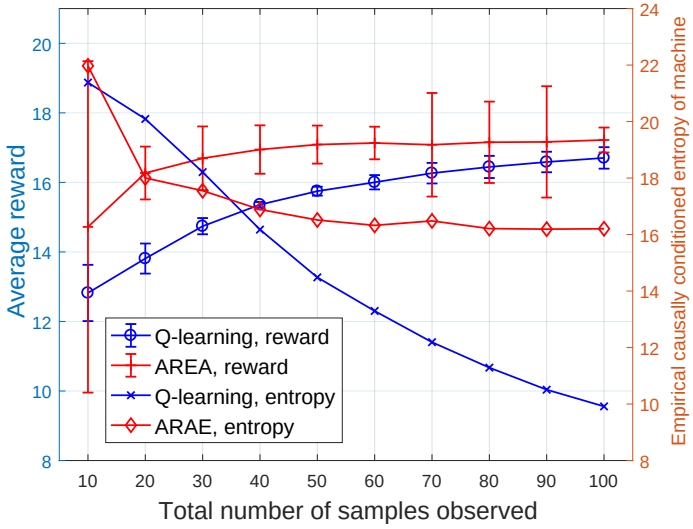
Throughout the paper we have assumed no sampling noise when estimating the moments of features. In practice the available data may be limited or costly and thus noisy estimates are inevitable. The robustness of maximum entropy inference against such noise is mathematically characterized in Theorem 6 of [76]. In this section, we will explore the robustness of the AREA algorithm to noise when the number of samples per iteration are limited. The detailed set-up of the LCA model for human-machine interactions is included in Section 5.8.1.

We consider a setting where $T = 30$, $\mathcal{H} = \{1, \ldots, 6\}$, $\mathcal{M} = \{1, \ldots, 6\}$ and $\gamma = 2$. The reward function is $r(h^T, m^T) = \sum_{t=1}^{T} r_t(h_t, m_t)$, where $r_t(h_t, m_t) = \mathbf{1}_{\{t \bmod 5=0\}}\mathbf{1}_{\{h_t=1\}} + \mathbf{1}_{\{t \bmod 5\neq 0\}}\mathbf{1}_{\{h_t\neq 1\}}$, i.e., we are looking to funnel the human behavior to choosing 1 only at $t = 5, 10, \cdots, 30$. The features include the reward function itself, together with the number of times human follows the machine $f^1(h^T, m^T) = \sum_{t=1}^{T} \mathbf{1}_{\{h_t=m_t\}}$, and a 'weighted' number of times of following occurs emphasizing later times, i.e., $t = 5, 10, \cdots, 30$, $f^2(h^T, m^T) = \sum_{t=1}^{T} f_t^2(h_t, m_t)$, where $f_t^2(h_t, m_t) = (\mathbf{1}_{\{t \bmod 5=0\}}$ $+ 0.25\mathbf{1}_{\{t \bmod 5\neq 0\}})\mathbf{1}_{\{h_t=m_t\}}$. The challenge here is for the machine to learn to drive human (nonlinear model) away from 1 and back to 1 periodically.

The results in Fig. 5.2a exhibit the convergence of the regularized reward function L vs the number of AREA steps, when different numbers of samples are used to estimate the moments in AREA's inference step. Clearly, AREA converges almost immediately although it exhibits variations when



(a) Convergence of AREA vs. sample size of data collection



(b) Average reward vs. number of samples observed

Figure 5.2: Evaluation results

small samples (≤ 100) are used.

5.8.3 Performance in Average Reward and Causally Conditioned Entropy

Next we compare the performance of AREA to a simple Q-learning algorithm [67] with finite memory. We shall compare the attained reward and empirical causally conditioned entropy of the optimized machine policies. In this setting the humans' actions are viewed as the environment. Thus, instead of 'scoring' each action based on the most recent humans' response, Q-learning scores each action based on the most recent τ interactions, together with t to accommodate the transient nature of the process, i.e., it keeps track of $Q((h_{t-\tau+1}, \ldots, h_t, m_{t-\tau+1}, \ldots, m_t), t+1, m_{t+1})$.

At time t, the machine chooses an action using a softmax of the Q function given the latest interaction history $(h_{t-\tau+1}, \ldots, h_t, m_{t-\tau+1}, \ldots, m_t)$ and $t+1$, and then updates the Q function accordingly. We shrink the state space to $|\mathcal{H}| = |\mathcal{M}| = 3$ and $T = 20$ so the Q function fits in the memory and also change γ to 4 to put more emphasis on reward. We shall consider the same rewards and features as in Section 5.8.2. of AREA. We will let both algorithms complete 100 'interactions' with our synthetic human model. For AREA, we collect 10 human-machine interaction samples per AREA iteration, and run 10 iterations in total. For Q-learning we also allow a total of 100 interactions. We set $\tau = 1$ since further experiments show that greater τ impairs the performance of Q-learning for it requires more samples to learn. The detailed setup for Q-learning can be found in Section 5.8.1. We kept track of the average reward obtained, estimated causally conditioned entropy of machine

obtained for both algorithms after integrating the first n samples. We run the simulation 5 rounds to obtain the average, and the results, together with the 90% confidence intervals are shown in Figure 5.2b. These representative results suggest that typically AREA algorithm is very efficient, delivering higher rewards than Q-learning while at the same time realizing (as desired) higher machine policy entropy with a very limited number of samples.

Chapter 6

Conclusion and Future work on Human-machine Interactive Processes

This thesis proposes a general data-driven framework to optimize possibly complex human-machine interaction processes. At the core is the AREA algorithm which jointly solves the problem of estimating a model for human behavior and optimizing the machine policy based on a constrained maximum entropy estimation. An underlying goal is to enable the integration of domain-specific knowledge regarding relevant interaction characteristics or known human biases by matching the observed moments of feature functions. This thesis details the formal optimization problems and solutions underlying the AREA algorithm and explores a modification to significantly reduce the complexity when the feature and reward functions are path-based and/or decomposable. The setting considered is fairly general, allowing one to incorporate human-machine interactions with long memory. The characterization of AREA is provided in terms of (i) its space and time complexity, and (ii) its convergence in various settings. A simple numerical evaluation is used to demonstrate the robustness of AREA to noise when sample sizes are limited, along with a performance comparison to Q-learning. The analysis and simple validation suggest that AREA may achieve most of its gains in one iteration

particularly if sufficient domain specific features/rewards are properly integrated.

In fact, if long-term dependency and sequential effect need to be taken into account, exponential complexity is always a substantial challenge in modelling. To accommodate it, different approaches to embedding the exponential dependencies to a lower-dimension space are proposed, achieving different trade-offs. For example, in deep Q-learning [71], a deep neural network is trained on a massive dataset to maintain the most significant information in the dependency, while in AREA, such embedding is achieved by limiting the feature functions to path-based and/or decomposable features only, which allows us to get rid of the need of massive data. Therefore AREA is more suitable in a reinforcement learning context where the interactions might take a significant amount of time. This is usually the case when the model interacts with real human.

Beyond the scope of this thesis, AREA can be extended in the following ways.

1. *Choice of feature functions.* The performance and complexity of AREA heavily depend on choice of feature functions. Thus, it is critical to investigate how to model aspects patterns and/or biases within human decisions via properly chosen feature functions.

2. *Change and transience detection in interactions.* AREA, as well as most of reinforcement learning algorithms, works under the assumption that

the underlying decision process is stable across iterations. However in practice, one can expect human behavior to change over time. For example, human's skill level, knowledge, and/or level of interest/attention might improve or decline as the interactions proceed. Once the underlying decision process deviates from the original one significantly, AREA needs to re-initialize in order to capture the characterization of the new underlying model. Presumably this can be done by a *significance test*, but further efforts are required in formalizing methods for tracking of change in human behavior and triggering re-optimization.

Bibliography

[1] Service aspects; Service principles. 3GPP TS 22.101, v14.0.0, Jun. 2015.

[2] Telecommunication management; Network sharing; Concepts and requirements. 3GPP TS 32.130, v12.0.0, Dec. 2014.

[3] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Nsdi*, volume 11, pages 24–24, 2011.

[4] Eitan Altman, Konstantin Avrachenkov, and Urtzi Ayesta. A survey on discriminatory processor sharing. *Queueing systems*, 53(1-2):53–63, 2006.

[5] Na Chen and Scott Jordan. Throughput in processor-sharing queues. *IEEE Transactions on Automatic Control*, 52(2):299–305, 2007.

[6] J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs. Statistical multiplexing and traffic shaping games for network slicing. In *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8, May 2017.

[7] J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs. Statistical multiplexing and traffic shaping games for network slicing (extended). *IEEE/ACM Transactions on Networking*, 26(6):2528–2541, Dec 2018.

[8] Jiaxiao Zheng and Gustavo de Veciana. Elastic multi-resource network slicing: Can protection lead to improved performance? In *17th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'19)*, Avignon, France, June 2019.

[9] Jiaxiao Zheng and Gustavo de Veciana. Elastic multi-resource network slicing: Can protection lead to improved performance (extended). https://arxiv.org/abs/1901.07497, 2019.

[10] Jiaxiao Zheng and Gustavo de Veciana. Modeling and optimization of human-machine interaction processes via the maximum entropy principle. *arXiv preprint arXiv:1903.07157*, 2019.

[11] Tao Guo and Rob Arnott. Active lte ran sharing with partial resource reservation. In *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pages 1–5. IEEE, 2013.

[12] Abhay Kumar J Parekh. *A Generalized Processor Sharing Approach to Flow Control In Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, 1992.

[13] Rayadurgam Srikant and Lei Ying. *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[14] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge,

2007.

[15] Albert Banchs. User fair queuing: fair allocation of bandwidth for users. In *IEEE INFOCOM*, volume 3, pages 1668–1677. IEEE, 2002.

[16] Michal Feldman et al. The proportional-share allocation market for computational resources. *IEEE TPDS*, 20(8):1075–1088, 2009.

[17] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez. Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads. *IEEE/ACM Transactions on Networking*, PP(99):1–15, 2017.

[18] Simina Brąnzei, Yiling Chen, Xiaotie Deng, Aris Filos-Ratsikas, Sǫren Kristoffer Stiil Frederiksen, and Jie Zhang. The fisher market game: equilibrium and welfare. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[19] Gustavo de Veciana, Tae-Jin Lee, and Takis Konstantopoulos. Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Transactions on Networking*, 9(1):2–14, 2001.

[20] Thomas Bonald and Laurent Massoulié. Impact of fairness on internet performance. In *ACM SIGMETRICS Performance Evaluation Review*, volume 29, pages 82–91. ACM, 2001.

[21] Thomas Bonald and Alexandre Proutiere. Insensitive bandwidth sharing in data networks. *Queueing systems*, 44(1):69–100, 2003.

[22] Virag Shah and Gustavo de Veciana. Performance evaluation and asymptotics for content delivery networks. In *INFOCOM, 2014 Proceedings IEEE*, pages 2607–2615. IEEE, 2014.

[23] Xuan Zhou, Rongpeng Li, Tao Chen, and Honggang Zhang. Network slicing as a service: enabling enterprises' own software-defined cellular networks. *IEEE Communications Magazine*, 54(7):146–153, Jul. 2016.

[24] M. Richart, J. Baliosian, J. Serrat, and J. L. Gorricho. Resource slicing in virtual wireless networks: A survey. *IEEE Transactions on Network and Service Management*, 13(3):462–476, Sept 2016.

[25] X. Costa-Perez et al. Radio access network virtualization for future mobile carrier networks. *IEEE Comm. Magazine*, 51(7):27–35, 2013.

[26] K. Samdanis, X. Costa-Perez, and V. Sciancalepore. From network sharing to multi-tenancy: The 5G network slice broker. *IEEE Communications Magazine*, 54(7):32–39, July 2016.

[27] R. Kokku et al. NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks. *IEEE/ACM TON*, 20(5):1333–1346, 2012.

[28] C. J. Bernardos et al. An architecture for software defined wireless networking. *IEEE Wireless Communications*, 21(3):52–61, 2014.

[29] Frank P Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 2011.

[30] Li Zhang. Proportional response dynamics in the Fisher market. *Theoretical Computer Science*, 412(24):2691–2698, May 2011.

[31] Shanchieh Jay Yang and Gustavo de Veciana. Enhancing both network and user performance for networks supporting best effort traffic. *IEEE/ACM TON*, 12(2):349–360, 2004.

[32] Laurent Massoulié and James Roberts. Bandwidth sharing: objectives and algorithms. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1395–1403. IEEE, 1999.

[33] François Baccelli and Pierre Brémaud. *Palm probabilities and stationary queues*, volume 41. Springer Science & Business Media, 2012.

[34] Stephem Boyd. Slides of convex optimization. `http://web.stanford.edu/~boyd/cvxbook/bv_cvxslides.pdf`, 2019.

[35] Christophe Dutang. Existence theorems for generalized Nash equilibrium problems. *J. Nonlinear Analysis & Opt.*, 4(2):115–126, 2013.

[36] Francisco Facchinei and Christian Kanzow. Penalty methods for the solution of generalized Nash equilibrium problems. *SIAM Journal on Optimization*, 20(5):2228–2253, 2010.

[37] Anna Von Heusinger and Christian Kanzow. Relaxation methods for generalized Nash equilibrium problems with inexact line search. *Journal of Optimization Theory and Applications*, 143(1):159–183, 2009.

[38] ITU-R. Report ITU-R M.2135-1, Guidelines for evaluation of radio interface technologies for IMT-Advanced. Technical Report, 2009.

[39] Qiaoyang Ye et al. User Association for Load Balancing in Heterogeneous Cellular Networks. *IEEE Trans. Wireless Comm.*, pages 2706–2716, 2013.

[40] Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures. 3GPP TS 36.213, v12.5.0, Rel. 12, Mar. 2015.

[41] E. Hyytia, P. Lassila, and J. Virtamo. Spatial node distribution of the random waypoint mobility model with applications. *IEEE TMC*, 5(6):680–694, June 2006.

[42] K. Lee et al. SLAW: Self-similar least-action human walk. *IEEE/ACM TON*, 20(2):515–529, 2012.

[43] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[44] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[45] Y. Cao, H. Song, O. Kaiwartya, B. Zhou, Y. Zhuang, Y. Cao, and X. Zhang. Mobile edge computing for big-data-enabled electric vehi-

cle charging. *IEEE Communications Magazine*, 56(3):150–156, March 2018.

[46] A. Ahmed and E. Ahmed. A survey on mobile edge computing. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–8, Jan 2016.

[47] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez. Multi-tenant radio access network slicing: Statistical multiplexing of spatial loads. *IEEE/ACM Transactions on Networking*, 25(5):3044–3058, Oct 2017.

[48] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

[49] Hal R Varian. Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63 – 91, 1974.

[50] H Peyton Young. *Equity: in theory and practice*. Princeton University Press, 1995.

[51] Thomas Bonald and James Roberts. Multi-resource fairness: Objectives, algorithms and performance. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, pages 31–42. ACM, 2015.

[52] T. Lan, D. Kao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.

[53] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[54] Omar Soliman, Abdelmounaam Rezgui, Hamdy Soliman, and Najib Manea. Mobile cloud gaming: Issues and challenges. In *International Conference on Mobile Web and Information Systems*, pages 121–128. Springer, 2013.

[55] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, Oct 2000.

[56] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[57] Michal Pióro and Deep Medhi. *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[58] Jim G Dai and Sean P Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control*, 40(11):1889–1904, 1995.

[59] Jim G Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, pages 49–77, 1995.

[60] JG Dai et al. A fluid limit model criterion for instability of multiclass queueing networks. *The Annals of Applied Probability*, 6(3):751–757, 1996.

[61] Wikipedia contributors. Norm (mathematics) — Wikipedia, the free encyclopedia, 2018. [Online; accessed 21-December-2018].

[62] Miranda JG van Uitert. *Generalized processor sharing queues*. Eindhoven University of Technology, 2003.

[63] Sissie Ling-Ie Hsiao, Chao Cai, Eric W Ewald, Cameron M Tangney, A Walker II Robert, Japjit Tulsi, Ming Lei, and Zhimin He. Conversion path performance measures and reports, January 26 2016. US Patent 9,245,279.

[64] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[65] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.

[66] Nicole Bäuerle and Ulrich Rieder. *Markov decision processes with applications to finance*. Springer Science & Business Media, 2011.

[67] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[68] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[69] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[70] Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 64(5):482, 1962.

[71] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[72] Francisco A Melo, M Isabel Ribeiro, and ISR Torre Norte. The use of transition entropy in partially observable markov decision processes. *Institute for Systems and Robotics, Tech. Rep. RT-601-05*, 2005.

[73] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646, 2014.

[74] Nicholas Epley. *Mindwise: Why we misunderstand what others think, believe, feel, and want*. Vintage, 2015.

[75] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Maximum causal entropy correlated equilibria for markov games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*,

pages 207–214. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[76] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. The principle of maximum causal entropy for estimating interacting processes. *Information Theory, IEEE Transactions on*, 59(4):1966–1980, 2013.

[77] Kevin Waugh, Brian D Ziebart, and J Andrew Bagnell. Computational rationalization: The inverse equilibrium problem. *arXiv preprint arXiv:1308.3506*, 2013.

[78] J.W. Payne, J.R. Bettman, and E.J. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, New York, 1993.

[79] W.T. Maddox and A. B. Markman. The motivation-cognition interface in learning and decision making. *Current Directions in Psychological Science*, 19(2):106–110, 2010.

[80] W.D. Gray and W.T. Fu. Soft constraints in interactive behavior: The case of ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head. *Cognitive Science*, 28:359–382, 2004.

[81] Christopher G Atkeson and Juan Carlos Santamaria. A comparison of direct and model-based reinforcement learning. In *In International Conference on Robotics and Automation*, 1997.

[82] Leonid Kuvayev Rich Sutton. Model-based reinforcement learning with an approximate, learned model. In *Proceedings of the ninth Yale workshop on adaptive and learning systems*, pages 101–105, 1996.

[83] Peng Guan, Maxim Raginsky, and Rebecca M Willett. Online markov decision processes with kullback–leibler control cost. *IEEE Transactions on Automatic Control*, 59(6):1423–1438, 2014.

[84] Hemant Tyagi and Bernd Gärtner. Continuum armed bandit problem of few variables in high dimensions. In *International Workshop on Approximation and Online Algorithms*, pages 108–119. Springer, 2013.

[85] John M Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. 1971.

[86] Willard I Zangwill. Convergence conditions for nonlinear programming algorithms. *Management Science*, 16(1):1–13, 1969.

[87] Marius Usher and James L McClelland. The time course of perceptual choice: the leaky, competing accumulator model. *Psychological review*, 108(3):550, 2001.

Vita

Jiaxiao Zheng was born in Zhengzhou, Henan, China on August 11th, 1990, the son of Shiyong Zheng and Limin Xie. He received the Bachelor of Science degree in Electronic Science and Technology from the School of Electrical and Information Engineering of Shanghai Jiao Tong University in 2013. After that, he applied to the University of Texas at Austin for enrollment in their doctoral program in Electrical and Computer Engineering. He was accepted and started his graduate studies under the supervision of Dr. Gustavo de Veciana from 2013. During his graduate studies, he finishes two internships with Google in the summer of 2017 and 2018, respectively, working on network anomaly detection and big data infrastructure.

Contact: jxzheng39@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.