Tractable Algorithms for Sequential Decision Making Problems

Nikhil Bhat

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2016

ABSTRACT

Tractable Algorithms for Sequential Decision Making Problems

Nikhil Bhat

Sequential decision making problems are ubiquitous in a number of research areas such as operations research, finance, engineering and computer science. The main challenge with these problems comes from the fact that, firstly, there is uncertainty about the future. And secondly, decisions have to be made over a period of time, sequentially. These problems, in many cases, are modeled as Markov Decision Process (MDP).

Most real-life MDPs are 'high dimensional' in nature making them challnging from a numerical point of view. We consider a number of such high dimensional MDPs. In some cases such problems can be approximately solved using Approximate Dynamic Programming. In other cases problem specific analysis can be solved to device tractable policies that are near-optimal.

In Chapter 2, we presents a novel and practical non-parametric approximate dynamic programming (ADP) algorithm that enjoys graceful, dimension-independent approximation and sample complexity guarantees. In particular, we establish both theoretically and computationally that our proposal can serve as a viable replacement to state of the art parametric ADP algorithms, freeing the designer from carefully specifying an approximation architecture. We accomplish this by 'kernelizing' a recent mathematical program for ADP (the 'smoothed' approximate LP) proposed by [Desai *et al.*, 2011].

In Chapter 3, we consider a class of stochastic control problems where the action space at each time can be described by a class of matching or, more generally, network flow polytopes. Special cases of this class of dynamic matching problems include many problems that are well-studied in the literature, such as: (i) online keyword matching in Internet advertising (the adwords problem); (ii) the bipartite matching of donated kidneys from cadavers to recipients; and (iii) the allocation of donated kidneys through exchanges over cycles of live donor-patient pairs. We provide

an approximate dynamic program (ADP) algorithm for dynamic matching with stochastic arrivals and departures. Our framework is more general than the methods prevalent in the literature in that it is applicable to a broad range of problems characterized by a variety of action polytopes and generic arrival and departure processes.

In Chapter 4, we consider the problem of A-B testing when the impact of the treatment is marred by a large number of covariates. Randomization can be highly inefficient in such settings, and thus we consider the problem of optimally allocating test subjects to either treatment with a view to maximizing the efficiency of our estimate of the treatment effect. Our main contribution is a tractable algorithm for this problem in the online setting, where subjects arrive, and must be assigned, sequentially. We characterize the value of optimized allocations relative to randomized allocations and show that this value grows large as the number of covariates grows. In particular, we show that there is a lot to be gained from 'optimizing' the process of A-B testing relative to the simple randomized trials that are the mainstay of A-B testing in the 'big data' regime of modern e-commerce applications, where the number of covariates is often comparable to the number of experimental trials.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Last five years have had many moments to look back upon with much fondness. I was fortunate enough to be able to work with my advisor, Ciamac. Over the years, whenever I hit a roadblock in my work, I could rely on him for the right direction. Ciamac's genuine willingness to help people out along with his clarity of thought make him an exceptional advisor.

A lot of credit for my work also goes to my coadvisor, Vivek. He backed up his ideas with an incredible work rate. Vivek has been a constant source of positivity.

I'd also like to thank Jose, Costis, Assaf and Garud for being on my commitee and their careful review of my work.

I was also fortunate to make some very close friends at Columbia. Juan, Daniela, Carlos, Daniel, Shyam and Peter, to name a few.

I'd like to thank my family for the unconditional support and kindness.

Finally, I'd like to thank Kritika for making the last few years extra special. Her kindness, companionship and sense of humor keeps making life blissful and helps put work in perspective.

Chapter 1

Introduction

1.1 Background

Sequential decision making problems are ubiquitous in a number of research areas such as operations research, finance, engineering and computer science. The main challenge with these problems comes from the fact that, firstly, there is uncertainty about the future. And secondly, decisions have to be made over a period of time, sequentially. These problems, in many cases, are modeled as Markov Decision Process (MDP). Advantage of this model is that, while making decisions at any stage, we can summarize the impact of all the history in the 'state' of the MDP. This is the so-called principle of Dynamic Programming (DP).

The DP principle allows us to find the *optimal solution* of the problem using backward induction. Numerical methods such as policy iteration and value iteration can then be used to solve the MDP to optimality. However, this is not a realistically feasible approach if the dimension of the state is large, say more than 5. This is widely known as the curse of dimensionality.

The high dimensionality of these problems is more of a norm than an exception. This is especially true in the internet and e-commerce application where the decision maker has a lot of 'context' at her disposal. This context is in the form of hundreds or potentially thousands of attributes. Such a proliferation of information usually manifests itself in a high dimensional state space for the MDP. Thus posing an algorithmic challenge to the decision maker.

In this thesis we focus on a number of such high-dimensional sequential decision making problems. We will develop a non-parametric algorithm which can be used for any generic MDP. We will

also focus on some real life sequential decision making problem posing them as MDPs. For these problems we will come up with algorithms that exploit the problem structure.

In Section 1.2, we will briefly introduce MDPs, in particular the infinite horizon discounted formulation of the MDP. In Section 1.3, we will give a number of examples of real-life problems that can be posed as MDPs. We will focus on examples that are studied extensively in the later chapters of this thesis. In Section 1.4, we will look at some ways in which people approach these problems algorithmically. Approximate Dynamic Programming is a term used, especially in the operations research community, for such approaches. In certain other cases, the problem structure of a high dimensional MDP can be exploited to device tractable algorithms. In Section 1.5 we list our contributions to the existing literature.

1.2 Markov Decision Process

In this section we will introduce the infinite horizon discounted cost version of Markov Decision Process formulation. Although this is not the most general formulation of MDPs possible, it does capture the challenges associated with solving such problems.

In a Markov Decision Process, one must take actions each time over a given time horizon. For the purpose of this chapter we consider a decision problem over an infinite time-horizon. Central to MDP is the idea of the state. State, at any point of time $0 \leq t < \infty$ captures all the information of the history of the process up to time t that is relevant for the future actions. We will call the state at time t as $x_t \in \mathcal{X}$, where $\mathcal{X}$ is a set of all possible states called the *state space*.

Given that at time t the state is x, we must pick an action from the action space $\mathcal{A}$. Given that the current state is x and we pick $a \in \mathcal{A}$, the state at time $t+1$ is drawn from a probability distribution $p(x_t, a_t, \cdot)$. In particular,

$$\mathsf{P}(X_{t+1} = x_{t+1} | X_t = x, a_t = a) = p(x, a, x_{t+1}).$$

We assume that this 'transition kernel' is known to the decision maker. At each step, we incur some cost. This cost depends on the current state and action. In particular, there exists a function g such that if at state x_t we take action a_t then the cost incurred is $g(x_t, a_t)$. Our aim is to pick

actions a_t to minimize,

$$\mathsf{E}_{X_0\sim\nu}\left[\sum_{t=0}^{\infty}\alpha^t g(X_t,a_t)\right].$$

Here $\alpha\in(0,1)$ is called the discount factor and ν is the initial distribution.

Due to the Markovian nature of the system, the action at each time period a_t needs to depend only on the state at that time period x_t. Thus, we must optimize over functions of the form $\{\mu_t\}_{0\leq t}$, where $\mu_t:\mathcal{X}\to\mathcal{A}$. $\{\mu_t\}_{0\leq t}$ is typically called a policy. Given a policy $\{\mu_t\}_{0\leq t}$, at time t, if the current state is x_t, we would take action $\mu_t(x_t)$. It is well know that for the problem under consideration the search for optimal policy can be restricted to time-homogeneous policies. Thus our optimization problem reduces to,

$$\underset{\mu\in\mathcal{M}}{\text{minimize}}\,\mathsf{E}_{X_0\sim\nu}\left[\sum_{t=0}^{\infty}\alpha^t g(X_t,\mu(X_t))\right],$$

where $\mathcal{M}$ is the set of all mappings from $\mathcal{X}$ to $\mathcal{A}$. Again, it is well known that the optimal of this program exists and let us call it μ^*. We call this the *optimal policy*.

Consider a function $J^*:\mathcal{X}\to\mathbb{R}$, such that $J^*(x)$ denotes the expected minimum achievable cost starting from state x. In other words,

$$J^*(x)\triangleq\min_{\mu\in\mathcal{M}}\mathsf{E}_{X_0=x}\left[\sum_{t=0}^{\infty}\alpha^t g(X_t,\mu(X_t))\right].$$

By the assertion that μ^* is optimal, we have,

$$J^*(x)=\mathsf{E}_{X_0=x}\left[\sum_{t=0}^{\infty}\alpha^t g(X_t,\mu^*(X_t))\right].$$

Similarly, let us define,

$$J^{\mu}(x)\triangleq\mathsf{E}_{X_0=x}\left[\sum_{t=0}^{\infty}\alpha^t g(X_t,\mu(X_t))\right],$$

for any policy μ.

The function J^* plays an important role in dynamic programming. It is typically called the *optimal cost-to-go* function. Consider a policy μ^{J^*} such that,

$$\mu^{J^*}(x)\in\underset{a\in\mathcal{A}}{\operatorname{argmin}}\left[g(x,a)+\alpha\mathsf{E}_{x,a}[J^*(X')]\right].$$

It turns out that μ^{J^*} is in fact an optimal policy. Thus the knowledge of J^* would be enough to find the optimal policy.

For a given cost-to-go function $J : \mathcal{X} \to \mathbb{R}$, let the policy μ^J be defined as,

$$\mu^J(x) \in \underset{a \in \mathcal{A}}{\operatorname{argmin}} \left[g(x,a) + \alpha \mathsf{E}_{x,a}[J^*(X')] \right].$$

We call such a policy *greedy* with respect to the cost-to-go function J. A common approach to solving high-dimensional sequential decision making problems is to approximate J^*. If we find a function $\tilde{J}$ that closely approximates J^*, then one would hope that the policy *greedy* with respect to $\tilde{J}$ would be close to μ^*.

1.3 Motivating Examples

In this section we will look at a number of real life problems that can be viewed from the MDP framework. We will revisit these problems in the later chapters in the thesis.

1.3.1 Controlled Queueing Network

Consider a service network where jobs arrive asynchronously. Each job must follow a designated route through the service network. Along the way these jobs must be processed by servers. In general, a job might not be allocated a server instantaneously and in the case it will be added to a First In First Out (FIFO) queue before servicing. When more than one queue is full at a service station, the decision maker must pick a queue to serve at a particular point point of time.

Queueing networks are naturally modeled in continuous time with asynchronous arrivals and completions of jobs. If the inter arrival times and service times are assumed to follow the exponential distribution then the decision problem can be posed in discrete time. This process is called *uniformization*. For details, we refer the reader to [Moallemi *et al.*, 2008].

In Section 2.5 we demonstrate how this problem can be approached from the MDP point of view. The state of the system, at any point of time is just a vector of the queue lengths in all the buffers. Naturally the MDP is a high dimensional one, since any realistic queueing network is likely to consist of a number of buffers.

Fabrication processes, network routers and call centers can be modeled as a service network. Not surprisingly, a lot a literature has focused on coming up with policies specifically with this problem in mind. In Section 2.5, we apply a non-parametric ADP algorithm to a controlled queueing

network. We show that our approach outperforms many heuristic approaches designed specifically for this problem.

1.3.2 Stochastic Assignment

Stochastic Assignment is a classic problem studied in the operations research literature. The problem essentially is a bipartite matching problem. The challenging part is that only one side of the bipartite matching is known and the items on the other side arrive, one at a time, in a sequential way. One must assign these arriving items immediately without the knowledge of the future arrivals, or the item will perish. Each match gives us some reward that depends on the types of the items involved in the match. Many real-life problems such as ad allocation and kidney exchanges have been modeled as stochastic assignment.

Under the assumption that the types of the arriving items are i.i.d. from a set, it is quite straightforward to see that this is in fact an MDP. The state space for this problem is all the multi-sets on the set of types, a truly enormous space. There are a number of approaches in the literature that have studied this problem under the assumption that the number of types is small and finite, in the asymptotic regime. In real-life situations such as kidney exchanges, this assumption is invalid.

In Chapter 3, we will look at this problem in more detail. We will introduce a broad framework for dynamic matching problems and Stochastic Assignment will just be a special case of this model.

1.3.3 Optimal A-B Testing

Consider an e-commerce company testing an alternative recommendation system. The company wants to access the impact the new system has on the metrics like clicks and sales. In most cases before launching a major change like this, the company will perform user experiments. The obvious way to approach is to implement an A-B test.

Typically, as users arrive on the site they will be assigned to either treatment or control. Based on this assignment, the visitor will be shown recommendations from either the old system or new. Typically in A-B test the assignments will be done by flipping an unbiased coin. After the experiment runs for some time, the data can analyzed to estimate the effect of the treatment.

Typically the experimenter will have a lot of information about the subjects. This context summarized with a covariate vector will affect the outcome of each trial. Thus the presence of

these covariates will make the inference process more complex. If we assume a linear dependence of the outcome on the covariates and treatment choice then the treatment effect can be estimated by least squares.

We consider the alternative to randomly assigning subjects to treatment and control. We consider the making allocations in an 'optimal' way. Our optimality will be from the point of minimizing the variance over the collection of all unbiased estimators.

In Chapter 4, we pose this problem as an MDP. In essence, the state of the MDP imbalance in the number of assignments in the treatment and control group and the imbalance in the sum of covariates in the two groups. The objective tries to minimize this imbalance.

1.4 Approximate Dynamic Programming

Approximate Dynamic Programming (ADP) is an umbrella term used to describe tractable algorithms applicable to a broad category of MDPs. ADP algorithms typically try to approximate J^* of the form $J^* \approx z^\top \Phi(x) \triangleq \tilde{J}(x)$, where $\Phi : \mathcal{X} \to \mathcal{H}$ is referred to as the 'approximation architecture'. The ADP algorithm computes a weight vector $z \in \mathcal{H}$. One then employs a policy that is greedy with respect to the corresponding approximation $\tilde{J}$.

In most ADP algorithms $\mathcal{H}$ is just an m dimensional Euclidean space for some finite m. However, one can imagine $\mathcal{H}$ being an infinite dimensional Hilbert space and $J^*(x) \approx \langle z, \Phi(x) \rangle$, with the corresponding inner product. The former type of ADP algorithms can be classified as 'parametric' ADP algorithms and the later 'non-parametric' ADP algorithms.

In this thesis we will focus on optimization based ADP algorithms. It can be shown that the optimal cost-to-go J^* can be obtained by solving the following linear program:

$$\begin{aligned} \text{maximize} \quad & \nu^\top J \\ \text{subject to} \quad & J(x) \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[J(X')], \quad \forall\, x \in \mathcal{X},\ a \in \mathcal{A}, \\ & J \in \mathbb{R}^{\mathcal{X}}, \end{aligned} \tag{1.1}$$

for any strictly positive vector $\nu \in \mathbb{R}^{\mathcal{X}}_+$.

Instead of solving for an arbitrary vector $J \in \mathbb{R}^{\mathcal{X}}$, one can restrict the search of $J(x) = \Phi(x)^\top z$.

This yields the so-called Approximate Linear Programming formulation of ADP:

$$\begin{aligned} \text{maximize} \quad & \sum_{x \in \mathcal{X}} \nu_x z^\top \Phi(x) \\ \text{subject to} \quad & z^\top \Phi(x) \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[z^\top \Phi(X')], \quad \forall\ x \in \mathcal{X},\ a \in \mathcal{A}, \\ & z \in \mathbb{R}^m,\ s \in \mathbb{R}_+^{\mathcal{X}}. \end{aligned} \tag{1.2}$$

This formulation was first proposed in [Schweitzer and Seidman, 1985] and later theoretically analyzed in [de Farias and Van Roy, 2004]. Their analysis that, stated loosely, showed,

$$\|J^* - z^{*\top} \Phi\|_{1,\nu} \leq \frac{2}{1-\alpha} \inf_z \ \|J^* - z^\top \Phi\|_\infty,$$

for an optimal solution z^* to the ALP.

Based on this [Desai *et al.*, 2011] introduced the Smoothed Approximate Linear Program:

$$\begin{aligned} \text{maximize} \quad & \sum_{x \in \mathcal{X}} \nu_x z^\top \Phi(x) - \kappa \sum_{x \in \mathcal{X}} \pi_x s_x \\ \text{subject to} \quad & z^\top \Phi(x) \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[z^\top \Phi(X')] + s_x, \quad \forall\ x \in \mathcal{X},\ a \in \mathcal{A}, \\ & z \in \mathbb{R}^m,\ s \in \mathbb{R}_+^{\mathcal{X}}. \end{aligned} \tag{1.3}$$

In their paper, they showed that the ALP bounds could be improved upon by 'smoothing' the constraints of the ALP, i.e., permitting positive slacks. The ALP constraints impose the restriction that $TJ \geq J$. This ensures that the optimal solution lower bounds the optimal value function at each state. But at the cost of providing a lower bound, the ALP might do a poor job of approximating J^* due to presence of constraints associated with certain states that are rarely visited. SALP prevents this by allowing violations of Bellman constrains. Frequently visited states are given more weight to prevent large Bellman errors at those states.

1.5 Contributions

Here we list our contributions:

A Novel Non-parametric ADP algorithm. We presents a novel and practical non-parametric approximate dynamic programming (ADP) algorithm that enjoys graceful, dimension-independent approximation and sample complexity guarantees. In particular, we establish both theoretically and computationally that our proposal can serve as a viable replacement to state of the art parametric

ADP algorithms, freeing the designer from carefully specifying an approximation architecture. We accomplish this by 'kernelizing' a recent mathematical program for ADP (the 'smoothed' approximate LP) proposed by [Desai *et al.*, 2011]. Our theoretical guarantees establish that the quality of the approximation produced by our procedure improves gracefully with sampling effort. Via a computational study on a controlled queueing network, we show that our non-parametric procedure outperforms the state of the art parametric ADP approaches and established heuristics.

A Broad Framework for Dynamic Matching Problems with Tractable Algorithms. We consider a class of stochastic control problems where the action space at each time can be described by a class of matching or, more generally, network flow polytopes. Special cases of this class of dynamic matching problems include many problems that are well-studied in the literature, such as: (i) online keyword matching in Internet advertising (the adwords problem); (ii) the bipartite matching of donated kidneys from cadavers to recipients; and (iii) the allocation of donated kidneys through exchanges over cycles of live donor-patient pairs. We provide an approximate dynamic program (ADP) algorithm for dynamic matching with stochastic arrivals and departures. Our framework is more general than the methods prevalent in the literature in that it is applicable to a broad range of problems characterized by a variety of action polytopes and generic arrival and departure processes. We apply our methodology to a series of kidney matching problems calibrated to realistic kidney exchange statistics, where we obtain a significant performance improvement over established benchmarks.

Optimal Algorithm for A-B Testing Assignments. We consider the problem of A-B testing when the impact of the treatment is marred by a large number of covariates. Randomization can be highly inefficient in such settings, and thus we consider the problem of optimally allocating test subjects to either treatment with a view to maximizing the efficiency of our estimate of the treatment effect. Our main contribution is a tractable algorithm for this problem in the online setting, where subjects arrive, and must be assigned, sequentially. We characterize the value of optimized allocations relative to randomized allocations and show that this value grows large as the number of covariates grows. In particular, we show that there is a lot to be gained from 'optimizing' the process of A-B testing relative to the simple randomized trials that are the mainstay of A-B testing in the 'big data' regime of modern e-commerce applications, where the number of covariates is often comparable to the number of experimental trials.

Chapter 2

Non-parametric Approximate Dynamic Programming

2.1 Introduction

Problems of dynamic optimization in the face of uncertainty are frequently posed as Markov decision processes (MDPs). The central computational problem is then reduced to the computation of an optimal 'value' or 'cost-to-go' function that encodes the value garnered under an optimal policy starting from any given MDP state. MDPs for many problems of practical interest frequently have intractably large state spaces precluding exact computation of the cost-to-go function. Approximate dynamic programming (ADP) is an umbrella term for algorithms designed to produce good approximations to this function. Such approximations then imply a natural 'greedy' control policy.

ADP algorithms are, in large part, *parametric* in nature. In particular, the user specifies an 'approximation architecture' (i.e., a set of basis functions) and the algorithm then produces an approximation in the span of this basis. The strongest theoretical results available for such algorithms typically share the following two features:

- The quality of the approximation produced is comparable with the best possible within the basis specified.
- The sample complexity or computational effort required for these algorithms scales, typically polynomially, with the dimension of the basis.

These results highlight the importance of selecting a 'good' approximation architecture, and remain somewhat dissatisfying in that additional sampling or computational effort cannot remedy a bad approximation architecture.

In contrast, an ideal *non-parametric* approach would, in principle, free the user from carefully specifying a suitable low-dimensional approximation architecture. Instead, the user would have the liberty of selecting a very rich architecture (such as, say, the Haar basis). The quality of the approximation produced by the algorithm would then improve — gracefully — with the extent of computational or sampling effort expended, ultimately becoming exact. Unfortunately, existing non-parametric proposals for ADP fall short of this ideal on one or more fronts. In particular, the extant proposals include:

Kernelizing Policy Evaluation. The key computational step in approximate policy iteration methods is 'policy evaluation'. The aim of this step is to find the cost-to-go function for a given policy. This step involves solving the projected Bellman equation, a linear stochastic fixed point equation. Due to the curse of dimensionality, solving this exactly is infeasible and a large portion of ADP literature focuses on approximately solving the policy evaluation step. This usually involves approximating the cost-to-go function for a particular policy with a parametric approximation architecture. Subsequently, there have been efforts to make this step non-parametric. [Bethke *et al.*, 2008] study the problem of minimizing Bellman errors from the point of view of kernel support vector regression and Gaussian processes. [Engel *et al.*, 2003] consider a generative model for the cost-to-go function which is based on Gaussian processes. [Xu *et al.*, 2007] introduce the non-parametric version of the LSTD-(λ) method, a numerically stable approach to do policy evaluation. [Farahmand *et al.*, 2009] study regularized versions of some popular policy evaluation methods and perform the sample complexity analysis for one policy evaluation step. Unfortunately approximate policy iteration schemes have no convergence guarantees in parametric settings, and these difficulties remain in non-parametric variations. It is consequently difficult to characterize the computational effort or sample complexity required to produce a good approximation (if this is at all possible) with such approaches. More importantly, the practical performance or viability (given that many rounds of regression will typically be called for) of these methods is not clear.

Local averaging. Another idea has been to use kernel-based local averaging ideas to approximate the solution of an MDP with that of a simpler variation on a sampled state space [Ormoneit and

Sen, 2002; Ormoneit and Glynn, 2002; Barreto *et al.*, 2011]. However, convergence rates for local averaging methods are exponential in the dimension of the problem state space. As in our setting, [Dietterich and Wang, 2002] construct kernel-based cost-to-go function approximations.These are subsequently plugged in to various ad hoc optimization-based ADP formulations. While their methods are closest in spirit to ours, they do not employ any regularization. This suggests potentially poor sample complexity, and, indeed, they do not provide theoretical justification or sample complexity results. Similarly, [Ernst *et al.*, 2005] replace the local averaging procedure used for regression by [Ormoneit and Sen, 2002] with non-parametric regression procedures such as the tree-based learning methods. This is done again without any theoretical justification.

Feature selection via ℓ_1-penalty (parametric). Closely related to our work, [Kolter and Ng, 2009] and [Petrik *et al.*, 2010] consider modifying the approximate linear program with an ℓ_1-regularization term to encourage sparse approximations in the span of a large, but necessarily *tractable* set of features. Along these lines, [Pazis and Parr, 2011] discuss a non-parametric method that explicitly restricts the smoothness of the value function. However, sample complexity results for this method are not provided and it appears unsuitable for high-dimensional problems (such as, for instance, the queuing problem we consider in our experiments). In contrast to this line of work, our approach will allow for approximations in a potentially *full-dimensional* approximation architecture that capable of an *exact* representation of the value function, with a constraint on an appropriate ℓ_2-norm of the weight vector to provide regularization.

This paper presents what we believe is a practical, non-parametric ADP algorithm that enjoys non-trivial approximation and sample complexity guarantees. In particular, we establish both theoretically and computationally that our proposal can serve as a viable replacement to state-of-the-art parametric ADP algorithms based on linear programming, freeing the designer from carefully specifying an approximation architecture. In greater detail, we make the following contributions:

- **A new mathematical programming formulation.** We rigorously develop a kernel-based variation of the 'smoothed' approximate LP (SALP) approach to ADP proposed by [Desai *et al.*, 2011]. The resulting mathematical program, which we dub the regularized smoothed approximate LP (RSALP), is distinct from simply substituting the local averaging approximation above in the SALP formulation. We develop a companion active set method that is capable of solving this mathematical program rapidly and with limited memory requirements.

- **Theoretical guarantees.** [1] Our algorithm can be interpreted as solving an approximate linear program in a (potentially infinite dimensional) Hilbert space. We provide a probabilistic upper bound on the approximation error of the algorithm relative to the best possible approximation one may compute in this space subject to a regularization constraint. We show that the number of samples grows polynomially as a function of a regularization parameter. As this regularization parameter is allowed to grow, so does the set of permissible approximations, eventually permitting an exact approximation.

  The sampling requirements for our method are independent of the dimension of the approximation architecture. Instead, they grow with the desired complexity of the approximation. This result can be seen as the 'right' generalization of the prior parametric approximate LP approaches of [de Farias and Van Roy, 2003; de Farias and Van Roy, 2004; Desai *et al.*, 2011], where, in contrast, sample complexity grows with the dimension of the approximating architecture.

- **A computational study.** To study the efficacy of our approach, we consider an MDP arising from a challenging queueing network scheduling problem. We demonstrate that our method yields significant improvements over tailored heuristics and parametric ADP methods, all while using a generic high-dimensional approximation architecture. In particular, these results suggest the possibility of solving a challenging high-dimensional MDP using an entirely generic approach.

The method we propose in the paper relies on the knowledge of the transition probabilities of the MDP. Further we require the number of possible next states for any state action pair to be finite and small. Such restriction are shared by all linear programming based ADP methods. In spite of this, these methods have been applied widely on problems such as controlled queueing network [de Farias and Van Roy, 2004], network revenue management [Adelman, 2007] and patient scheduling in diagnostic facilities [Patrick *et al.*, 2008].

The organization of the paper is as follows: In Section 2.2, we formulate an infinite dimensional

[1] These guarantees come under assumption of being able to sample from a certain idealized distribution. This is a common requirement in the analysis of ADP algorithms that enjoy approximation guarantees for general MDPs [de Farias and Van Roy, 2004; Van Roy, 2006; Desai *et al.*, 2011].

LP for our problem, and present an effective approximate solution approach. Section 2.3 provides theoretical guarantees for the quality of the approximations computed via our non-parametric algorithm. Theorem 1 in that Section provides our main guarantee. In Section 2.4, we provide an active set method that can be used to efficiently solve the required quadratic optimization problem central to our approach while respecting practical memory constraints. We also establishing the correctness of our active set approach. Section 2.5 describes a numerical study for a criss-cross queueing system benchmarking our approach against approximate linear programming approaches and tailor made heuristics. Section 2.6 concludes.

2.2 Formulation

2.2.1 Preliminaries

Consider a discrete time Markov decision process with finite state space $\mathcal{S}$ and finite action space $\mathcal{A}$. We denote by x_t and a_t respectively, the state and action at time t. For notational simplicity, and without loss of generality, we assume that all actions are permissible at any given state. We assume time-homogeneous Markovian dynamics: conditioned on being at state x and taking action a, the system transitions to state x' with probability $p(x, x', a)$ independent of the past. A policy is a map $\mu\colon \mathcal{S} \to \mathcal{A}$, so that

$$J^{\mu}(x) \triangleq \mathsf{E}_{x,\mu}\left[\sum_{t=0}^{\infty} \alpha^t g_{x_t,a_t}\right]$$

represents the expected (discounted, infinite horizon) cost-to-go under policy μ starting at state x, with the discount factor $\alpha \in (0,1)$. Letting Π denote the set of all policies, our goal is to find an optimal policy μ^* such that $\mu^* \in \operatorname{argmax}_{\mu\in\Pi} J^{\mu}(x)$ for all $x \in \mathcal{S}$ (it is well known that such a policy exists). We denote the optimal cost-to-go function by $J^* \triangleq J^{\mu^*}$. An optimal policy μ^* can be recovered as a 'greedy' policy with respect to J^*,

$$\mu^*(x) \in \operatorname*{argmin}_{a\in\mathcal{A}} \; g_{x,a} + \alpha \mathsf{E}_{x,a}[J^*(X')],$$

where we define the expectation $\mathsf{E}_{x,a}[f(X')] \triangleq \sum_{x'\in\mathcal{S}} p(x,x',a) f(x')$, for all functions $f\colon \mathcal{S} \to \mathbb{R}$ on the state space.

Since in practical applications the state space $\mathcal{S}$ is often intractably large, exact computation of J^* is untenable. ADP algorithms are principally tasked with computing approximations to J^* of the

form $J^*(x) \approx z^\top \Phi(x) \triangleq \tilde{J}(x)$, where $\Phi \colon \mathcal{S} \to \mathbb{R}^m$ is referred to as an 'approximation architecture' or a basis and must be provided as input to the ADP algorithm. The ADP algorithm computes a 'weight' vector $z \in \mathbb{R}^m$. One then employs a policy that is greedy with respect to the corresponding approximation $\tilde{J}$.

2.2.2 Primal Formulation

The approach we propose is based on the LP formulation of dynamic programming. It was observed by [Manne, 1960] that the optimal cost-to-go J^* can be obtained by solving the following linear program:

$$\begin{aligned} \text{maximize} \quad & \nu^\top J \\ \text{subject to} \quad & J(x) \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[J(X')], \quad \forall\ x \in \mathcal{S},\ a \in \mathcal{A}, \\ & J \in \mathbb{R}^{\mathcal{S}}, \end{aligned} \tag{2.1}$$

for any strictly positive state-relevance weight vector $\nu \in \mathbb{R}^{\mathcal{S}}_+$. Motivated by this, a series of ADP algorithms [Schweitzer and Seidman, 1985; de Farias and Van Roy, 2003; Desai *et al.*, 2011] have been proposed that compute a weight vector z by solving an appropriate modification of (2.1). In particular, [Desai *et al.*, 2011] propose solving the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{x \in \mathcal{S}} \nu_x z^\top \Phi(x) - \kappa \sum_{x \in \mathcal{S}} \pi_x s_x \\ \text{subject to} \quad & z^\top \Phi(x) \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[z^\top \Phi(X')] + s_x, \quad \forall\ x \in \mathcal{S},\ a \in \mathcal{A}, \\ & z \in \mathbb{R}^m,\ s \in \mathbb{R}^{\mathcal{S}}_+. \end{aligned} \tag{2.2}$$

Here $\kappa > 0$ is a penalty parameter and $\pi \in \mathbb{R}^{\mathcal{S}}_+$ is a strictly positive distribution on $\mathcal{S}$. In considering the above program, notice that if one insisted that the slack variables s were precisely 0, the program (3.11) would be identical to (2.1), with the additional restriction to value function approximations of the form $J(x) = z^\top \Phi(x)$. This case is known as the approximate linear program (ALP), and was first proposed by [Schweitzer and Seidman, 1985]. [de Farias and Van Roy, 2003] provided a pioneering analysis that, stated loosely, showed

$$\|J^* - {z^*}^\top \Phi\|_{1,\nu} \leq \frac{2}{1-\alpha} \inf_z \ \|J^* - z^\top \Phi\|_\infty,$$

for an optimal solution z^* to the ALP. [Desai *et al.*, 2011] showed that these bounds could be improved upon by 'smoothing' the constraints of the ALP, i.e., permitting positive slacks. The

resulting program (3.11) is called the smoothed approximate linear program (SALP). The ALP constraints impose the restriction that $TJ \geq J$. This ensures that the optimal solution lower bounds the optimal value function at each state. But at the cost of providing a lower bound, the ALP might do a poor job of approximating J^* due to presence of constraints associated with certain states that are rarely visited. SALP prevents this by allowing violations of Bellman constrains. Frequently visited states are given more weight to prevent large Bellman errors at those states. For a more detailed interpretation of SALP we refer you to Section 3 of [Desai *et al.*, 2011]. In both instances, in the general case of a large state space, one must solve a 'sampled' version of the above program.

Now, consider allowing Φ to map from $\mathcal{S}$ to a general (potentially infinite dimensional) Hilbert space $\mathcal{H}$. We use bold letters to denote elements in the Hilbert space $\mathcal{H}$, e.g., the weight vector is denoted by $\mathbf{z} \in \mathcal{H}$. We further suppress the dependence on Φ and denote the elements of $\mathcal{H}$ corresponding to their counterparts in $\mathcal{S}$ by bold letters. Hence, for example, $\mathbf{x} \triangleq \Phi(x)$ and $\mathbf{X} \triangleq \Phi(X)$.Denote the image of the state space under the map Φ by $\mathcal{X} \triangleq \Phi(\mathcal{S})$; $\mathcal{X} \subset \mathcal{H}$. The analogous value function approximation in this case would be given by

$$\tilde{J}_{\mathbf{z},b}(x) \triangleq \langle \mathbf{x}, \mathbf{z} \rangle + b = \langle \Phi(x), \mathbf{z} \rangle + b, \tag{2.3}$$

where b is a scalar offset corresponding to a constant basis function.[2] The following generalization of (3.11) — which we dub the regularized SALP (RSALP) — then essentially suggests itself:

$$\begin{aligned} \text{maximize} \quad & \sum_{x \in \mathcal{S}} \nu_x \langle \mathbf{x}, \mathbf{z} \rangle + b - \kappa \sum_{x \in \mathcal{S}} \pi_x s_x - \frac{\Gamma}{2} \langle \mathbf{z}, \mathbf{z} \rangle \\ \text{subject to} \quad & \langle \mathbf{x}, \mathbf{z} \rangle + b \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[\langle \mathbf{X}', \mathbf{z} \rangle + b] + s_x, \quad \forall\ x \in \mathcal{S},\ a \in \mathcal{A}, \\ & \mathbf{z} \in \mathcal{H},\ b \in \mathbb{R},\ s \in \mathbb{R}^{\mathcal{S}}_{+}. \end{aligned} \tag{2.4}$$

The only new ingredient in the program above is the fact that we regularize the weight vector $\mathbf{z}$ using the parameter $\Gamma > 0$. Penalizing the objective of (2.4) according to the square of the norm $\|\mathbf{z}\|_{\mathcal{H}} \triangleq \sqrt{\langle \mathbf{z}, \mathbf{z} \rangle}$ anticipates that we will eventually resort to sampling in solving thisprogram. In a sampled setting, regularization is necessary to avoid over-fitting and, in particular, to construct an approximation that generalizes well to unsampled states. This regularization, which plays a

[2]separating the scalar offset b from the linear term parameterized by $\mathbf{z}$ will permit us to regularize these two quantities differently in the sequel.

crucial role both in theory and practice, is easily missed if one directly 'plugs in' a local averaging approximation in place of $z^\top \Phi(x)$, as is the case in the earlier work of [Dietterich and Wang, 2002], or a more general non-parametric approximation as in the work of [Ernst *et al.*, 2005].

Since the RSALP of (2.4) can be interpreted as a regularized stochastic optimization problem, one may hope to solve it via its sample average approximation. To this end, define the likelihood ratio $w_x \triangleq \nu_x/\pi_x$, and let $\hat{\mathcal{S}} \subset \mathcal{S}$ be a set of N states sampled independently according to the distribution π. The sample average approximation of (2.4) is then

$$\begin{aligned} \text{maximize} \quad & \frac{1}{N}\sum_{x\in\hat{\mathcal{S}}} w_x \langle \mathbf{x}, \mathbf{z}\rangle + b - \frac{\kappa}{N}\sum_{x\in\hat{\mathcal{S}}} s_x - \frac{\Gamma}{2}\langle \mathbf{z}, \mathbf{z}\rangle \\ \text{subject to} \quad & \langle \mathbf{x}, \mathbf{z}\rangle + b \le g_{a,x} + \alpha \mathsf{E}_{x,a}[\langle \mathbf{X}', \mathbf{z}\rangle + b] + s_x, \quad \forall\ x \in \hat{\mathcal{S}},\ a \in \mathcal{A}, \\ & \mathbf{z} \in \mathcal{H},\ b \in \mathbb{R},\ s \in \mathbb{R}_+^{\hat{\mathcal{S}}}. \end{aligned} \tag{2.5}$$

We call this program the sampled RSALP. Even if $|\hat{\mathcal{S}}|$ were small, it is still not clear that this program can be solved effectively. We will, in fact, solve the dual to this problem.

2.2.3 Dual Formulation

We begin by establishing some notation. Let $\mathcal{N}_{x,a} \triangleq \{x\} \cup \{x' \in \mathcal{S} \ : p(x, x', a) > 0\}$ denote the set of states that can be reached starting at a state $x \in \mathcal{S}$ given an action $a \in \mathcal{A}$. For any states $x, x' \in \mathcal{S}$ and action $a \in \mathcal{A}$, define $q_{x,x',a} \triangleq \mathbf{1}_{\{x=x'\}} - \alpha p(x, x', a)$. Now, define the symmetric positive semi-definite matrix $Q \in \mathbb{R}^{(\hat{\mathcal{S}}\times\mathcal{A})\times(\hat{\mathcal{S}}\times\mathcal{A})}$ according to

$$Q(x, a, x', a') \triangleq \sum_{\substack{y\in\mathcal{N}_{x,a} \\ y'\in\mathcal{N}_{x',a'}}} q_{x,y,a} q_{x',y',a'} \langle \mathbf{y}, \mathbf{y}'\rangle, \tag{2.6}$$

the vector $R \in \mathbb{R}^{\hat{\mathcal{S}}\times\mathcal{A}}$ according to

$$R(x, a) \triangleq \Gamma g_{x,a} - \frac{1}{N}\sum_{\substack{x'\in\hat{\mathcal{S}} \\ y\in\mathcal{N}_{x,a}}} w_{x'} q_{x,y,a} \langle \mathbf{y}, \mathbf{x}'\rangle, \tag{2.7}$$

and the scalar S as

$$S \triangleq -\sum_{x\in\hat{\mathcal{S}}}\sum_{y\in\hat{\mathcal{S}}} w_x w_y \langle \mathbf{x}, \mathbf{y}\rangle.$$

Notice that Q, R and S depend only on inner products in $\mathcal{X}$ (and other easily computable quantities). The dual to (2.5) is then given by:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\lambda^\top Q\lambda + R^\top\lambda + S \\ \text{subject to} \quad & \sum_{a\in\mathcal{A}} \lambda_{x,a} \leq \frac{\kappa}{N}, \qquad \forall \;\; x \in \hat{\mathcal{S}}, \\ & \sum_{\substack{x\in\hat{\mathcal{S}} \\ a\in\mathcal{A}}} \lambda_{x,a} = \frac{1}{1-\alpha}, \\ & \lambda \in \mathbb{R}_+^{\hat{\mathcal{S}}\times\mathcal{A}}. \end{aligned} \tag{2.8}$$

Assuming that Q, R and S can be easily computed, this finite dimensional quadratic program, *is* tractable – its size is polynomial in the number of sampled states. We may recover a primal solution (i.e., the weight vector $\mathbf{z}^*$) from an optimal dual solution:

Proposition 1. *Programs* (2.5) *and* (2.8) *have equal (finite) optimal values. The optimal solution to* (2.8) *is attained at some* λ^*. *The optimal solution to* (2.5) *is attained at some* (z^*, s^*, b^*) *with*

$$\mathbf{z}^* = \frac{1}{\Gamma}\left[\frac{1}{N}\sum_{x\in\hat{\mathcal{S}}} w_x\mathbf{x} - \sum_{x\in\hat{\mathcal{S}},a\in\mathcal{A}} \lambda^*_{x,a}\Big(\mathbf{x} - \alpha\mathsf{E}_{x,a}[\mathbf{X}']\Big)\right]. \tag{2.9}$$

The proof of Proposition 1 is presented in Appendix A.1. Having solved this program, we may, using Proposition 1, recover our approximate cost-to-go function $\tilde{J}(x) = \langle\mathbf{z}^*, \mathbf{x}\rangle + b^*$ as

$$\tilde{J}(x) = \frac{1}{\Gamma}\left[-\sum_{y\in\hat{\mathcal{S}},a\in\mathcal{A}} \lambda^*_{y,a}\big(\langle\mathbf{y},\mathbf{x}\rangle - \alpha\mathsf{E}_{y,a}[\langle\mathbf{X}',\mathbf{x}\rangle]\big) + \frac{1}{N}\sum_{y\in\hat{\mathcal{S}}} w_y\langle\mathbf{y},\mathbf{x}\rangle\right] + b^*. \tag{2.10}$$

A policy greedy with respect to $\tilde{J}$ is not affected by constant translations, hence in (2.10), the value of b^* can be set to be zero arbitrarily. Again note that given λ^*, evaluation of $\tilde{J}$ only involves inner products in $\mathcal{X}$.

2.2.4 Kernels

As pointed out earlier, the sampled RSALP is potentially difficult to work with. Proposition 1 establishes that solving this program (via its dual) is a computation that scales polynomially in N so that it can be solved efficiently provided inner products in $\mathcal{H}$ can be evaluated cheaply. Alternatively, we may have arrived at a similar conclusion by observing that in any optimal solution

to (2.5), we must have that $\mathbf{z}^* \in \text{span}\left\{\mathbf{x} \ : \ x \in \hat{\mathcal{S}} \cup \mathcal{N}(\hat{\mathcal{S}})\right\}$, where $\mathcal{N}(\hat{\mathcal{S}})$ denotes the set of states that can be reached from the sampled states of $\hat{\mathcal{S}}$ in a single transition. Then, one can restrict the feasible region of (2.5) to this subspace. In either approach, we observe that one need not necessarily have explicitly characterized the feature map $\Phi(\cdot)$; knowing $\langle \Phi(x), \Phi(y) \rangle$ for all $x, y \in \mathcal{S}$ would suffice and so the algorithm designer can focus on simply specifying these inner products. This leads us to what is popularly referred to as the 'kernel trick' which we discuss next without the assumption that $\mathcal{S}$ is necessarily a finite set.

A kernel is a map $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$; we will call such a kernel positive definite if for any finite collection of elements $\{x_i\}_{1 \leq i \leq n}$ in $\mathcal{S}$, the Gram matrix $G \in \mathbb{R}^{n \times n}$ defined by $G_{ij} \triangleq K(x_i, x_j)$ is symmetric and positive semi-definite. Given such a kernel, we are assured of the existence of a Hilbert space $\mathcal{H}$ and a map $\Phi \colon \mathcal{S} \to \mathcal{H}$ such that[3] $\langle \Phi(x), \Phi(y) \rangle = K(x, y)$. The kernel trick then allows us to implicitly work with this space $\mathcal{H}$ by replacing inner products of the form $\langle \mathbf{x}, \mathbf{y} \rangle \triangleq \langle \Phi(x), \Phi(y) \rangle$ in (2.6), (2.7), and (2.10) by $K(x, y)$. Of course, the quality of the approximation one may produce depends on $\mathcal{H}$ and consequently on the kernel employed.

One case of special interest is where $\mathcal{S}$ is finite and the Gram matrix corresponding to this set is positive definite. A kernel satisfying this condition is known as *full-dimensional* or *full-rank*. In this case, we may take $\mathcal{H}$ to be the $|\mathcal{S}|$-dimensional Euclidean space. Working with such a kernel would imply working with an approximation architecture where *any* function $J \colon \mathcal{S} \to \mathbb{R}$ can be exactly expressed in the form $J(x) = \langle \Phi(x), \mathbf{z} \rangle$, for some weight vector $\mathbf{z} \in \mathcal{H}$.

There are a number of kernels one can specify on the state space; we list a few examples here for the case where $\mathcal{S} \subset \mathbb{R}^n$. The polynomial kernel is defined according to

$$K(x, y) \triangleq (1 + x^\top y)^d.$$

A corresponding Hilbert space $\mathcal{H}$ is given by the span of all monomials of degree up to d on $\mathbb{R}^n$. A Gaussian kernel is defined according to

$$K(x, y) \triangleq \exp\left(-\frac{\|x - y\|^2}{\sigma}\right).$$

The Gaussian kernel is known to be full-dimensional (see, e.g., Theorem 2.18, [Scholkopf and Smola, 2001]), so that employing such a kernel in our setting would correspond to working with an infinite

[3] A canonical such space is the so-called 'reproducing kernel' Hilbert space, by the Moore-Aronszajn theorem. For certain sets $\mathcal{S}$, Mercer's theorem provides another important construction of such a Hilbert space.

dimensional approximation architecture. A thorough exposition on this topic, along with many more examples can be found in the text of [Scholkopf and Smola, 2001].

2.2.5 Overall Procedure

Our development thus far suggests the following non-parametric algorithm that requires as input a kernel K, and the distributions π and ν. It also requires that we set the parameters κ and Γ and the number of samples N.

1. Sample N states from a distribution π.

2. Solve (2.8) with

$$Q(x,a,x',a') \triangleq \sum_{y\in\mathcal{N}_{x,a}} \sum_{y'\in\mathcal{N}_{x',a'}} q_{x,y,a} q_{x',y',a'} K(y,y'), \tag{2.11}$$

   and

$$R(x,a) \triangleq \left(\Gamma g_{x,a} - \frac{1}{N} \sum_{y\in\hat{\mathcal{S}}} \sum_{x'\in\mathcal{N}_x,a} w_y q_{x,x',a} K(y,x') \right). \tag{2.12}$$

   The value of S may be set to be zero.

3. Let λ^* be the dual optimal. Define an approximate value function by

$$\tilde{J}(x) \triangleq \frac{1}{\Gamma} \left[\frac{1}{N} \sum_{y\in\hat{\mathcal{S}}} w_y K(x,y) - \sum_{y\in\hat{\mathcal{S}},a\in\mathcal{A}} \lambda^*_{y,a} (K(x,y) - \alpha \mathsf{E}_{y,a}[K(X',x)]) \right]. \tag{2.13}$$

In the next section we will develop theory to characterize the sample complexity of our procedure and the approximation quality it provides. This theory will highlight the roles of the kernel K and the sampling distributions used.

2.3 Theory

2.3.1 Overview

Recall that we are employing an approximation $\tilde{J}_{\mathbf{z},b}$ of the form

$$\tilde{J}_{\mathbf{z},b} = \langle \mathbf{x}, \mathbf{z} \rangle + b$$

parameterized by the weight vector $\mathbf{z}$ and the offset parameter b. For the purpose of establishing theoretical guarantees, in this section, we will look at the following variation of the RSALP of (2.4):

$$\begin{aligned} \text{maximize} \quad & \sum_{x\in\mathcal{S}} \nu_x \langle \mathbf{x}, \mathbf{z}\rangle + b - \kappa \sum_{x\in\mathcal{S}} \pi_x s_x \\ \text{subject to} \quad & \langle \mathbf{x}, \mathbf{z}\rangle + b \leq g_{a,x} + \alpha \mathsf{E}_{x,a}[\langle \mathbf{X}', \mathbf{z}\rangle + b] + s_x, \quad \forall\ x \in \mathcal{S},\ a \in \mathcal{A}, \\ & \|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B, \\ & \mathbf{z} \in \mathcal{H},\ b \in \mathbb{R},\ s \in \mathbb{R}_+^{\mathcal{S}}. \end{aligned} \tag{2.14}$$

Here, rather than regularizing by penalizing according to the weight vector $\mathbf{z}$ in the objective as in (2.4), we regularize by restricting the size of the weight vector as a constraint. This regularization constraint is parameterized by the scalar $C \geq 0$. It is easy to see that (2.14) is equivalent to the original problem (2.4), in the following sense: for any $\Gamma > 0$, there exists a $C \geq 0$ so that for all B sufficiently large, (2.4) and (2.14) have the same optimal solutions and value. Let $C(\Gamma)$ by any such value of C, corresponding to a particular Γ.

Now let

$$\mathcal{C}\big(C(\Gamma), B\big) \triangleq \{(\mathbf{z}, b) \in \mathcal{H} \times \mathbb{R} \ :\ \|\mathbf{z}\|_{\mathcal{H}} \leq C(\Gamma),\ |b| \leq B\}.$$

The best approximation to J^* within this set has ℓ_∞-approximation error

$$\inf_{(\mathbf{z},b)\in\mathcal{C}\left(C(\Gamma),B\right)} \|J^* - \tilde{J}_{\mathbf{z},b}\|_\infty. \tag{2.15}$$

Now observe that if $\text{span}\{\mathbf{x} \ :\ x \in \mathcal{S}\} \subset \mathcal{H}$ has dimension $|\mathcal{S}|$ (i.e., if the kernel is full-dimensional), there exists a $\tilde{\mathbf{z}} \in \mathcal{H}$ satisfying $\tilde{J}_{\tilde{\mathbf{z}},0} = J^*$. Consequently, for $C(\Gamma)$ sufficiently large and any value of $B \geq 0$, we have that $\tilde{\mathbf{z}} \in \mathcal{C}\big(C(\Gamma), B\big)$ — the approximation error in (2.15) reduces to zero. More generally, as $C(\Gamma)$ and B grow large, we expect the approximation error in (2.15) to monotonically decrease; the precise rate of this decrease will depend, of course, on the feature map Φ, which in turn will depend on the kernel we employ.

Loosely speaking, in this section, we will demonstrate that:

1. For any given $C(\Gamma)$ and B, exact solution of the RSALP (2.14) will produce an approximation with ℓ_∞-error comparable to the optimal ℓ_∞-error possible for approximations $\tilde{J}_{\mathbf{z},b}$ with $\|\mathbf{z}\|_{\mathcal{H}} \leq C(\Gamma)$ and $b \leq B$.

2. Under a certain set of idealized assumptions, solving a *sampled* variation of the RSALP (2.14) with a number of samples $N\big(C(\Gamma), B\big)$ that scales gracefully with $C(\Gamma)$, B, and other natural parameters, produces a near optimal solution to the RSALP with high probability. These sample complexity bounds will *not* depend on the dimension of the approximation architecture $\mathcal{H}$.

Since $C(\Gamma)$ and B are parameters of the algorithm designer's choosing, these results will effectively show that with the ability to use a larger number of samples, the designer may choose larger values of $C(\Gamma)$ and B and consequently produce approximations of improving quality. Under mild assumptions on the kernel, the approximation error can be made arbitrarily small in this fashion.

2.3.2 Preliminaries and an Idealized Program

Before proceeding with our analysis, we introduce some preliminary notation. For a vector $x \in \mathbb{R}^n$, and a 'weight' vector $v \in \mathbb{R}^n_+$, we denote by $\|x\|_v \triangleq \sum_{i=1}^n v_i|x_i|$ the weighted 1-norm of x. Let $\Psi = \{\psi \in \mathbb{R}^{\mathcal{S}} \ : \ \psi \geq \mathbf{1}\}$, be the set of all functions on the state space bounded from below by 1. For any $\psi \in \Psi$, let us define the weighted ∞-norm $\|\cdot\|_{\infty,1/\psi}$ by

$$||J||_{\infty,1/\psi} \triangleq \max_{x\in\mathcal{S}} \ |J(x)|/\psi(x).$$

Our use of such weighted norms will allow us to emphasize approximation error differently across the state space. Further, we define

$$\beta(\psi) \triangleq \max_{x\in\mathcal{X},\ a\in\mathcal{A}} \ \frac{\sum_{x'} p(x,x',a)\psi(x')}{\psi(x)}.$$

For a given ψ, $\beta(\psi)$ gives us the worst-case expected gain of the Lyapunov function ψ for any state action pair (x,a).

Finally, we define an idealized sampling distribution. Let ν be an arbitrary distribution over $\mathcal{S}$ and denote by μ^* and P_{μ^*} an optimal policy and the transition probability matrix corresponding to this policy, respectively. We define a distribution over $\mathcal{S}$, $\pi_{\mu^*,\nu}$ according to

$$\pi^\top_{\mu^*,\nu} \triangleq (1-\alpha)\nu^\top(I-\alpha P_{\mu^*})^{-1} = (1-\alpha)\sum_{t=0}^{\infty}\alpha^t\nu^\top P^t_{\mu^*}. \tag{2.16}$$

This idealized distribution will play the role of the sampling distribution π in the sequel.

It is notationally convenient for us to introduce the Bellman operator defined by

$$(TJ)(x) \triangleq \min_{a \in \mathcal{A}} \left[g(x,a) + \alpha \mathsf{E}_{x,a}[J(X')] \right],$$

for all $x \in \mathcal{S}$ and $J \colon \mathcal{S} \to \mathbb{R}$. As before, let $\hat{\mathcal{S}}$ be a set of N states drawn independently at random from $\mathcal{S}$; here we pick a specific sampling distribution $\pi = \pi_{\mu^*,\nu}$. Given the definition of $\tilde{J}_{\mathbf{z},b}$ in (2.3), the 'idealized' sampled program we consider is:

$$\begin{aligned} \text{maximize} \quad & \nu^\top \tilde{J}_{\mathbf{z},b} - \frac{2}{1-\alpha} \frac{1}{N} \sum_{x \in \hat{\mathcal{S}}} s_x \\ \text{subject to} \quad & \tilde{J}_{\mathbf{z},b}(x) \leq T\tilde{J}_{\mathbf{z},b}(x) + s_x, \quad \forall\, x \in \hat{\mathcal{S}}, \\ & \|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B, \\ & \mathbf{z} \in \mathcal{H},\ b \in \mathbb{R},\ s \in \mathbb{R}_+^{\hat{\mathcal{S}}}. \end{aligned} \tag{2.17}$$

This program is a sampled variant of (2.14) and is closely related to the sampled RSALP (2.5) introduced in the last section. Before proceeding with an analysis of the quality of approximation afforded by solving this program, we discuss its connection with the sampled RSALP (2.5) of the previous section:

1. The distributions ν and π: We allow for an arbitrary distribution ν, but given this distribution require that $\pi = \pi_{\mu^*,\nu}$. In particular, the distribution ν might be chosen as the empirical distribution corresponding to N independent draws from $\mathcal{S}$ under some measure; one would then draw N independent samples under $\pi = \pi_{\mu^*,\nu}$ to construct the second term in the objective. In a sense, this is the only 'serious' idealized assumption we make here. Given the broad nature of the class of problems considered it is hard to expect meaningful results without an assumption of this sort, and indeed much of the antecedent literature considering parametric LP based approaches makes such an assumption. When the sampling distribution π is a close approximation to $\pi_{\mu^*,\nu}$ (say, the likelihood ratio between the two distributions is bounded), then it is possible to provide natural extensions to our results that account for how close the sampling distribution used is to the idealized sampling distribution.

2. Regularization: We regularize the weight vector $\mathbf{z}$ with an explicit constraint on $\|\mathbf{z}\|_{\mathcal{H}}$; this permits a transparent analysis and is equivalent to placing the 'soft' regularization term $\frac{\Gamma}{2}\|\mathbf{z}\|_{\mathcal{H}}$ in the objective. In particular, the notation $C(\Gamma)$ makes this equivalence explicit.

In addition, we place a separate upper bound on the offset term B. This constraint is not binding if $B > 2C \max_{x \in \mathcal{S}} \|\mathbf{x}\|_{\mathcal{H}} + \|g\|_\infty/(1-\alpha)$ but again, permits a transparent analysis of the dependence of the sample complexity of solving our idealized program on the offset permitted by the approximation architecture.

3. The choice of κ: The smoothing parameter κ can be interpreted as yet another regularization parameter, here on the (one-sided) Bellman error permitted for our approximation. Our idealized program chooses a specific value for this smoothing parameter, in line with that chosen by [Desai *et al.*, 2011]. Our experiments will use the same value of κ; experience with the parametric SALP suggests that this is a good choice of the parameter in practice.

2.3.3 The Approximation Guarantee

Let $(\hat{\mathbf{z}}, \hat{b})$ be an optimal solution to the idealized sampled program (2.17) and let $K \triangleq \max_{x \in \mathcal{S}} \|\mathbf{x}\|_{\mathcal{H}}$. Further define,

$$\Xi(C, B, K, \delta) \triangleq \left(4CK(1+\alpha) + 4B(1-\alpha) + 2\|g\|_\infty\right)\left(1 + \sqrt{\frac{1}{2}\ln(1/\delta)}\right).$$

Notice that $\Xi(C, B, K, \delta)^2$ scales as the square of C, K, and B and further is $O(\ln 1/\delta)$. The following result will constitute our main approximation guarantee:

Theorem 1. *For any $\epsilon, \delta > 0$, let $N \geq \Xi(C, B, K, \delta)^2/\epsilon^2$. If $(\hat{\mathbf{z}}, \hat{b})$ is an optimal solution to (2.17), then with probability at least $1 - \delta - \delta^4$,*

$$\left\| J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}} \right\|_{1,\nu} \leq \inf_{\|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B,\ \psi \in \Psi} \|J^* - \tilde{J}_{\mathbf{z},b}\|_{\infty,1/\psi} \left(\nu^\top \psi + \frac{2(\pi_{\mu^*,\nu}^\top \psi)(\alpha\beta(\psi)+1)}{1-\alpha}\right) + \frac{4\epsilon}{1-\alpha}. \tag{2.18}$$

The remainder of this section is dedicated to parsing and interpreting this guarantee:

1. Optimal approximation error: Taking ψ to be the vector of all ones, we see that the right side of the approximation guarantee (2.18) is bounded above by

$$\frac{3+\alpha}{1-\alpha} \inf_{\|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B} \|J^* - \tilde{J}_{\mathbf{z},b}\|_\infty + \frac{4\epsilon}{1-\alpha}.$$

In particular, ignoring the ϵ-dependent error term, we see that the quality of approximation provided by $(\hat{\mathbf{z}}, \hat{b})$ is essentially within a constant multiple (at most $(3+\alpha)/(1-\alpha)$) of the

optimal (in the sense of ℓ_∞-error) approximation to J^* possible using a weight vector $\mathbf{z}$ and offsets b permitted by the regularization constraints. This is a 'structural' error term that will persist even if one were permitted to draw an arbitrarily large number of samples. It is analogous to the approximation results produced in parametric settings with the important distinction that *one allows comparisons to approximations in potentially full-dimensional sets* which might, as we have argued earlier, be substantially superior.

2. Dimension independent sampling error: In addition to the structural error above, one incurs an additional additive 'sampling' error that scales like $4\epsilon/(1-\alpha)$. The result demonstrates that

$$\epsilon = O\left(\frac{(CK+B)\sqrt{\ln 1/\delta}}{\sqrt{N}}\right)$$

This is an important contrast with existing parametric sample complexity guarantees. In particular, *existing guarantees typically depend on the dimension of the space spanned by the basis function architecture* $\{\mathbf{x} : x \in \mathcal{S}\}$. Here, this space may be full-dimensional, so that such a dependence would translate to a vacuous guarantee. Instead we see that the dependence on the approximation architecture is through the constants C, K and B. Of these K can, for many interesting kernels be upper bounded by a constant that is independent of $|\mathcal{S}|$, while C and B are user-selected regularization bounds. Put another way, the guarantee allows for arbitrary 'simple' (in the sense of $\|\mathbf{z}\|_{\mathcal{H}}$ being small) approximations in a rich feature space as opposed to restricting us to some a priori fixed, low-dimensional feature space. This yields some intuition for why we expect the approach to perform well even with a relatively general choice of kernel.

3. A non-parametric interpretation: As we have argued earlier, in the event that span $\{\mathbf{x} : x \in \mathcal{S}\}$ is full-dimensional, there exists a choice of C and B for which the optimal approximation error is, in fact, zero. A large number of kernels would guarantee such feature maps. More generally, as C and B grow large, $\inf_{\|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B} \|J^* - \tilde{J}_{\mathbf{z},b}\|_\infty$ will decrease. In order to maintain the sampling error constant, one would then need to increase N at a rate that is roughly $\Omega((CK+B)^2)$. In summary, by increasing the number of samples in the sampled program, we can (by increasing C and B appropriately) hope to compute approximations of increasing quality, approaching an *exact* approximation. In the event that the feature space

permits good approximations that are 'simple' for that space (i.e., with $\|\mathbf{z}\|_{\mathcal{H}}$ small), the approach is capable of producing good approximations for a tractable number of samples.

2.3.4 Proof of Theorem 1

We prepare the ground for the proof by developing appropriate uniform concentration guarantees for appropriate function classes.

2.3.4.1 Uniform Concentration Bounds

We begin with defining the empirical Rademacher complexity of a class of functions $\mathcal{F}$ from $\mathcal{S}$ to R as

$$\hat{R}_n(\mathcal{F}) = \mathsf{E}\left[\sup_{f\in\mathcal{F}} \frac{2}{n}\sum_{i=1}^{n} \sigma_i f(X_i) \middle| X_1, X_2, \ldots, X_n\right],$$

where σ_i are i.i.d. random variable that take value 1 with probability $1/2$ and -1 with probability $1/2$. The X_i are i.i.d. $\mathcal{S}$-valued random variables drawn with the distribution π. We denote by $R_n(\mathcal{F}) \triangleq \mathsf{E}\hat{R}_n(\mathcal{F})$ the Rademacher complexity of $\mathcal{F}$.

We begin with the following abstract sample complexity result: let $\mathcal{F}$ be a class of functions mapping $\mathcal{S}$ to $\mathbb{R}$ that are uniformly bounded by some constant $\overline{B}$. Moreover denote for any function $f \in \mathcal{F}$, the empirical expectation $\hat{\mathsf{E}}_n f(X) \triangleq \frac{1}{n}\sum_{i=1}^{n} f(X_i)$, where the X_i are i.i.d. random draws from $\mathcal{S}$ as above. We then have the following sample complexity result:

Lemma 1.

$$\mathsf{P}\left(\sup_{f\in\mathcal{F}} \mathsf{E}f(X) - \hat{\mathsf{E}}_n f(X) \geq R_n(\mathcal{F}) + \sqrt{\frac{2\overline{B}^2 \ln(1/\delta)}{n}}\right) \leq \delta.$$

This result is standard; for completeness, the proof may be found in Appendix A.2. Next, we establish the Rademacher complexity of a specific class of functions. Fixing a policy μ, consider then the set of functions mapping $\mathcal{S}$ to $\mathbb{R}$ defined according to:

$$\mathcal{F}_{\mathcal{S},\mu} \triangleq \left\{x \mapsto \langle \mathbf{x}, \mathbf{z}\rangle - \alpha \mathsf{E}_{x,\mu(x)}[\langle \mathbf{X}', \mathbf{z}\rangle] \ : \ \|\mathbf{z}\|_{\mathcal{H}} \leq C\right\}.$$

We have:

Lemma 2. *For any policy μ,*

$$R_n\left(\mathcal{F}_{\mathcal{S},\mu}\right) \leq \frac{2CK(1+\alpha)}{\sqrt{n}}.$$

Proof. Observe that, due to triangle inequality

$$\|\mathbf{x} - \alpha \mathsf{E}_{x,\mu(x)}[\mathbf{X}']\|_{\mathcal{H}} \leq \|\mathbf{x}\|_{\mathcal{H}} + \alpha \mathsf{E}_{x,\mu(x)}[\|\mathbf{X}'\|_{\mathcal{H}}] \leq K(1+\alpha),$$

for all $x \in \mathcal{S}$. Now, let X_i be i.i.d. samples in $\mathcal{S}$ and $\mathbf{X}_i$ be the corresponding elements in $\mathcal{H}$,

$$\begin{aligned}
\hat{R}_n\left(\mathcal{F}_{\mathcal{S},\mu}\right) &= \frac{2}{n}\mathsf{E}\left[\sup_{z:\ \|z\|_{\mathcal{H}} \leq C} \left\langle \sum_i \sigma_i \left(\mathbf{X}_i - \alpha \mathsf{E}_{X_i,\mu(X_i)}[\mathbf{X}']\right), z \right\rangle \middle| X_1, \ldots, X_n \right] \\
&\leq \frac{2}{n}\mathsf{E}\left[\sup_{z:\ \|z\|_{\mathcal{H}} \leq C} \left\| \sum_i \sigma_i \left(\mathbf{X}_i - \alpha \mathsf{E}_{X_i,\mu(X_i)}[\mathbf{X}']\right) \right\|_{\mathcal{H}} \|z\|_{\mathcal{H}} \middle| X_1, \ldots, X_n \right] \\
&= \frac{2C}{n}\mathsf{E}\left[\left\| \sum_i \sigma_i \left(\mathbf{X}_i - \alpha \mathsf{E}_{X_i,\mu(X_i)}[\mathbf{X}']\right) \right\|_{\mathcal{H}} \middle| X_1, \ldots, X_n \right] \\
&\leq \frac{2C}{n}\sqrt{\sum_i \|\mathbf{X}_i - \alpha \mathsf{E}_{X_i,\mu(X_i)}[\mathbf{X}']\|_{\mathcal{H}}^2} \\
&\leq \frac{2CK(1+\alpha)}{\sqrt{n}}.
\end{aligned}$$

■

Now, consider the class of functions mapping $\mathcal{S}$ to $\mathbb{R}$, defined according to:

$$\overline{\mathcal{F}}_{\mathcal{S},\mu} \triangleq \left\{ x \mapsto \left(\tilde{J}_{\mathbf{z},b}(x) - (T_\mu \tilde{J}_{\mathbf{z},b})(x) \right)^+ \ : \ \|\mathbf{z}\|_{\mathcal{H}} \leq C,\ |b| \leq B \right\}.$$

Now, $\overline{\mathcal{F}}_{\mathcal{S},\mu} = \phi\left(\mathcal{F}_{\mathcal{S},\mu} + (1-\alpha)\mathcal{F}_B - g_\mu\right)$, where $\phi \triangleq (\cdot)^+$ and $\mathcal{F}_B \triangleq \{x \mapsto b \ : \ |b| \leq B\}$. It is easy to show that $R_n(\mathcal{F}_B) \leq 2B/\sqrt{n}$, so that with the previous lemma, the results of [Theorem 12, parts 4 and 5 [Bartlett and Mendelson, 2002]] allow us to conclude

Corollary 1.

$$R_n(\overline{\mathcal{F}}_{\mathcal{S},\mu}) \leq \frac{4CK(1+\alpha) + 4B(1-\alpha) + 2\|g_\mu\|_\infty}{\sqrt{n}} \triangleq \frac{\overline{C}}{\sqrt{n}}.$$

Now, define

$$\overline{\mathcal{F}}_{\mathcal{S}} \triangleq \left\{ x \mapsto \left(\tilde{J}_{\mathbf{z},b}(x) - (T \tilde{J}_{\mathbf{z},b})(x) \right)^+ \ : \ \|\mathbf{z}\|_{\mathcal{H}} \leq C, |b| \leq B \right\}.$$

We have:

Lemma 3. *For every $f \in \overline{\mathcal{F}}_{\mathcal{S}}$ we have that $\|f\|_\infty \leq \overline{C}/2$. Moreover,*

$$\mathsf{P}\left(\hat{\mathsf{E}}_N f(X) - \mathsf{E} f(X) \geq \epsilon \right) \leq \delta^4,$$

provided $N \geq \Xi(C, B, K, \delta)^2/\epsilon^2$.

The first claim above follows from routine algebra and the Cauchy-Schwartz inequality; the second is Hoeffding's inequality. Corollary 1, Lemma 1, and the first part of Lemma 3 yields the following sample complexity result:

Theorem 2. *Provided* $N \geq \Xi(C, B, K, \delta)^2/\epsilon^2$ *, we have*

$$\mathsf{P}\left(\sup_{f \in \bar{\mathcal{F}}_{\mathcal{S},\mu}} \mathsf{E}f(X) - \hat{\mathsf{E}}_N f(X) \geq \epsilon\right) \leq \delta.$$

Theorem 2 will constitute a crucial sample complexity bound for our main result; we now proceed with the proof of Theorem 1.

2.3.4.2 Proof of Theorem 1

Let $(\hat{\mathbf{z}}, \hat{b}, \hat{s})$ be the optimal solution to the sampled program (2.17). Define

$$\hat{s}_{\mu^*} \triangleq (\tilde{J}_{\hat{\mathbf{z}},\hat{b}} - T_{\mu^*}\tilde{J}_{\hat{\mathbf{z}},\hat{b}})^+.$$

Observe that we may assume, without loss of generality, that

$$\hat{s} = (\tilde{J}_{\hat{\mathbf{z}},\hat{b}} - T\tilde{J}_{\hat{\mathbf{z}},\hat{b}})^+,$$

so that $\hat{s} \geq \hat{s}_{\mu^*}$. Now, by definition, $\tilde{J}_{\hat{\mathbf{z}},\hat{b}} \leq T_{\mu^*}\tilde{J}_{\hat{\mathbf{z}},\hat{b}} + \hat{s}_{\mu^*}$, so that by Lemma 2 of [Desai *et al.*, 2011], we have that

$$\tilde{J}_{\hat{\mathbf{z}},\hat{b}} \leq J^* + \Delta^*\hat{s}_{\mu^*},$$

where $\Delta^* = (I - \alpha P_{\mu^*})^{-1}$. Let $\hat{\pi}_{\mu^*,\nu}$ be the empirical distribution obtained by sampling N states according to $\pi_{\mu^*,\nu}$. Now let $\mathbf{z}, b$ satisfying $\|\mathbf{z}\|_{\mathcal{H}} \leq C, |b| \leq B$ be given, and define $s_{\mathbf{z},b} \triangleq (T\tilde{J}_{\mathbf{z},b} - \tilde{J}_{\mathbf{z},b})^+$. Then, $(\mathbf{z}, b, s_{\mathbf{z},b})$ constitute a feasible solution to (2.17). Finally, let $\psi \in \Psi$ be arbitrary. We

then have with probability at least $1 - \delta - \delta^4$,

$$\begin{aligned}
\|J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\|_{1,\nu} &= \|J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}} + \Delta^* \hat{s}_{\mu^*} - \Delta^* \hat{s}_{\mu^*}\|_{1,\nu} \\
&\leq \|J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}} + \Delta^* \hat{s}_{\mu^*}\|_{1,\nu} + \|\Delta^* \hat{s}_{\mu^*}\|_{1,\nu} \\
&= \nu^\top \left(J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\right) + 2\nu^\top \Delta^* \hat{s}_{\mu^*} \\
&= \nu^\top \left(J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\right) + \frac{2}{1-\alpha} \pi_{\mu^*,\nu}^\top \hat{s}_{\mu^*} \\
&\leq \nu^\top \left(J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\right) + \frac{2}{1-\alpha} \hat{\pi}_{\mu^*,\nu}^\top \hat{s}_{\mu^*} + \frac{2\epsilon}{1-\alpha} \\
&\leq \nu^\top \left(J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\right) + \frac{2}{1-\alpha} \hat{\pi}_{\mu^*,\nu}^\top \hat{s} + \frac{2\epsilon}{1-\alpha} \\
&\leq \nu^\top \left(J^* - \tilde{J}_{\mathbf{z},b}\right) + \frac{2}{1-\alpha} \hat{\pi}_{\mu^*,\nu}^\top s_{\mathbf{z},b} + \frac{2\epsilon}{1-\alpha} \\
&\leq \nu^\top \left(J^* - \tilde{J}_{\mathbf{z},b}\right) + \frac{2}{1-\alpha} \pi_{\mu^*,\nu}^\top s_{\mathbf{z},b} + \frac{4\epsilon}{1-\alpha}. \\
&\leq (\nu^\top \psi)\|J^* - \tilde{J}_{\mathbf{z},b}\|_{\infty,1/\psi} + \frac{2}{1-\alpha}(\pi_{\mu^*,\nu}^\top \psi)\|T\tilde{J}_{\mathbf{z},b} - \tilde{J}_{\mathbf{z},b}\|_{\infty,1/\psi} + \frac{4\epsilon}{1-\alpha}.
\end{aligned} \tag{2.19}$$

The second equality follows by our observation that $\tilde{J}_{\hat{\mathbf{z}},\hat{b}} \leq J^* + \Delta^* \hat{s}_{\mu^*}$ and since $\Delta^* \hat{s}_{\mu^*} \geq 0$. The second inequality above holds with probability at least $1 - \delta$ by virtue of Theorem 2 and the fact that $\hat{s}_{\mu^*} \in \overline{\mathcal{F}}_{\mathcal{S},\mu^*}$. The subsequent inequality follows from our observation that $\hat{s} \geq \hat{s}_{\mu^*}$. The fourth inequality follows from the assumed optimality of $(\hat{\mathbf{z}}, \hat{b}, \hat{s})$ for the sampled program (2.17) and the feasibility of $(\mathbf{z}, b, s_{\mathbf{z},b})$ for the same. The fifth inequality holds with probability $1 - \delta^4$ and follows from the Hoeffding bound in Lemma 3 since $s_{\mathbf{z},b} \in \overline{\mathcal{F}}_{\mathcal{S}}$. The final in equality follows from the observation that for any $s \in \mathbb{R}^{\mathcal{S}}, \psi \in \Psi$ a and probability vector ν, $\nu^\top s \leq (\nu^\top \psi)\|s\|_{\infty,1/\psi}$.

Now the proof of Theorem 2 of [Desai *et al.*, 2011] establishes that for any $\psi \in \Psi$ and $J \in \mathbb{R}^{\mathcal{S}}$, we have,

$$\|TJ - J\|_{\infty,1/\psi} \leq (1 + \alpha\beta(\psi))\|J^* - J\|_{\infty,1/\psi}.$$

Applied to (2.19), this yields

$$\|J^* - \tilde{J}_{\hat{\mathbf{z}},\hat{b}}\|_{1,\nu} \leq \|J^* - \tilde{J}_{\mathbf{z},b}\|_{\infty,1/\psi} \left(\nu^\top \psi + \frac{2(\pi_{\mu^*,\nu}^\top \psi)(\alpha\beta(\psi) + 1)}{1-\alpha}\right) + \frac{4\epsilon}{1-\alpha}.$$

Since our choice of $\mathbf{z}, b$ was arbitrary (beyond satisfying $\|\mathbf{z}\|_{\mathcal{H}} \leq C, |b| \leq B$), the result follows.

2.4 Numerical Procedure

This section outlines an efficient numerical scheme we use to solve the regularized SALP. In particular, we would like to solve the sampled dual problem (2.8), introduced in Section **??**, in order to find an approximation to the optimal cost-to-go function. This approach requires solving a quadratic program (QP) with $N \times A$ variables, where $N \triangleq |\hat{\mathcal{S}}|$ is the number of sampled states and $A \triangleq |\mathcal{A}|$ is the number of possible actions. Furthermore, constructing the coefficient matrices Q and R for (2.8) requires $O(N^2A^2H^2)$ arithmetic operations, where H is the maximum number of states that can be reached from an arbitrary state-action pair, i.e.,

$$H \triangleq \max_{(x,a)\in\mathcal{S}\times\mathcal{A}} \left|\{x' \in \mathcal{S} \ : \ p(x,x',a) > 0\}\right|.$$

These computationally expensive steps may prevent scaling up solution of the QP to a large number of samples. Also, an off-the-shelf QP solver will typically store the matrix Q in memory, so that the memory required to solve our QP with an off-the-shelf solver effectively scales like $O(N^2A^2)$.

In this section, we develop an iterative scheme to solve the program (2.8) that, by exploiting problem structure, enjoys low computational complexity per iteration and attractive memory requirements. Our scheme is an active set method in the vein of the approaches used by [Osuna *et al.*, 1997] and [Joachims, 1999], among others, for solving large SVM problems. The broad steps of the scheme are as follows:

1. At the tth iteration, a subset $\mathcal{B} \subset \hat{\mathcal{S}} \times \mathcal{A}$ of the decision variables of (2.8) – the 'active set' – is chosen. Only variables in this set may be changed in a given iteration; these changes must preserve feasibility of the new solution that results. Our algorithm will limit the size of the active set to two variables, i.e., $|\mathcal{B}| = 2$. The methodology for choosing this active set is crucial and will be described in the sequel.

2. Given the subset $\mathcal{B}$, we solve (2.8) for $\lambda^{(t)}$, where all variables except those in $\mathcal{B}$ must remain unchanged. In other words, $\lambda_{x,a}^{(t)} \triangleq \lambda_{x,a}^{(t-1)}$ for all $(x,a) \notin \mathcal{B}$. This entails the solution of a QP with only $|\mathcal{B}|$ decision variables. In our case, we will be able to solve this problem in closed form.

3. Finally, if the prior step does not result in a decrease in objective value we conclude that we are at an optimal solution; Proposition 2 establishes that this is, in fact, a correct termination

criterion.

In the following section, we will establish an approach for selecting the active set at each iteration and show how Step 2 above can be solved while maintaining feasibility at each iteration. We will establish that steps one and two together need no more that $O(NA \log NA)$ arithmetic operations and comparisons, and moreover that the memory requirement for our procedure scales like $O(NA)$. Finally, we will establish that our termination criterion is correct: in particular, if no descent direction of cardinality at most two exists at a given feasible point, we must be at an optimal solution.

2.4.1 Subset Selection

The first step in the active set method is to choose the subset $\mathcal{B} \subset \hat{\mathcal{S}} \times \mathcal{A}$ of decision variables to optimize over. Given the convex objective in (2.8), if the prior iteration of the algorithm is at a sub-optimal point $\lambda \triangleq \lambda^{(t-1)}$, then there exists a direction $d \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}$ such that $\lambda + \epsilon d$ is feasible with a lower objective value for $\epsilon > 0$ sufficiently small. To select a subset to optimize over, we look for such a descent direction d of low cardinality $\|d_0\|_0 \leq q$, i.e., a vector d that is zero on all but at most q components. If such a direction can be found, then we can use the set of non-zero indices of d as our set $\mathcal{B}$.

This problem of finding a 'sparse' descent direction d can be posed as

$$\begin{aligned}
\text{minimize} \quad & h(\lambda)^\top d \\
\text{subject to} \quad & \sum_{a \in \mathcal{A}} d_{x,a} \leq 0, \quad \forall\ x \in \hat{\mathcal{S}} \text{ with } \sum_{x \in \mathcal{A}} \lambda_{x,a} = \frac{\kappa}{N}, \\
& d_{x,a} \geq 0, \qquad \forall\ (x,a) \in \hat{\mathcal{S}} \times \mathcal{A} \text{ with } \lambda_{x,a} = 0, \\
& \sum_{\substack{x \in \hat{\mathcal{S}} \\ a \in \mathcal{A}}} d_{x,a} = 0, \\
& ||d||_0 \leq q, \\
& ||d||_\infty \leq 1, \\
& d \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}.
\end{aligned} \tag{2.20}$$

Here, $h(\lambda) \triangleq Q\lambda + R$ is the gradient of the objective of (2.8) at a feasible point λ, thus the objective $h(\lambda)^\top d$ seeks to find a direction d of steepest descent. The first three constraints ensure that d is a feasible direction. The constraint $\|d\|_0 \leq q$ is added to ensure that the direction is of cardinality

at most q. Finally, the constraint $\|d\|_\infty \leq 1$ is added to ensure that the program is bounded, and may be viewed as normalizing the scale of the direction d.

The program (2.20) is, in general, a challenging mixed integer program because of the cardinality constraint. [Joachims, 1999] discusses an algorithm to solve a similar problem of finding a low cardinality descent direction in an SVM classification setting. Their problem can be easily solved provided that the cardinality q is even, however no such solution seems to exist for our case. However, in our case, when $q = 2$, there is a tractable way to solve (2.20). We will restrict attention to this special case, i.e., consider only descent directions of cardinality two. In Section 2.4.3, we will establish that this is, in fact, sufficient: if the prior iterate λ is sub-optimal, then there will exist a direction of descent of cardinality two.

To begin, define the sets

$$\mathcal{P}_1 \triangleq \left\{(x,a) \in \hat{\mathcal{S}} \times \mathcal{A} \ : \ \lambda_{a,x} = 0\right\}, \qquad \mathcal{P}_2 \triangleq \left\{x \in \hat{\mathcal{S}} \ : \ \sum_{a \in \mathcal{A}} \lambda_{x,a} = \frac{\kappa}{N}\right\}.$$

Consider the following procedure:

1. Sort the set of indices $\hat{\mathcal{S}} \times \mathcal{A}$ according to their corresponding component values in the gradient vector $h(\lambda)$. Call this sorted list $\mathcal{L}_1$.

2. Denote by (x_1, a_1) the largest element of $\mathcal{L}_1$ such that $(x_1, a_1) \notin \mathcal{P}_1$, and denote by (x_2, a_2) the smallest element of $\mathcal{L}_1$ such that $x_2 \notin \mathcal{P}_2$. Add the tuple (x_1, a_1, x_2, a_2) to the list $\mathcal{L}_2$.

3. Consider all $x \in \mathcal{P}_2$. For each such x, denote by (x, a_1) the largest element of $\mathcal{L}_1$ such that $(x, a_1) \notin \mathcal{P}_1$, and denote by (x, a_2) the smallest element of $\mathcal{L}_1$. Add each tuple (x, a_1, x, a_2) to $\mathcal{L}_2$.

4. Choose the element $(x_1^*, a_1^*, x_2^*, a_2^*)$ of $\mathcal{L}_2$ that optimizes

$$\min_{(x_1,a_1,x_2,a_2) \in \mathcal{L}_2} \ h(\lambda)_{x_2,a_2} - h(\lambda)_{x_1,a_1}. \tag{2.21}$$

   Set $d_{x_1^*,a_1^*} = -1$, $d_{x_2^*,a_2^*} = 1$, and all other components of d to zero..

This procedure finds a direction of maximum descent of cardinality two by examining considering candidate index pairs (x_1, a_1, x_2, a_2) for which $h(\lambda)_{x_2,a_1} - h(\lambda)_{x_1,a_1}$ is minimal. Instead of considering at all N^2A^2 such pairs, the routine selectively checks only pairs with describe feasible directions

with respect to the constraints of (2.20). Step 2 considers all feasible pairs with different states x, while Step 3 considers all pairs with the same state. It is thus easy to see that the output of this procedure is an optimal solution to (2.20), i.e., a direction of steepest descent of cardinality two. Further, if the value of the minimal objective (2.21) determined by this procedure is non-negative, then no descent direction of cardinality two exists, and the algorithm terminates.

In terms of computational complexity, this subset selection step requires us to first compute the gradient of the objective function $h(\lambda) \triangleq Q\lambda + R$. If the gradient is known at the first iteration, then we can update it at each step by generating two columns of Q, since λ only changes at two co-ordinates. Hence, the gradient calculation can be performed in $O(NA)$ time, and with $O(NA)$ storage (since it is not necessary to store Q). For Step 1 of the subset selection procedure, the component indices must be sorted in the order given by the gradient $h(\lambda)$. This operation requires computational effort of the order $O(NA \log NA)$. With the sorted indices, the effort required in the remaining steps to find the steepest direction is via the outlined procedure is $O(NA)$. Thus, our subset selection step requires a total of $O(NA \log NA)$ arithmetic operations and comparisons.

The initialization of the gradient requires $O(N^2A^2)$ effort. In the cases where such a computation is prohibitive, one can think of many approaches to approximately evaluate the initial gradient. For example, if we use the Gaussian kernel, the matrix Q will have most of its large entries close to the diagonal. In this case, instead of the expensive evaluation of Q, we can only evaluate the entries $Q(x, a, x', a')$ where $x = x'$ and set the rest of them to zero. This block diagonal approximation might be used to initialize the gradient. Another approach is to sample from the distribution induced by λ to approximately evaluate $Q\lambda$. As the algorithm makes progress it evaluates new columns of Q. With a bit of book-keeping one could get rid of the errors associated with the gradient initialization. The convergence properties of the active set method with approximate initialization is an issue not tackled in this paper.

2.4.2 QP Sub-problem

Given a prior iterate $\lambda^{(t-1)}$, and a subset $\mathcal{B} \triangleq \{(x_1, a_1), (x_2, a_2)\}$ of decision variable components of cardinality two as computed in Section 2.4.1, we have the restricted optimization problem

$$\begin{aligned}
\text{minimize} \quad & \sum_{(x,a)\in\mathcal{B}} \sum_{(x',a')\in\mathcal{B}} \lambda_{x',a'} Q(x,a,x',a') \lambda_{x,a} \\
& + \sum_{(x,a)\in\mathcal{B}} \lambda_{x,a} \left(R(x,a) + 2 \sum_{(x',a')\notin\mathcal{B}} Q(x,a,x',a') \lambda^{(t-1)}_{x',a'} \right) \\
\text{subject to} \quad & \sum_{a:\ (x,a)\in\mathcal{B}} \lambda_{x,a} \leq \frac{\kappa}{N} - \sum_{a:\ (x,a)\notin\mathcal{B}} \lambda^{(t-1)}_{x,a}, \qquad \forall\ x \in \{x_1, x_2\} \\
& \sum_{(x,a)\in\mathcal{B}} \lambda_{x,a} = \frac{1}{1-\alpha} - \sum_{(x,a)\notin\mathcal{B}} \lambda^{(t-1)}_{x,a}, \\
& \lambda \in \mathbb{R}^{\mathcal{B}}_{+}.
\end{aligned} \tag{2.22}$$

This sub-problem has small dimension. In fact, the equality constraint implies that $\lambda_{x_1,a_1} + \lambda_{x_2,a_2}$ is constant, hence, the problem is in fact a one-dimensional QP that can be solved in closed form. Further, to construct this QP, two columns of Q are required to be generated. This requires computation effort of order $O(NA)$.

Note that the subset $\mathcal{B}$ is chosen so that it is guaranteed to contain a descent direction, according to the procedure in Section 2.4.1. Then, the solution of (2.20) will produce an iterate $\lambda^{(t)}$ that is feasible for the original problem (2.22) and has lower objective value than the prior iterate $\lambda^{(t-1)}$.

2.4.3 Correctness of Termination Condition

The following proposition establishes the correctness of our active set method: if the prior iterate $\lambda \triangleq \lambda^{(t-1)}$ is sub-optimal, then there must exist a direction of descent of cardinality two. Our iterative procedure will therefore improve the solution, and will only terminate when global optimality is achieved.

Proposition 2. *If $\lambda \in \mathbb{R}^{\hat{\mathcal{S}}\times\mathcal{A}}$ is feasible but sub-optimal for* (2.8) *then, there exists a descent direction of cardinality two.*

The proof of Proposition 2 requires the following lemma:

Lemma 4. *Suppose $x, y \in \mathbb{R}^n$ are vectors such that $\mathbf{1}^\top x = 0$ and $x^\top y < 0$. Then there exist co-ordinates $\{i, j\}$, such that $y_i < y_j$, $x_i > 0$, and $x_j < 0$.*

Proof. Define the index sets $S^+ \triangleq \{i \ : \ x_i > 0\}$ and $S^- \triangleq \{i \ : \ x_i < 0\}$. Under the given hypotheses, both sets are non-empty. Using the fact that $\mathbf{1}^\top x = 0$, define

$$Z \triangleq \sum_{i \in S^+} x_i = \sum_{i \in S^-} (x_i)^-,$$

where $(x_i)^- \triangleq -\min(x_i, 0)$ is the negative part of the scalar x_i. Observe that $Z > 0$. Further, since $x^\top y < 0$,

$$\frac{1}{Z} \sum_{i \in S^+} x_i y_i < \frac{1}{Z} \sum_{i \in S^-} (x_i)^- y_i.$$

Since the weighted average of y over S^- is more than its weighted average over S^+, we can pick an element in S^+, i and an element of S^-, j such that $y_i < y_j$. ■

Proof of Proposition 2. If λ is sub-optimal, since (2.8) is a convex quadratic program, there will exist some vector $d \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}$ is a feasible descent direction at λ. Let $g \triangleq h(\lambda)$ be the gradient at that point. We must have that $g^\top d < 0$, so that it is a descent direction, and that d satisfies the first three constraints of (2.20), so that it is a feasible direction.

Define

$$\mathcal{T} \triangleq \left\{ x \in \hat{\mathcal{S}} \ : \ \sum_{a \in \mathcal{A}} \lambda_{x,a} = \frac{\kappa}{N}, \ \max_{a \in \mathcal{A}} \ d_{x,a} > 0, \ \min_{a \in \mathcal{A}} \ d_{x,a} < 0 \right\}, \quad \mathcal{P}_x \triangleq \{a \in \mathcal{A} \ : \ d_{x,a} \neq 0\}.$$

First, consider the case of $\mathcal{T} = \emptyset$. In this case, for all x such that $\sum_{a \in \mathcal{A}} \lambda_{x,a} = \kappa/N$, we have $d_{x,a} \leq 0$. Since d is a descent direction, we have $g^\top d < 0$. Lemma 4 can be applied to get a pair (x_1, a_1) and (x_2, a_2) such that $d_{x_1,a_1} > 0$ and $d_{x_2,a_2} < 0$, with $g_{x_1,a_1} < g_{x_2,a_2}$. Further x_1 is such that $\sum_{a \in \mathcal{A}} \lambda_{x_1,a} < \kappa/N$ and (x_2, a_2) is such that $\lambda_{x_2,a_2} > 0$. These conditions ensure that if (x_1, a_1) and (x_2, a_2) are increased and decreased respectively in the same amount, then the objective is decreased. And hence we obtain a descent direction of cardinality 2. By assuming that $\mathcal{T} = \emptyset$, we have avoided the corner case of not being able to increase d_{x_1,a_1} due to $\sum_{a \in \mathcal{A}} \lambda_{x_1,a} = \kappa/N$.

For $\mathcal{T} \neq \emptyset$, without loss of generality, assume that $|\mathcal{P}_x| = 2$ for each $x \in \mathcal{T}$, i.e., d has exactly two non-zero components corresponding to the state x. This is justified at the end of the proof. Denote these indices by (x, a_+) and (x, a_-), so that $d_{x,a_+} > 0$ and $d_{x,a_-} < 0$. From the first constraint of (2.20), we must have that $d_{x,a_+} \leq (d_{x,a_-})^-$.

There are two cases:

(i) Suppose $g_{x,a_+} < g_{x,a_-}$, for some $x \in \mathcal{T}$.

Then, we can define a descent direction $\tilde{d} \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}$ of cardinality two by setting $\tilde{d}_{x,a_+} = 1$, $\tilde{d}_{x,a_-} = -1$, and all other components of $\tilde{d}$ to zero.

(ii) Suppose that $g_{x,a_+} \geq g_{x,a_-}$, for all $x \in \mathcal{T}$.

For all $x \in \mathcal{T}$, define $\hat{d}_x \triangleq (d_{x,a_-})^- - d_{x,a_+} \geq 0$. Then, the fact that $\sum_{x,a} d_{x,a} = 0$ implies that

$$\sum_{\substack{x \notin \mathcal{T} \\ a \in \mathcal{A}}} d_{x,a} - \sum_{x \in \mathcal{T}} \hat{d}_x = 0. \tag{2.23}$$

At the same time, for all $x \in \mathcal{T}$, we have that

$$d_{x,a_+} g_{x,a_+} + d_{x,a_-} g_{x,a_-} = -\hat{d}_x g_{x,a_-} + d_{x,a_+}(g_{x,a_+} - g_{x,a_-}) \geq -\hat{d}_x g_{x,a_-}.$$

Then, since d is a descent direction, we have that

$$\sum_{\substack{x \notin \mathcal{T} \\ a \in \mathcal{A}}} d_{x,a} g_{x,a} - \sum_{x \in \mathcal{T}} \hat{d}_x g_{x,a_-} < 0. \tag{2.24}$$

Now, define the vector $\tilde{d} \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}$ by

$$\tilde{d}_{x,a} = \begin{cases} d_{x,a} & \text{if } x \notin \mathcal{T}, \\ -\hat{d}_x & \text{if } x \in \mathcal{T} \text{ and } (x,a) = (x,a_-), \\ 0 & \text{otherwise.} \end{cases}$$

Applying Lemma 4 to (2.23) and (2.24), there must be a pair of indices (x_1, a_1) and (x_2, a_2) such that $\tilde{d}_{x_1,a_1} > 0$, $\tilde{d}_{x_2,a_2} < 0$ and $g_{x_1,a_1} < g_{x_2,a_2}$. For such (x_1, a_1) and (x_2, a_2) we have a descent direction where we can increase λ_{x_1,a_1} and decrease λ_{x_2,a_2} by the same amount and get a decrease in the objective. Note that since $\tilde{d}_{x_1,a_1} > 0$, we have that $d_{x_1,a_1} > 0$ and $x_1 \notin \mathcal{T}$, hence $\sum_a \lambda_{x_1,a} < \kappa/N$. Also, by construction, (x_2, a_2) is chosen such that $d_{x_2,a_2} < 0$, implying that $\lambda_{x_2,a_2} > 0$. Thus the specified direction is also feasible, and we have a feasible descent direction of cardinality two.

Finally, to complete the proof, consider the case where there are some $x \in \mathcal{T}$ with $|\mathcal{P}_x| > 2$, i.e.,

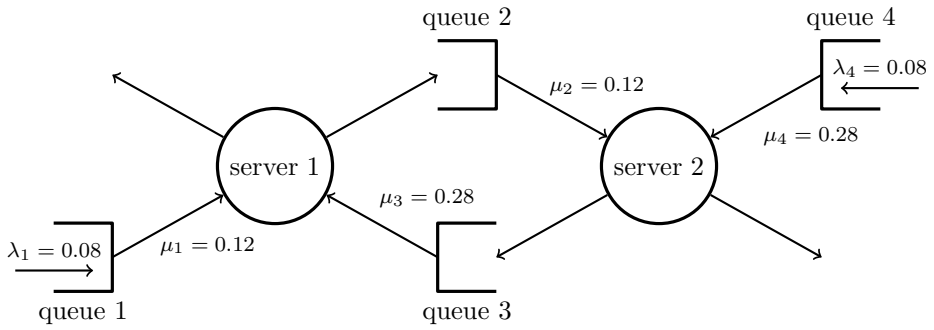

Figure 2.1: The queueing network example.

d has more than two non-zero components corresponding to the state x. For each $x \in \mathcal{T}$, define

$$\mathcal{A}_x^+ \triangleq \{a \in \mathcal{A} \ : \ d_{x,a} > 0\}, \qquad \mathcal{A}_x^- \triangleq \{a \in \mathcal{A} \ : \ d_{x,a} < 0\},$$

$$a_1 \in \operatorname*{argmin}_{a \in \mathcal{A}_x^+} g_{x,a}, \qquad a_2 \in \operatorname*{argmax}_{a \in \mathcal{A}_x^-} g_{x,a}.$$

Define a new direction $\tilde{d} \in \mathbb{R}^{\hat{\mathcal{S}} \times \mathcal{A}}$ by

$$\tilde{d}_{x,a} = \begin{cases} \sum_{a' \in \mathcal{A}_x^+} d_{x,a'} & \text{if } x \in \mathcal{T} \text{ and } (x,a) = (x,a_1), \\ \sum_{a' \in \mathcal{A}_x^-} d_{x,a'} & \text{if } x \in \mathcal{T} \text{ and } (x,a) = (x,a_2), \\ d_{x,a} & \text{otherwise.} \end{cases}$$

It is easy to verify that $\tilde{d}$ is also a feasible descent direction. Furthermore, $\tilde{d}$ has only two non-zero components corresponding to each start $x \in \mathcal{T}$. ■

2.5 Experiments

This section considers the problem of controlling the queuing network illustrated in Figure 2.1, with the objective of minimizing long run average delay. There are two 'flows' in this network: the first through server 1 followed by server 2 (with buffering at queues 1 and 2, respectively), and the second through server 2 followed by server 1 (with buffering at queues 4 and 3, respectively). Here, all inter-arrival and service times are exponential with rate parameters summarized in Figure 2.1.

This specific network has been studied by [de Farias and Van Roy, 2003; Chen and Meyn, 1998; Kumar and Seidman, 1990], for example, and closely related networks have been studied by [Harrison and Wein, 1989; Kushner and Martins, 1996; Martins *et al.*, 1996; Kumar and Muthuraman,

2004]. It is widely considered to be a challenging control problem. As such, a lot of thought has been invested in designing scheduling policies with networks of this type in mind. Our goal in this section will be two fold. First, we will show that the RSALP, used 'out-of-the-box' with a generic kernel, can match or surpass the performance of tailor made heuristics and state of the art parametric ADP methods. Second, we will show that the RSALP can be solved efficiently.

2.5.1 MDP Formulation

Although the control problem at hand is nominally a continuous time problem, it is routinely converted into a discrete time problem via a standard uniformization device; see [Moallemi *et al.*, 2008], for instance, for an explicit such example. In the equivalent discrete time problem, at most a single event can occur in a given epoch, corresponding either to the arrival of a job at queues 1 or 4, or the arrival of a service token for one of the four queues with probability proportional to the corresponding rates. The state of the system is described by the number of jobs in each of the four queues, so that $\mathcal{S} \triangleq \mathbb{Z}_+^4$, whereas the action space $\mathcal{A}$ consists of four potential actions each corresponding to a matching between servers and queues. We take the single period cost as the total number of jobs in the system, so that $g_{x,a} = \|x\|_1$; note that minimizing the average number of jobs in the system is equivalent to minimizing average delay by Little's law. Finally, we take $\alpha = 0.9$ as our discount factor.[4]

2.5.2 Approaches

The following scheduling approaches were considered for the queueing problem:

RSALP (this paper). We solve (2.8) using the active set method outlined in Section 2.4, taking as our kernel the standard Gaussian radial basis function kernel $K(x,y) \triangleq \exp\left(-\|x-y\|_2^2/h\right)$, with the bandwidth parameter[5] $h \triangleq 100$. Note that this implicitly corresponds to an full-dimensional basis function architecture. Since the idealized sampling distribution, $\pi_{\mu^*,\nu}$ is unavailable to us, we use in its place the geometric distribution

$$\pi(x) \triangleq (1-\zeta)^4 \zeta^{\|x\|_1}, \tag{2.25}$$

[4]Note that while we will solve a problem with a discounted infinite horizon optimality criterion, we will report long run average costs. This is in keeping with [Desai *et al.*, 2011] and [de Farias and Van Roy, 2003]

[5]The sensitivity of our results to this bandwidth parameter appears minimal.

with the sampling parameter ζ set at 0.9. This choice mimics that of [de Farias and Van Roy, 2003]. The regularization parameter Γ was chosen via a line-search;[6] we report results for $\Gamma \triangleq 10^{-6}$. In accordance with the theory we set the constraint violation parameter $\kappa \triangleq 2/(1-\alpha)$, as suggested by the analysis of Section 2.3, as well as by [Desai *et al.*, 2011] for the SALP.

SALP [Desai et al., 2011]. The SALP formulation (3.11), is, as discussed earlier, the parametric counterpart to the RSALP. It may be viewed as a generalization of the ALP approach proposed by [de Farias and Van Roy, 2003] and has been demonstrated to provide substantial performance benefits relative to the ALP approach. Our choice of parameters for the SALP mirrors those for the RSALP to the extent possible, so as to allow for an 'apples-to-apples' comparison. Thus, as earlier, we solve the sample average approximation of this program using the same sampling distribution π as in (2.25), and we set $\kappa \triangleq 2/(1-\alpha)$. Being a parametric approach, one needs to specify an appropriate approximation architecture. Approximation architectures that span polynomials are known to work well for queueing problem. We use the basis functions used by [de Farias and Van Roy, 2003] for a similar problem modeled directly in discrete time. In particular, we use all monomials with degree at most 3, which we will call the *cubic* basis, as our approximation architectures.

Longest Queue First (generic). This is a simple heuristic approach: at any given time, a server chooses to work on the longest queue from among those it can service.

Max-Weight [Tassiulas and Ephremides, 1992]. Max-Weight is a well known scheduling heuristic for queueing networks. The policy is obtained as the greedy policy with respect to a value function approximation of the form

$$\tilde{J}_{MW}(x) \triangleq \sum_{i=1}^{4} |x_i|^{1+\epsilon},$$

given a parameter $\epsilon > 0$. This policy has been extensively studied and shown to have a number of good properties, for example, being throughput optimal [Dai and Lin, 2005] and offering good performance for critically loaded settings [Stolyar, 2004]. Via a line-search we chose to use $\epsilon \triangleq 1.5$ as the exponent for our experiments.

[6] Again, performance does not appear to be very sensitive to Γ, so that a crude line-search appears to suffice. Specifically, we tried values of the form $\Gamma = 10^k$, for k between -12 and 2.

policy	**performance**									
Longest Queue	8.09									
Max-Weight	6.55									
sample size	1000		3000		5000		10000		15000	
SALP, cubic basis	7.19	(1.76)	7.89	(1.76)	6.94	(1.15)	6.63	(0.92)	6.58	(1.12)
RSALP, Gaussian kernel	6.72	(0.39)	6.31	(0.11)	6.13	(0.08)	6.04	(0.05)	6.02	(0.06)

Table 2.1: Performance results in the queueing example. For the SALP and RSALP methods, the number in the parenthesis gives the standard deviation across sample sets.

2.5.3 Results

Policies were evaluated using a common set of arrival process sample paths. The performance metric we report for each control policy is the long run average number of jobs in the system under that policy,

$$\frac{1}{T}\sum_{t=1}^{T}\|x_t\|,$$

where we set $T \triangleq 10000$. We further average this random quantity over an ensemble of 300 sample paths. Here we have used the average cost associated with a particular policy as the evaluation criterion. We justify this in Section A.3 of the Appendix.

Further, in order to generate SALP and RSALP policies, state sampling is required. To understand the effect of the sample size on the resulting policy performance, the different sample sizes listed in Table 2.1 were used. Since the policies generated involve randomness to the sampled states, we further average performance over 10 sets of sampled states. The results are reported in Table 2.1 and have the following salient features:

1. *RSALP outperforms established policies:* Approaches such as the Max-Weight or 'parametric' ADP with basis spanning polynomials have been previously shown to work well for the problem of interest. We see that the RSALP with more than 300 samples achieves performance that is superior to these extant schemes.

2. *Sampling improves performance:* This is expected from the theory in Section 2.3. Ideally, as the sample size is increased one should relax the regularization. However, for our experiments

we noticed that the performance is quite insensitive to the parameter Γ. Nonetheless, it is clear that larger sample sets yield a significant performance improvement.

3. *RSALP in less sensitive to state sampling:* We notice from the standard deviation values in Table 2.1 that our approach gives policies whose performance varies significantly less across different sample sets of the same size.

In summary, we view these results as indicative of the possibility that the RSALP may serve as a practical and viable alternative to state-of-the-art parametric ADP techniques.

2.6 Conclusion

This paper set out to present a practical non-parametric algorithm for approximate dynamic programming building upon the success of linear programming based methods that require the user to specify an approximation architecture. We believe that the RSALP, presented and studied here, is such an algorithm. In particular, the theoretical guarantees presented here establish the 'non-parametric' nature of the algorithm by showing that increased sampling effort leads to improved approximations. On the empirical front, we have shown that our essentially 'generic' procedure was able to match the performance of tailor made heuristics as well as ADP approaches using pre-specified approximation architectures. Nevertheless, several interesting directions for future work are evident at this juncture:

- The choice of kernel: The choice of kernel matters in so much as the feature map it encodes allows for approximations with small Hilbert norm (i.e., small C). Thus, a good choice of kernel would require fewer samples to achieve a fixed approximation accuracy than a poor choice of kernel. That said, picking a useful kernel is an apparently easier task than picking a low-dimensional architecture — there are many full-dimensional kernels possible, and with sufficient sampling, they will achieve arbitrarily good value function approximations. Nevertheless, it would be valuable to understand the interplay between the choice of kernel and the corresponding value of C for specific problem classes (asking for anything more general appears rather difficult).

- The active set method: In future work, we would like to fine tune/ build more extensible

software implementing our active set method for wider use. Given the generic nature of the approach here, we anticipate that this can be a technology for high-dimensional MDPs that can be used 'out of the box'.

Chapter 3

Dynamic Matching Problems

3.1 Introduction

In recent years a number of two-sided marketplaces have sprung up. A company like Uber must match drivers with passengers who are in need for a ride. In another example, a company like Airbnb has hosts, people who are willing to offer their house for a fee, and guests, people who are looking for a temporary accommodations. In both cases there are agents on different sides of the market place, and a central agency who facilitates matching. On one hand Airbnb is a platform where guests and hosts interact over a period of time to organically get matched with one another. On the other hand on Uber, the central authority (Ubers allocation strategy) matches drivers with passengers. In the later case the agents can only decide whether or not they want accept the allocation the central authority makes. We will focus on this case.

In another context, organ exchanges like UNOS[1] have been used allocate organs, typically from a cadaveric patient, to those in need. The matching of organs to patients is complicated by the fact that the life span of the graft organ depends on the tissue and blood types of the organ and the patient. In such marketplaces the central authority performs the matching of organs with patients.

In yet another example, in online advertising, the advertiser typically has an inventory of ads to be shown to visitors of webpages it has access to. Thus the online advertiser must perform the allocation of ads to impressions. A typical advertiser may have a number of campaigns. Each

[1]United Network for Organ Sharing http://www.unos.org/.

of these campaigns will add to the inventory of ads the advertiser has. Ads are usually targeted and the advertiser usually has a lot of information about the impression. Advertisers problem in this case is to carefully allocate ads to impressions over a period of time, while staying within the inventory limit.

Examples we discussed so far are that two-sided marketplaces. But in general the allocations to be performed may have a more complicated structure. For example, in recent years UNOS has started performing cyclic transfers of kidneys. A patient in need of a kidney may have a close kin or a friend willing to donate their kidney to him. However, the patient and the donor might be incompatible with each other. We could also have another pair of donor and patient in the same situation. Additionally, the donor of the first pair may be compatible with the patient from the second pair and vice versa. In such a case, a transfer of kidneys form the donors to patients of the other pair will result in both patients receiving kidneys. In general, there can be a cyclic exchange with kidneys with more than two such pairs.

All of these marketplaces have the following in common:

Dynamic decisions. In all these problems decisions have to be made in a dynamic way. The agents can enter and leave the market. A matched agent leaves the market, so while matching one must consider the future value of the agent.

Compatibility. When matching decisions are made the central authority must consider the compatibility of the agents being matched. In case of a kidney exchange. compatibility comes from blood and tissue types. In the case of Uber, compatibility comes from geographical distance. In the case of online advertising, the compatibility is how likely a user is to click on an add. This can be learnt from past observations of user behavior.

Static problems are easy. In each of the cases if these problems were static in nature, they would be straightforward. In two-sided marketplace, assuming that the compatibility of agents on two sides is known, the problem can be modeled as a bipartite matching problem. It is well known that bipartite matching can be solved efficiently as the size of the problem grows. On the other hand the dynamic nature makes the optimization problems very challenging.

In this chapter we come up with a framework that can be used to model the allocation problems in such marketplaces. The kind of problems we mentioned can be seen as special cases of this general model. Our model is a finite horizon MDP, where the state is the current populations of agents

in the system and the actions are the matchings. Not surprisingly, due to the enormity of the state space, the MDP is intractable. To come up with a tractable approach we use the linear programming approach to Approximate Dynamic Programming. Our approach has the following features:

- Our algorithm is easy to implement. The numerical effort required to evaluate a policy given by our approach is similar to that of solving a corresponding static problem. We will make these statements precise in the subsequent sections.
- The approach comes with theoretical guarantees. As all parametric ADP based algorithms the performance of the approach depends on the basis functions chosen. Firstly, we provide a guarantee that shows that if the approximate architecture is able to approximate the value function well it results in a low approximation with our approach. Secondly, we show that if we use the *separable* basis functions our approach is approximately optimal in the asymptotic regime for Stochastic Assignment problem, a special case of our model.
- Numerical simulations show a promising gain over heuristic approaches. We will see that myopic matching, where agents in the current population are matched to maximize the value gained, will be only obvious approach to these dynamic problems. In the high-dimensional regime that we focus on, common approaches in the literature cannot be applied. We will show with numerical simulations that our ADP based approach yields a large gain over the myopic policy.

In Section 3.2 we discuss out model. Section 3.3 gives a number of real-life examples of our model. We will focus on these examples in the rest of the paper. Section 3.4 describes our algorithmic contribution. In Section 3.5 we discuss a couple of theoretical guarantees on our algorithm. Finally, in Section 3.6 we discuss numerical experiments.

3.2 Setup

Our decision problem involves dynamically *matching* certain items from a population. Here matching refers to a generalized allocation which we will discuss shortly. When items are matched, we obtain a certain reward which depends on the *type* of the items matched. We assume that the types

are from a set $\mathcal{T}$. Our running examples in the paper involve kidney allocation problems. In these cases, the type refer to the collection of attributes of a kidney such as age, sex, blood type and tissue types. What makes the decision problem challenging is that these items arrive and depart and decisions are made each time period on which items are to be matched. We now describe the dynamic decision making problem in more detail.

We have a decision problem with the time horizon T. Time periods are indexed as 0 to $T-1$. At time $t=0$ we start with a random initial population $\mathcal{S}_0$. Each element of the set $\mathcal{S}_0$ has a type in the set $\mathcal{T}$.[2] The set $\mathcal{S}_0$ is drawn from a distribution over multi-sets of $\mathcal{T}$. We denote this set of multi-sets $\mathbb{N}^{\mathcal{T}}$ by $\mathcal{X}$. This population changes over time. The population of items at time t is denoted by $\mathcal{S}_t \in \mathcal{X}$. The population changes according to the following dynamics:

- **Decision at each time period:** The set of decision at each time period depends on the population at that time period, i.e. $\mathcal{S}_t$. Consider a network where with the nodes are the items in the population. In addition, we have two special nodes, the source and the sink. We denote them by s and d respectively. We call the collection of these nodes at time t as $\overline{\mathcal{S}_t} \triangleq \mathcal{S}_t \cup \{s,d\}$. Similarly, let $\overline{\mathcal{T}} = \mathcal{T} \cup \{s,d\}$. We use $\sigma \in \overline{\mathcal{S}_t}$ to denote an arbitrary item. With an abuse of notation we use σ to also denote the type of the item σ.

  In particular, the set of actions available at any time period can be described by the following set -

$$\pi \in \{0,1\}^{\overline{\mathcal{S}_t}\times\overline{\mathcal{S}_t}} \tag{3.1a}$$

$$\pi_{\sigma\sigma} = \pi_{\sigma s} = \pi_{d\sigma} = 0,\ \forall \sigma \in \mathcal{S}_t, \tag{3.1b}$$

$$\sum_{\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma'\sigma} = \sum_{\sigma' \in \bar{\mathcal{S}}_t} \pi_{\sigma\sigma'}, \forall \sigma \in \mathcal{S}_t \tag{3.1c}$$

$$\sum_{\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma'\sigma} \leq 1,\ \forall \sigma \in \mathcal{S}_t. \tag{3.1d}$$

  The constraints (3.1a) ensure the integrality of the action. Any given edge, thus, can have a unit flow or no flow. We disallow self loops. Further s only has outgoing flows and d only has incoming flows. These restrictions are reflected in (3.1b). The constraints (3.1c) just ensure that the flow-in equals the flow-out for all nodes except source and the sink. In addition, we

[2] This allows for multiple items of the same type.

have the constraint that the flow-in (or the flow-out) for all the regular nodes is less than 1, a capacity constraint on the nodes. This set depends on $\mathcal{S}_t$ and we call it $\mathcal{P}(\mathcal{S}_t)$.

For each of the action described above we collect a certain reward. Consider a function $w : \overline{\mathcal{T}} \times \overline{\mathcal{T}} \to \mathbb{R} \cup \{-\infty\}$. The immediate reward obtained for any action $\pi \in \mathcal{P}(\mathcal{S}_t)$ is,

$$w(\pi) \triangleq \sum_{\sigma,\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma\sigma'} w(\sigma, \sigma). \tag{3.2}$$

Thus if $\pi_{\sigma\sigma'} = 1$ then we obtain the reward associated with matching σ to σ', $w(\sigma, \sigma')$. We call this function the *compatibility function*. For example, in kidney allocation the score associated with matching a donor kidney to a recipient can be used as the compatibility function. With an abuse of notation, we use w to denote both the compatibility function and the reward function associated with taking an action π as in (3.2).

To disallow certain flows in the decision set, for some $\sigma, \sigma' \in \mathcal{S}$, we set $w(\sigma, \sigma') = -\infty$. This will induce a certain graph structure on the set of nodes. Thus, when $w(\sigma, \sigma') = -\infty$, we can interpret this as not having an edge from σ to σ'. When $\sigma = d$ or $\sigma' = s$, $w(\sigma, \sigma')$ is always $-\infty$. In addition $w(s, d) = -\infty$ to prevent flows directly from source to sink.

- **Arrival Process:** The arrival process can be thought of as happening in two steps. First the number of arrivals is sampled with the random number N_t. N_t is i.i.d. across time periods. Each of the arrival item has a type that has a distribution from F. Again the draw of each of N_t items is independent of the other and of N_t.

- **Departure Process:** Departure happen in two ways. At time period t if the item $\sigma \in \mathcal{S}_t$ is matched, i.e.

$$\sum_{\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma'\sigma} = \sum_{\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma\sigma'} = 1,$$

then it leaves the system. All the items that are not matched can depart with probability that depends on the type of the item q_σ. The departure of any such item is independent of departures of any other items or any arrivals.

We are interested in maximizing the total reward,

$$\begin{aligned} \text{(P-DP)} \triangleq \text{maximize} \quad & \mathsf{E}\left[\sum_{t=0}^{T} w(\pi_t)\right] \\ & \pi_t \in \mathcal{P}(\mathcal{S}_t) \text{ is } \mathcal{F}_t\text{-measurable}, \quad \forall\ 0 \le t < T, \end{aligned} \tag{3.3}$$

$\mathcal{F}_t \triangleq \sigma(\mathcal{S}_0, \ldots, \mathcal{S}_t)$. The constraint that π_t is $\mathcal{F}_t$ measurable insures that the policies we come up with are 'non-anticipating', i.e. actions at a certain point of time do not depend upon the future randomness. In Section 3.4 we pose this problem as a Markov Decision Process (MDP).

Consider a simple approach to this problem where instead of worrying about the future rewards, at a certain point of time t, we look to maximize the immediately available reward. We call this the *myopic* policy. For this policy the optimization problem we must solve at each step is,

$$\begin{aligned} \text{maximize} \quad & w(\pi) = \sum_{\sigma,\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma\sigma'} w(\sigma, \sigma) \\ & \pi \in \mathcal{P}(\mathcal{S}_t). \end{aligned} \tag{3.4}$$

Let $\text{conv}(\mathcal{P}(\mathcal{S}_t))$ be the convex hull of $\mathcal{P}(\mathcal{S}_t)$ and consider the relaxed problem,

$$\begin{aligned} \text{maximize} \quad & w(\pi) = \sum_{\sigma,\sigma' \in \overline{\mathcal{S}_t}} \pi_{\sigma\sigma'} w(\sigma, \sigma) \\ & \pi \in \text{conv}(\mathcal{P}(\mathcal{S}_t)). \end{aligned} \tag{3.5}$$

Now note that the objective is linear in π. This ensures that the relaxation is without loss, i.e. the integral optimal solution of (3.5) lies in the optimal solution of (3.4).[3] It follows that the decision problem at each step for the myopic policy is a simple linear program.[4] In this paper, we come up with policies that have the same computational complexity as the myopic policy.

3.3 Examples

Having described the abstract decision problem, we illustrate many well known problems fall under this framework. Thus the algorithm we prescribe for this model can be used for these seemingly different problems.

[3]It is well known that network flow problems with integral capacity constraints have integral flows in the optimal solution set. This follows from the fact that the matrix associated with the linear network constraints are Totally Unimodular. For more details we refer you to Chapter 8 of [Schrijver, 2000].

[4]There are many specialized algorithms to solve network flow problems that outperform the linear programming approach. We refer the reader to [Lawler, 2001].

3.3.1 Stochastic Assignment

The stochastic assignment problem is a classical problem in the operations research literature first described in the seminal work of [Derman *et al.*, 1972]. This is a bipartite matching problem where one side of the matching is known apriori and items the other side arrives one at a time. Let the initial population on the known side be $\mathcal{A}_0$, with $|\mathcal{A}_0| = T$. At each time period t we observe an arrival of of an item $\beta_t \in \mathcal{B}$. Here $\mathcal{B}$ is the space of all possible types of the unknown side. It is assumed that the types of β_t are independent and identically distributed.

At each time period we must match the item β_t with one of the items from $\mathcal{A}_t$, where $\mathcal{A}_t$ denotes the population of the known side at time t. If item $\alpha \in \mathcal{A}$ is matched with the item $\beta \in \mathcal{B}$ then we obtain the reward $w(\alpha, \beta)$ where $w : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ is a known function. If $\alpha \in \mathcal{A}_t$ is matched at time period t then it disappears and $\mathcal{A}_{t+1} = \mathcal{A}_t / \{\alpha\}$. This process is repeated for T time periods.

To check that this is a special case of the general formulation given in the previous section, let $\mathcal{T}$ be the set $\mathcal{A} \cup \mathcal{B}$. Let the departure rate be such that $q_\sigma = 1$ if $\sigma \in \mathcal{B}$ and $q_\sigma = 0$ if $\sigma \in \mathcal{B}$. Thus the types $\mathcal{B}$ perish immediately if not matched and the types $\mathcal{A}$ never experience an exogenous departure. For the arrival process let us have $N_t = 1$ deterministically and as with the general case the arrival is sampled with the distribution F on $\mathcal{B}$. F has no mass on $\mathcal{A}$, so the known side has no additions.

With an extension of the function w from $\mathcal{A} \times \mathcal{B}$ to $\overline{\mathcal{T}} \times \overline{\mathcal{T}}$, we would obtain the compatibility function for the general model. This extension which we call $\overline{w}$ here is of the form,

$$\overline{w}(\sigma, \sigma') = \begin{cases} 0 & \text{if } \sigma = s, \sigma' \in \mathcal{A} \\ 0 & \text{if } \sigma \in \mathcal{B}, \sigma' = d \\ w(\sigma, \sigma') & \sigma \in \mathcal{A}, \sigma' \in \mathcal{B} \\ -\infty & \text{otherwise.} \end{cases}$$

In general, with high dimensional space of types $\mathcal{A}$ and $\mathcal{B}$ this problem (a high dimensional DP) is numerically intractable. [Derman *et al.*, 1972] and [Derman *et al.*, 1975] consider a number of special cases of this problem with assumptions about w such as linearily and concavity along with the fact that $\mathcal{B}$ and $\mathcal{A}$ are small and finite to come up with exact solutions to this dynamic program. However, in general an exact solution would be intractable.

A natural kidney allocation problem that can be modeled using this framework is that of

matching patients in need of kidneys with cadaveric donors. Since such kidneys must transplanted within a few days, the allocation system must decide which patient is to be offered the kidney in a short span of time. The allocation system typically has a scoring system where the compatibility of donors with recipients using attributes such tissue and blood types, age and gender. This score can be used as the compatibility function of the model. The stochastic assignment model has been used for this kidney allocation problem previously in [Zenios *et al.*, 2000; Su and Zenios, 2005].

The typical approach used for such a problem is to assume that there are a few number of types and large number of time periods. Under this assumption, the empirical distribution of the arriving kidneys is close to the real distribution with high probability. Policies can then be devised that act assuming such average-case arrival in the future. However, typically kidneys have a high dimensional space of attributions and there is an explosion in the space of types. Policies that rely on this *mean field* assumptions will typically not work well. Indeed in most cases obtaining policies for such high dimensional problems might be intractable. Our approach on the other hand does not rely on this assumption and is tractable even with a very large number of types.

This problem has also generated considerable interest in the computer science community with the motivating application being online advertising. The focus in this literature tends to be on proving constant factor approximations algorithms against an adversary. The seminal work of [Karp *et al.*, 1990] introduced a $1 - e^{-1}$ constant factor algorithm for this model. In their work, the bipartite matching was done on an unweighted graph, which in our framework corresponds to compatibility function taking values on $\{1, -\infty\}$. Given the sequence of matchings up to a time period t, the adversary can pick an arrival sequence to minimize the total reward. The algorithm described by [Karp *et al.*, 1990] are equivalent to the greedy algorithm in the case of stochastic arrivals.

This result has been extended to the more general 'ad-words problem' in [Mehta *et al.*, 2005; Buchbinder *et al.*, 2007]. Recently, [Goel and Mehta, 2008] showed that the algorithm also yields a $1 - 1/e$ guarantee for the case of i.i.d. arrivals. These approximation results in the case weighted matching rely heavily on the 'fractional inventory' assumption which, roughly stated, requires that the space $\mathcal{A}$ is finite and the loss of any item is negligible considering the overall inventory. Thus, this assumption is similar to the mean field assumption described earlier.

3.3.2 General Bipartite Matching

Although stochastic assignment is a model widely studied in the literature, its applicability to real life scenarios in restrictive. This limitation comes from the restrictive nature of the arrival and departure processes. In the kidney allocation example, the assumption that the patients waiting for kidneys will stay forever if not matched is inconsistent with the observations. Further any realistic model will allow for arrival of new patients. This is not modeled in the stochastic assignment case. Finally, the kidney exchanges also have non-cadaveric donors that might persist in the system for longer period of time than just one. Hence a better model will allow for more general arrival and departure processes.

Again, as with the previous model, we have sets $\mathcal{A}$ and $\mathcal{B}$ and the space of types is $\mathcal{T} = \mathcal{A} \cup \mathcal{B}$. Let $\mathcal{A}_t$ and $\mathcal{B}_t$ be the set of items an each side that exist at period t. We use the same extended function $\bar{w}$ as in the previous section. This makes the decision problem at each time period a problem on a bipartite matching polytope with the two nodes on the two sides being $\mathcal{A}_t$ and $\mathcal{B}_t$. In this case we can match one or more pairs of the type (α, β) with $\alpha \in \mathcal{A}_t$ and $\beta \in \mathcal{B}_t$.

The arrival and departure processes are now more general. Hence the items in $\mathcal{A}_t$ might depart with the exogenous process, modeling the death of patients in the kidney allocation case. There might also be additions to $\mathcal{A}_t$, hence F has measure on $\mathcal{A}$ as well. Further some $\mathcal{B}$ might persist in the system even if not matched.

3.3.3 Cycles and Chains in Kidney Matching Pools

Lastly, we give the example of a problem where the matching problem at each time period does not have the bipartite structure as before and we exploit the full generality of the network flow constraints. This setting is the so-called kidney paired exchange [Roth *et al.*, 2004]. In the model involves pairs of donor and recipients which are incompatible with each other. Such a pair typically consists of a patient in need of a kidney and her friend or relative who is willing to donate to her but is incompatible. If one has a large number of such pairs, two or more of such pairs can be matched to ensure that each patient in this matching receives a compatible kidney. This can be achieved by forming a cycle on such pairs such that patient of a particular pair receives kidney from a donor from a pair before it in this cycle.

Traditionally two of such pairs were matched as a two way exchange. In this case, the donor

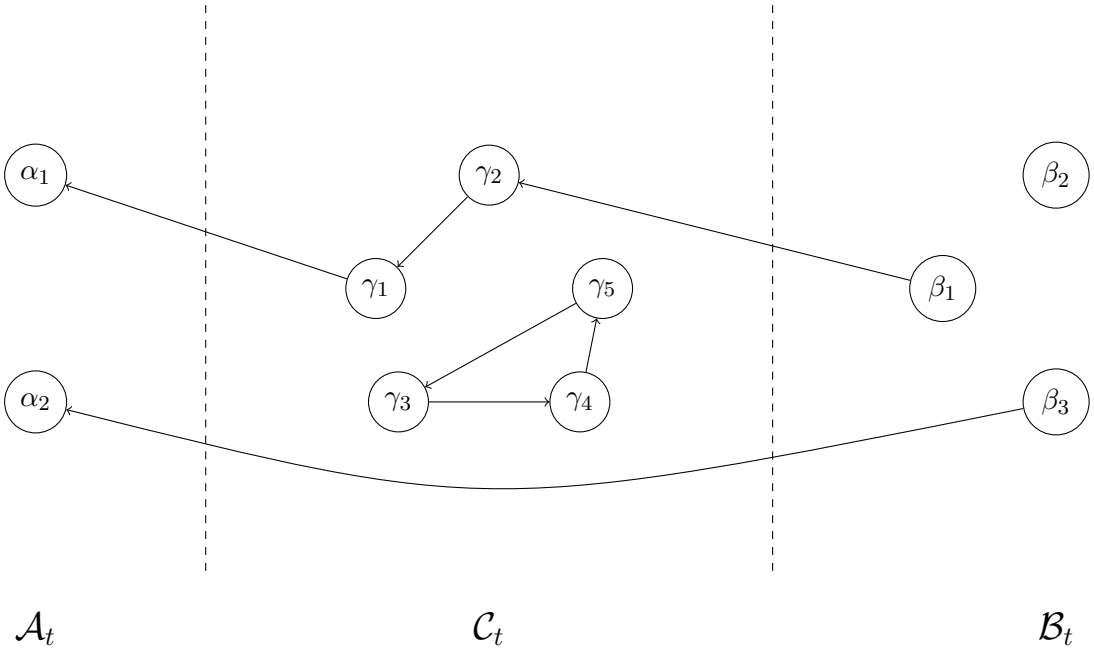

Figure 3.1: A possible state at time t and a possible set of matches.

of the first pair will donate to the patient in the second pair and the donor of the second pair will donate to the patient in the first. United Network for Organ Sharing (UNOS) has recently started a pilot program where such paired donations with more than two pairs are performed. In general, a cycle with a large number of pairs poses many logistic challenges and are not preferred.

We may combine the exchange for paired kidney donation with the more traditional two way exchange mentioned in the earlier sections. In this model, we allow for altruistic donors, patients in need of kidneys and donor-recipient pairs.We allow for,

1. A traditional transplant with a patient receiving a kidney from a donor,

2. A cycle of donor-recipient pairs,

3. A chain of transplants that starts from an altruistic donor and ends with either a pair or a patient.

Within our framework, the pairs, the altruistic donors and patients will each form an item. The type of such a pair will be denoted by the the pair consisting of attributes of the donor and the recipient in the pair. Our space of types is now $\mathcal{T} = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$, where $\mathcal{A}$ and $\mathcal{B}$ as mentioned in the

previous model. In addition, we have the pairs which are from the set $\mathcal{C}$. At each time period we have the current population as the triplet $\mathcal{A}_t$, $\mathcal{B}_t$ and $\mathcal{C}_t$. Each item in $\mathcal{A}_t$ may depart exogenously if not matched with probability $q_{\mathcal{A}}$, and similarly we have $q_{\mathcal{B}}$ and $q_{\mathcal{C}}$ for $\mathcal{B}$ and $\mathcal{C}$ respectively. We allow for the most general arrival process within our framework.

To explain the decision problem let us consider Figure 3.1, where the nodes denote the current population for a given time-period. In this case, we have two patients, α_1 and α_2, three altruistic donors, β_1 to β_3 and five pairs, γ_1 to γ_5. A possible allocation is shown in the figure. We use the kidney donated by β_3 for treating α_2, this is similar to matchings performed in the models described previously. A chain of kidney transfers starts from the donor β_1 and ends at α_1. Thus patients in the pairs γ_1 and γ_2 receive kidneys and so does α_1. Lastly, we form a cycle with γ_3, γ_4 and γ_5, which involves a three way exchange of kidneys, from γ_3 to γ_5, from γ_5 to γ_4 and from γ_4 to γ_3.

In general the space of actions we may have a combination of cycles and paths on the items in the current populations with the following conditions -

- Any path must start with a node in $\mathcal{B}_t$,
- Any node in a cycle and any inner node of a path must be in $\mathcal{C}_t$,
- Any path may end in $\mathcal{A}_t$ or $\mathcal{C}_t$,
- No path or cycle may intersect.

The feasible set of actions described by these conditions is given by the network flow constrains in (3.1). In addition we must have a reward function such that,

- $w(s, \sigma) = -\infty$ for any $\sigma \in \mathcal{A}_t \cup \mathcal{C}_t$.
- $w(\sigma, d) = -\infty$ for any $\sigma \in \mathcal{B}_t$
- $w(\sigma, \sigma') = -\infty$ if $\sigma, \sigma' \in \mathcal{A}_t$ or $\sigma, \sigma' \in \mathcal{B}_t$.

In practice, due to the logistic complications associated with more than three way exchanges, we have a cap of the size of the cycles we may form. Such a restriction on π results in a combinatorial set of constraints. Even the static problem associated with such a constraint is numerically challenging and finding efficient algorithm for it is an active area of research. In our model we place no such restriction on the set of actions.

3.4 Policies

In this section we describe how the matching decision problem of Section 3.2 is a Markov Decision Process (MDP). Unfortunately the MDP has very high dimensional state space and, indeed, its optimal solution is intractable. Approximate dynamic programming (ADP) is a collection of algorithms used to deal with such problems. The idea is to find a functional approximation of the optimal *value function*, i.e., the optimal reward obtained if we take the right decisions at each time period. In particular, we consider an optimization based approach to ADP which builds on the previous work of [Desai *et al.*, 2011; Schweitzer and Seidman, 1985; de Farias and Van Roy, 2003].

We first note the challenges associated with the problem at hand:

- The value function is approximated using a linear combination of the so-called basis functions. Given such set of basis functions, also called as the approximation architecture, any ADP algorithm must come up with appropriate weights. A challenge common to any application of ADP is choosing an expressive approximation architecture, i.e. an approximation architecture which for a certain choice of weights yields a reasonable approximation of the value function. We give a very natural expressive approximation architecture in Section 3.4.2.

- We must chose an approximation architecture that yields policies that are easy to evaluate. Note that the possible actions at each time period for the general problem will combinatorial and enumerating them will not be an option. However, we have that the myopic policy for the model is easy to evaluate. This is due to the fact that actions form the vertices of a network flow polytope. We show in 3.4.2 that certain choice of the approximation architecture leads to policies that are as easy to evaluate as the myopic policy.

- The enormous size of the action space makes the use of Approximate Linear Program (ALP), the most widely known linear programming based ADP algorithm, unsuitable since we must have one constraint per state action pair. To this end, we use the Smoothed Approximate Linear Programming (SALP) due to [Desai *et al.*, 2011]. We show that this formulation can be alternatively viewed as an unconstrained convex stochastic optimization problem. Further the stochastic sub-gradient for such a problem is easily computed. This naturally leads us to the use of stochastic sub-gradient method.

- The guarantee for the SALP algorithm applies for the infinite horizon discounted case. To be applicable to our problem, we extend this guarantee to include the finite horizon undiscounted case.

3.4.1 Markov Decision Process

We set up the problem of Section 3.2 as a Markov Decision Process.

- **State:** We have a finite horizon problem with $t = 0, \ldots, T-1$ as the decision-making epochs. The state at time t, $\mathcal{S}_t \in \mathcal{X}$.
- **Action:** The set of actions admissible in state $\mathcal{S}_t$ is $\mathcal{P}(\mathcal{S}_t)$. This is given by the constraints in (3.1). We note that in general the space of actions is exponential in $|\mathcal{S}_t|$. Let $\mathcal{A} \triangleq \cup_{\mathcal{S}\in\mathcal{X}}\mathcal{P}(\mathcal{S})$, the set of all possible actions.
- **Reward Function:** The immediate reward for taking a particular action $\pi \in \mathcal{P}(\mathcal{S}_t)$ is,

$$w(\pi) = \sum_{\sigma,\sigma' \in \mathcal{S}_t} \pi_{\sigma\sigma'} w(\sigma, \sigma'),$$

  the aggregate compatibility function.

The optimization problem is succinctly written down as (P-DP) in (3.3). A policy used for an MDP specifies which action is to be taken at a particular time-period t given the history of evolution of the process up to time t.

Proposition 3. *The optimal solution to (P-DP) exists. There exists an optimal solution π^* and a sequence of maps $\mu_t^* : \mathcal{X} \to \mathcal{A}$, for $0 \le t < T$, such that $\pi_t^* = \mu_t^*(\mathcal{S}_t)$.*

Proof. ■

In other words there exists an optimal Markovian policy for the MDP.

Let μ denote an arbitrary time-dependent Markovian policy. Given a policy μ the sequence $\{(\mathcal{S}_t, \pi_t)\}_{0\le t<T}$ evolves according to a Markov chain. Let us define J_μ as the expected reward obtained by a policy. Thus,

$$J_\mu^t(s) = \mathsf{E}_\mu \left[\sum_{\tau=t}^{T-1} w(\pi_\tau) \right].$$

We define $J^* = J_{\mu^*}$. The function J^* is of central importance to the ADP since given J^*, we can obtain μ^* as any policy that satisfies,

$$\mu^{*,t}(s) \in \underset{\pi \in \mathcal{P}(s)}{\operatorname{argmax}} \left(w(\pi) + E_\pi J^{*,t+1}(s) \right), \quad \forall 0 \leq t < T, \ s \in \mathcal{X}.$$

Let $\mathcal{J}$ be the set of all possible maps of the form $\{0, \ldots, T\} \times \mathcal{X} \to \mathbb{R}$. This is the space of value functions. We denote $J(t, \cdot)$ as $J^t(\cdot)$. Also define,

$$E_\pi J^{t+1}(s) \triangleq \mathsf{E}[J^{t+1}(\mathcal{S}'_{t+1}) | \mathcal{S}_t = s, \pi_t = \pi], \tag{3.6}$$

the expected value of $J^{t+1}(S_{t+1})$, given all information up to time t. Similarly, for a policy μ we define,

$$E_\mu J^{t+1}(s) \triangleq \mathsf{E}[J^{t+1}(\mathcal{S}'_{t+1}) | \mathcal{S}_t = s, \pi_t = \mu(s)]. \tag{3.7}$$

The idea is to find an approximation to J^*, say $\tilde{J}$, and find a policy that is *greedy* with respect to $\tilde{J}$, i.e. a μ that satisfies,

$$\mu^t(s) \in \underset{\pi \in \mathcal{P}(s)}{\operatorname{argmax}} \left(w(\pi) + E_\pi \tilde{J}^{t+1}(s) \right), \quad \forall 0 \leq t < T, \ s \in \mathcal{X}.$$

The most straightforward way to compute J^* is by backward recursion. It is a well-known fact that,

$$J^{*,t}(s) = \max_{\pi \in \mathcal{P}(s)} \left[w(\pi) + E_\pi J^{*,t+1}(s) \right].$$

Setting $EJ^{*,T}$ to be 0, uniformly, we recursively compute $J^{*,t}$ from $T-1$ to 0, yielding the value function.Note that this algorithm is intractable due to the enormity of the state space. In the next sections we will look at ways to approximate J^*.

3.4.2 Approximation Architecture

Consider elements of $\mathcal{T}$ to be mapped in some Euclidean space $\mathbb{R}^m$ via a certain feature map $\Phi : \mathcal{T} \to \mathbb{R}^m$. With an abuse of notation we suppress this mapping and represent the elements in $\mathbb{R}^m$ by the points in $\mathcal{T}$ themselves. In particular, let

$$\langle \kappa, \sigma \rangle \triangleq \langle \kappa, \Phi(\sigma) \rangle = \sum_{i=1}^{m} \Phi_i(\sigma) \kappa_i. \tag{3.8}$$

Assumption 1. *We have that $\Phi_m(\sigma) = 1$, for all $\sigma \in \mathcal{S}$, i.e. one of the basis functions is a constant function.*

Let $\kappa_t \in \mathbb{R}^m$ for $t \in \{0, \ldots, T-1\}$. The approximation architecture we use is of the form -

$$\tilde{J}_t(\mathcal{S}_t; \kappa) = \sum_{\sigma \in \mathcal{S}_t} \langle \kappa_t, \sigma \rangle.$$

We approximate the value function at two levels:

1. Any item has a contribution that is independent of existence of any other item in the population at a particular point of time. In particular the existence of item σ at the time t has a contribution $\langle \kappa_t, \sigma \rangle$ to the value function approximation. We call such an approximation architecture *separable*. Such an approximation architecture ignores the higher order effects such as pairwise interactions of items.

2. Since we have a large number of types in general it would not be possible to fit arbitrary function over the space of types. Thus we restrict our search to the space of all functions in the range of Φ, i.e. all functions of the form (3.8). This ensures that eventually we optimize over a small number of variables as opposed to having the number of variables that scale with the size of the space $\mathcal{T}$.

Given this approximation architecture the decision problem at each time step is to solve the following optimization problem:

$$\underset{\pi \in \mathcal{P}(s)}{\text{argmax}} \qquad w(\pi) + E_\pi J^{t+1}(s).$$

The objective can be rewritten as,

$$w(\pi) + E_\pi J^{t+1}(s) \quad = \quad w(\pi) + \textstyle\sum_{\sigma \in s}(1 - q_\sigma)(1 - \sum_{\sigma'} \pi_{\sigma\sigma'})\langle \kappa_{t+1}, \sigma \rangle + \mathsf{E}[N_t \langle \kappa_{t+1}, \sigma \rangle].$$

Getting rid of the terms that do not depend on π we are left with the optimization problem,

$$\underset{\pi \in \mathcal{P}(s)}{\text{argmax}} \qquad w(\pi) - \sum_{\sigma \in s}(1 - q_\sigma) \sum_{\sigma'} \pi_{\sigma\sigma'} \langle \kappa_{t+1}, \sigma \rangle.$$

Such a policy has the following features:

- Since $w(\pi)$ is a linear function in π, the policy prescribed asks us to maximize a linear function over a network flow polytope. This is tractable problem which can be solved in general by linear programming. In the special cases when the polytope has the bipartite matching

structure for example Section 3.3.1 and Section 3.3.2, we may use more powerful algorithms e.g. the Hungarian algorithm [Lawler, 2001]. Indeed applying this policy is requires the same numerical effort as applying the greedy policy (μ_{greedy}).

- The policy has a nice interpretation. We can think of $\langle \kappa_{t+1}, \sigma \rangle$ as the estimate of the value of σ if not matched and does not experience any exogenous departure at time t. If the item is not matched we estimate the future value to be $(1-q_\sigma)\langle \kappa_{t+1}, \sigma \rangle$, where the factor $(1-q_\sigma)$ is due to the fact that it may disappear with probability q_σ. Indeed, if $q_\sigma = 1$, no future value is given to that item.

The limitation of this approximation architecture is that it does not consider the effect of pairwise (and higher order) interactions of the items. Two rare items might not be very valuable by themselves but together might be matched with a more frequent item to reap a high reward. Hence their pairwise potential would be higher than the sum of the individual ones. Having said that, such an approximation architecture will not yield tractable policies and thus will be difficult to implement in practice.

3.4.3 Smoothed Approximate Linear Program

In this section we describe our SALP-based procedure. The SALP approach as it appears in [Desai *et al.*, 2011] is an infinite horizon problem discounted problem. We will formulate a similar approach for the finite horizon undiscounted case. To find the approximation to J^* we solve the optimization problem,

$$\inf_{J \in \tilde{\mathcal{J}}} \mathsf{E}\left[(1+\gamma) \sum_{t=0}^{T} \max_{\pi \in \mathcal{P}(\mathcal{S}_t)} \left(w(\pi) + E_\pi J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t) \right)^+ + J_0(\mathcal{S}_0) \right]. \tag{3.9}$$

This program can be interpreted as follows -

- In this problem we are optimizing over some restricted space of value functions $\tilde{\mathcal{J}}$. A concrete example is the separable value function introduced in the previous section.

- The expectation is taken with respect to a sample path $\{\mathcal{S}_t\}_{0 \le t \le T-1}$ sampled according to a certain policy. For the purpose of theory this sampling must be done with the optimal policy. This is a standard assumption which guarantees an algorithm which does not require exponential computational effort despite the enormity of the state space [de Farias and Van Roy, 2003;

Desai *et al.*, 2011]. In the absence of this assumption we would need to sample all state-action pairs, an undesirable situation, especially since the state space under consideration is infinite dimensional. In practice the sampling would be done with a surrogate policy. The use of the surrogate is consistent with previous papers on the linear programming approach to ADP [de Farias and Van Roy, 2003; Desai *et al.*, 2011].

- Consider the Problem (3.9) when γ is set to ∞. This special case yields the optimization problem -

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}[J^0(\mathcal{S}_0)] \\ \text{subject to} \quad & J^t(\mathcal{S}_t) \geq w(\pi) + E_\pi J^{t+1}(\mathcal{S}_t) \\ & \qquad \forall \mathcal{S}_t \in \mathcal{X}, 0 \leq t \leq T-1, \pi \in \mathcal{P}(\mathcal{S}_t) \\ & J \in \tilde{\mathcal{J}}. \end{aligned} \tag{3.10}$$

  This is the finite version of the Approximate Linear Program. The infinite dimensional version of this formulation was first introduced in [Schweitzer and Seidman, 1985] and later studied in [de Farias and Van Roy, 2003]. We can argue by induction that it gives an upper bound on the true value function at each step. Further if $J^* \in \tilde{\mathcal{J}}$ then the program recovers J^*.

- The constraints applied in (3.10) can be restrictive from the point of view of approximation the value function. In the formulation (3.9), instead of imposing these constraints, we penalize the violation of them in the objective. Further the weight given to violation depends on how likely the state is likely to be observed under the sampling distribution. γ is used as a trade-off parameter. Thus our optimization problem for general γ is -

$$\begin{aligned} \text{minimize} \quad & \mathsf{E}[J^0(\mathcal{S}_0) + (1+\gamma)\textstyle\sum_{t=1}^{T-1} s(\mathcal{S}_t)] \\ \text{subject to} \quad & s(\mathcal{S}_t) + J^t(\mathcal{S}_t) \geq w(\pi) + E_\pi J^{t+1}(\mathcal{S}_t) \\ & \qquad \forall \mathcal{S}_t \in \mathcal{X}, 0 \leq t \leq T-1, \pi \in \mathcal{P}(\mathcal{S}_t) \\ & J \in \tilde{\mathcal{J}}. \end{aligned} \tag{3.11}$$

- The problems (3.9) and (3.11) are equivalent. The (3.9) is stated as convex stochastic optimization problem. If we use the approximation architecture described in the previous section then this becomes an unconstrained optimization problem in with parameters $\kappa \triangleq \{\kappa_t\}_{0 \leq t \leq T-1}$. We would be able to apply stochastic sub-gradient algorithm as long as

sub-gradient for each sample path is easily computable. We will show that this is indeed the case.

Let v^* be the optimal value of the problem. The following result states that the optimization problem in general gives an upper bound for all $\gamma \geq 0$ and further recovers the exact solution when the value function lies in the approximation architecture. This can be seen as the generalization of the more straightforward fact for $\gamma = \infty$. Although this is not a strong approximation guarantee we can think of this as a sanity check – in the idealistic sampling distribution the algorithm recovers the optimal cost to go function.

Proposition 4. *Assume that the states $\{\mathcal{S}_t\}_{0\leq t\leq T-1}$ are sampled with the optimal distribution. Let ν^{lp} be the optimal value of the (3.11) then $\nu^* \leq \nu^{lp}$. Further if $J^* \in \tilde{\mathcal{J}}$ then $\nu^* = \nu^{lp}$.*

Proof. We have for any $J \in \tilde{\mathcal{J}}$,

$$\begin{aligned}
v^* &= \mathsf{E}_{\mu^*}\left[\sum_{t=0}^{T} w(\mu^{*,t})\right] \\
&= \mathsf{E}_{\mu^*}\left[\sum_{t=0}^{T}\left[w(\mu^{*,t}) + E_{\mu^*}J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t) + \left(J^t(\mathcal{S}_t) - E_{\mu^*}J^{t+1}(\mathcal{S}_t)\right)\right]\right] \\
&\leq \mathsf{E}_{\mu^*}\left[\sum_{t=0}^{T}\left[\left(w(\mu^{*,t}) + E_{\mu^*}J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t)\right)^+ + \left(J^t(\mathcal{S}_t) - E_{\mu^*}J^{t+1}(\mathcal{S}_t)\right)\right]\right] \\
&\leq \mathsf{E}_{\mu^*}\left[\sum_{t=0}^{T}\left[\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi) + E_{\pi}J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t)\right)^+ + \left(J^t(\mathcal{S}_t) - E_{\mu^*}J^{t+1}(\mathcal{S}_t)\right)\right]\right] \\
&\leq \inf_{J\in\mathcal{J}} \mathsf{E}\left[\sum_{t=0}^{T}\left[(1+\gamma)\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi) + E_{\pi}J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t)\right)^+ + \left(J^t(\mathcal{S}_t) - E_{\mu^*}J^{t+1}(\mathcal{S}_t)\right)\right]\right] \\
&= \inf_{J\in\mathcal{J}} \mathsf{E}\left[\sum_{t=0}^{T}\left[(1+\gamma)\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi) + E_{\pi}J^{t+1}(\mathcal{S}_t) - J^t(\mathcal{S}_t)\right)^+ + \mathcal{J}_0(\mathcal{S}_0)\right]\right]
\end{aligned} \tag{3.12}$$

Here the last inequality follows from the telescoping nature of the sum in the previous equation.

Now assuming $J^* \in \tilde{\mathcal{J}}$, we can prove that this bound is actually tight and is achieved when $J = J^*$. To this end, notice that by the dynamic programming principle under μ^*, we have that:

$$\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi) + E_{\pi}J^{*,t+1}(\mathcal{S}_t)\right) = J^{*,t}(\mathcal{S}_t).$$

Thus,

$$\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi) + E_{\pi}J^{*,t+1}(\mathcal{S}_t) - J^{*,t}(\mathcal{S}_t)\right)^+ = 0,$$

and we have that the objective is -

$$\mathsf{E}\left[\sum_{t=0}^{T}\left[(1+\gamma)\max_{\pi\in\mathcal{P}(\mathcal{S}_t)}\left(w(\pi)+E_\pi J^{*,t+1}(\mathcal{S}_t)-J^{*,t}(\mathcal{S}_t)\right)^+ + J^{*,0}(\mathcal{S}_t)\right]\right] = \mathsf{E}\left[\sum_{t=0}^{T} w(\mu^{*,t})\right].$$

Thus we have that J^* lies in the space of optimal J. ■

As mentioned earlier we will numerically solve the problem described in (3.9) with Monte-Carlo sampling. As a stochastic approximation problem we can think of two general strategies for solving such a problem -

1. **Sample Average Approximation:** This method described in the paper [Desai *et al.*, 2011] samples states from the given distribution and approximates the distribution with an empirical distribution on these sampled states. When expressed as such the optimization problem becomes a linear program. Let us say that the sample paths are index by ω with $\omega \in \{1,\ldots,N\}$. Further let $\mathcal{S}_t(\omega)$ be the state sampled at time t in the sample path ω. We would have that the sample average approximation is -

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{N}\sum_{\omega=1}^{N}\left(J^0(\mathcal{S}_0)(w)+(1+\gamma)\sum_{t=1}^{T-1}s(\mathcal{S}_t)(\omega)\right) \\ \text{subject to} \quad & s(\mathcal{S}_t(\omega))+J^t(\mathcal{S}_t(\omega)) \geq w(\pi)+E_\pi J^{t+1}(\mathcal{S}_t(\omega)) \\ & \qquad \forall 1\leq\omega\leq N, 0\leq t\leq T-1, \pi\in\mathcal{P}(\mathcal{S}_t) \\ & J\in\tilde{\mathcal{J}}. \end{aligned} \tag{3.13}$$

   The main difficulty with this approach is that in general the size of the action space explodes with the state of the state, i.e. $|\mathcal{S}_t|$. To remedy this we may heuristically sample few of the actions. For example, we might just sample one action per state. If the sampling policy is optimal, it is not hard to see that the guarantee equivalent to Proposition 4 can be obtained for such an approach.

2. **Stochastic Sub-gradient Descent:** The other class of approaches to solving such problems fall under the umbrella term stochastic approximations. The most basic method in this category is the stochastic sub-gradient (or stochastic gradient when the objective is differentiable) method, which for simplicity is the only method we consider in this case.

Consider a sample ω the given distribution, i.e. say $\{\mathcal{S}_t(\omega)\}_{0\leq t\leq T-1}$ are sampled from the policy described. The term in the objective associated with this sample is,

$$(1+\gamma)\sum_{t=0}^{T}\max_{\pi\in\mathcal{P}(\mathcal{S}_t(\omega))}\left(w(\pi)+E_\pi J^{t+1}(\mathcal{S}_t(\omega))-J^t(\mathcal{S}_t(\omega))\right)^+ + J_0(\mathcal{S}_0(\omega)).$$

In our case $J^t(\mathcal{S}_t)$ is a linear function of κ. Thus the given expression is a piecewise linear convex function of r, let us call it $f(\kappa,\omega)$. Thus we must solve the optimization problem,

$$\min_{\kappa}\mathsf{E}[f(\kappa,\omega)].$$

We also note that the sub-gradient of $f(\kappa,\omega)$ with respect to r at a particular value can be easily obtained. To compute this for the nonlinear term,

$$\max_{\pi\in\mathcal{P}(\mathcal{S}_t(\omega))}\left(w(\pi)+E_\pi J^{t+1}(\mathcal{S}_t(\omega))-J^t(\mathcal{S}_t(\omega))\right)^+$$

we can use the envelope theorem and thus the sub-gradient $g(r,\omega)$ is,

$$g(\kappa,\omega)=(1+\gamma)\sum_{t=0}^{T}\left(\bigtriangledown_\kappa E_{\pi^*}J^{t+1}(\mathcal{S}_t(\omega))-\bigtriangledown_\kappa J^t(\mathcal{S}_t(\omega))\right)I(\omega,t,\kappa)+\bigtriangledown_\kappa J_0(\mathcal{S}_0(\omega)),$$

where π^* is to be interpreted as the action that achieves this maximum and,

$$I(\omega,t,\kappa)\triangleq\mathbb{I}_{w(\pi^*)+E_{\pi^*}J^{t+1}(\mathcal{S}_t(\omega))-J^t(\mathcal{S}_t(\omega)\geq 0}.$$

Having computed $g(\kappa,\omega)$ we can change the current estimate of κ_ω by $-\gamma_\omega g(r_\omega,\omega)$. The steps size can be chosen such that $\sum_\omega\gamma_\omega=\infty$ and $\sum_\omega\gamma_\omega^2<\infty$. A typical choice is

$$\gamma_\omega=A/(B+\omega),$$

with A and B some positive constants.

Thus the sub-gradient descent procedure is as follows -

- Start with a guess of κ_0, for example with all components set to 0.
- At each sample ω perform the update -

$$\kappa_{\omega+1}\leftarrow\kappa_\omega-\gamma_\omega g(\kappa_\omega,\omega)$$

- Keep track of the *best* κ_ω

It is not clear at this point what do we mean by the best κ_ω. In theory such a procedure is guaranteed to produce a sequence of κ_ω's with the property that,

$$\lim_{n\to\infty} \max_{\omega\in[1,n]} f(\kappa_\omega) = f(\kappa^*),$$

where κ^* is the optimal value. In practice, we do not have a way to evaluate f. We can approximate f by sampling. Alternatively we can pick an κ that yields the best policy. This is the approach we use in the numerics.

3.5 Theory

In this section we present theory that provides further insights into our procedure. For this purpose we present two results. The first one is a generic approximation guarantee for the finite-horizon version of the SALP algorithm, extending the results for the infinite horizon case discussed in the paper [Desai *et al.*, 2011]. This guarantee bounds the approximation error in the value function approximation in terms of the best achievable approximation error. In short, the result states that if the approximation architecture is able to approximate the value function well then our procedure is guaranteed to have a low approximation error. The second guarantee applies to the Stochastic Assignment case. It is an asymptotic result which states that on scaling the problem in terms of time periods, the approximation error in the value function scales sublinearly. To this end, we show that SALP approximation to the *value* of the problem is tighter than the *deterministic fluid approximation*.

3.5.1 Approximation Guarantee

In this section we provide a bound for the finite horizon version of SALP. This extends the results for the infinite horizon case discussed in [Desai *et al.*, 2011] in Theorem 2.

Recall that the space of types is the set $\mathcal{T}$, which we have assumed is a finite set. The natural state space for the MDP is a set of all multi-sets of $\mathcal{T}$, a space that is isomorphic to $\mathbb{N}^{|\mathcal{T}|}$. This is an infinite state space. To simplify our analysis we make a further assumption.

Assumption 2. *There exist finite numbers m_0 and m_a such that,*

$$|\mathcal{S}_0| \leq m_0, \qquad a.s.$$

and the number of arrivals for any time period,

$$N_t \leq m_a, \qquad a.s., \ \forall 0 \leq t < T.$$

Thus we have that the total number of items in the system at time t can be at most $m_0+(t+1)m_a$, we call this number m_t. Let $\mathcal{X}$ be the set of all multi-sets of $\mathcal{T}$. Also let,

$$\mathcal{X}_t \triangleq \{s \in \mathcal{X} || s| \leq m_t\}.$$

Our state space at time t is $\mathcal{X}_t$.

It follows that Assumption 2 insures that the state space is finite at each time period. This assumption is made to simplify the subsequent analysis.

Before describing the result, we set up some notation. Let $\tilde{\mathcal{J}} = \{\tilde{\mathcal{J}}_t\}_{0\leq t<T}$ be the space of value functions spanned by our approximation architecture. In particular, $\tilde{\mathcal{J}}_t$ is a set of functions from $\mathcal{X}_t$ to $\mathbb{R}$ (or simply $|\mathcal{X}_t|$ dimensional vectors) such that,

$$\tilde{\mathcal{J}}_t = \left\{ \mathcal{S}_t \mapsto \sum_{\sigma \in \mathcal{S}_t} \langle \kappa_t, \sigma \rangle + \theta_t \middle| \kappa_t \in \mathcal{H}, \theta_t \in \mathbb{R} \right\}.$$

Our result applies to the case when $\gamma = 2$, in (3.9), the optimization problem of interest we call SALP. Let $\{J_t^{SALP}\}_{0\leq t<T}$ be the optimal solution to (3.11) for this value of γ, where J^{SALP} is a function from $\mathcal{X}_t$ to $\mathbb{R}$. Similarly, let $\{J_t^*\}_{0\leq t<T}$ be the optimal value function.

Further, for each $0 \leq t < T$, let $\epsilon_t = \inf_{J\in\tilde{\mathcal{J}}_t} \|J - J_t^*\|_\infty$. Thus ϵ_t is the minimum achievable approximation error using the approximation architecture, measured using the ∞-norm. A low error would suggest a good approximation architecture.

Finally, let ν be the initial distribution on the states $\mathcal{X}$ induced by the MDP. And for any such distribution let the weighted 1-norm error be,

$$\|J\|_{1,\nu} = \sum_{\mathcal{S}\in\mathcal{X}_0} \nu(\mathcal{S})|J(\mathcal{S})|.$$

We are now ready to state our result. The proof of the result is added to Appendix B.1.

Theorem 3. *Given that $\gamma = 2$ and Assumption 2 holds,*

$$\|J_0^* - J_0^{SALP}\|_{1,\nu} \leq 3\epsilon_0 + 4\sum_{t=1}^{T-1} \epsilon_t.$$

The right side of the equation depends on the approximation architecture. In particular, it scales with the best achievable error measured using the infinity norm. The left side depends is the approximation error achieved by our procedure. It is measure with respect to the weighted 1-norm. Qualitative meaning of our result is that good approximation architectures result in a low error in approximation using our approach. This error scales linearly with the number of time periods. Similar results for the infinite dimensional discounted case result in a $(1-\alpha)^{-1}$ factor on the upper bound, suggesting a linear dependence on the time horizon cannot be avoided.

We also note that we can improve on this approximation guarantee by using weighted ∞-norm. This would result in an analysis analogous to the proof of Theorem 2 of [Desai *et al.*, 2011]. For the sake of simplicity we do not pursue this direction.

3.5.2 Fluid Approximation

The aim of this section is to show the relation of the SALP method with the *deterministic linear program* formulation also known as the *fluid approximation*. In the case of the Stochastic Assignment problem the deterministic LP formulation is well studied and is known to be asymptotic optimal in the sense we'll make clear soon. This connection helps us show the asymptotic optimality for our procedure.

We build on the result by [Adelman, 2007] for the ALP formulation applied to the problem of network revenue management problem. We show that such analysis can be carried out for the SALP formulation applied to stochastic assignment.

3.5.2.1 Approximation Architecture

The main assumption we make in this section is that we have a small number of types. Thus $|\mathcal{T}|$ is small. With this assumption, as the state space, we use the set $\mathcal{X} = \mathbb{N}^{\mathcal{T}}$. This state space does not change with time. Here for $x \in \mathcal{X}$, x_σ coordinate denotes the count of items of type $\sigma \in \mathcal{T}$.

Recall that our approximation architecture is separable in nature, i.e.

$$\tilde{J}_t(\mathcal{S}_t) = \sum_{\sigma \in \mathcal{S}_t} \langle \kappa_t, \sigma \rangle + \theta_t = \sum_{\sigma \in \mathcal{S}_t} \langle \kappa_t, \Phi(\sigma) \rangle + \theta_t.$$

In this section we make a further assumption about Φ, the *inner* approximation architecture. We assume that inner approximation architecture has one basis function per type. Thus for each type

$\tau \in \mathcal{T}$ we have a basis function of the type $\sigma \mapsto \mathbf{1}_{\sigma=\tau}$, the indicator variable of that type. Let us call this function I_τ. Since $|\mathcal{T}|$ is small, this is a reasonable assumption.

With this state space our approximation architecture is,

$$J(x) = \sum_{\sigma \in \mathcal{T}} x_\sigma^t \kappa_t^\sigma + \theta_t.$$

Here κ^σ is the weight on the basis function I_σ.

If we are dealing with the stochastic assignment problem in particular, to distinguish between the known and the unknown side, we partition the set of type into $\mathcal{T} = \mathcal{A} \cup \mathcal{B}$. Here if $\sigma \in \mathcal{A}$ then it belongs to the known side, else it belongs to the unknown side. For clarity we denote a particular element of $\mathcal{A}$ as α and a particular element in $\mathcal{B}$ as β. Thus our approximation architecture takes the form,

$$J(\mathcal{S}_t) = \sum_{\alpha \in \mathcal{A}} x_\alpha^t \kappa_t^\alpha + \sum_{\beta \in \mathcal{B}} x_\beta^t \kappa_t^\beta + \theta_t.$$

Thus each item of type σ contributes κ^σ to the value function.

3.5.2.2 Deterministic LP

A popular way to devise policies for the stochastic assignment problem with a small space of types is by solving the so-called deterministic LP or the fluid approximation. Consider relaxing the decision problem in the following way,

1. Instead of arrival of a single item that is random, we have fractional amounts of arrivals of items where the arrival amount is the arrival probability of the particular item. Thus if p_β denotes the probability that type β arrives in the original model then in the deterministic LP we assume at each time period p_β amount of β arrives. The amount of β that is not matched immediately leaves the system.

2. We assume that fractional amounts of α on the known side can be assigned to those of β on the unknown side. Let this decision of allocating fractional amount of α to β at time t be noted as $\pi_{\alpha\beta}$.

We will call the amount leftover of a particular type α at time t as Y_α^t. Y^t is the vector of these quantities. Finally let Y^T be the leftover inventory at time T. Such a decision problem can be

solved for optimality using the LP,

$$\begin{aligned}
\text{(P-DLP1)} \triangleq \text{maximize} \quad & \sum_{0\le t<T} \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} \pi^t_{\alpha\beta} w(\alpha,\beta) \\
\text{subject to} \quad & Y^0 = x^0 \\
& Y^t_\alpha - \sum_{\beta\in\mathcal{B}} \pi^t_{\alpha\beta} = Y^{t+1}_\alpha \qquad \forall \alpha\in\mathcal{A},\ 0\le t<T \\
& \sum_{\alpha\in\mathcal{A}} \pi^t_{\alpha\beta} \le p_\beta \qquad \forall \beta\in\mathcal{B},\ 0\le t<T \\
& \pi \ge 0, \quad Y \ge 0.
\end{aligned} \tag{3.14}$$

Here the second constraint along with $Y \ge 0$ implies conservation of the inventory on the known side. The third constraint enforces that we don't assign more β that it is available.

The Y in (P-DLP1) can be eliminated to get the program,

$$\begin{aligned}
\text{(P-DLP2)} \triangleq \text{maximize} \quad & \sum_{\substack{0\le t<T\\ T-1}} \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} \pi^t_{\alpha\beta} w(\alpha,\beta) \\
\text{subject to} \quad & \sum_{t=0} \sum_{\beta\in\mathcal{B}} \pi^t_{\alpha\beta} \le x^0_\alpha \qquad \forall \alpha\in\mathcal{A}, \\
& \sum_{\alpha} \pi^t_{\alpha\beta} \le p_\beta \qquad \forall \beta\in\mathcal{B},\ 0\le t<T \\
& \pi \ge 0.
\end{aligned} \tag{3.15}$$

In this program we allowed π to vary across time periods. But in fact this is not required. Consider the linear program,

$$\begin{aligned}
\text{(P-DLP-TH)} \triangleq \text{maximize} \quad & \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} \Pi_{\alpha\beta} w(\alpha,\beta) \\
\text{subject to} \quad & \sum_{\beta\in\mathcal{B}} \Pi_{\alpha\beta} \le \hat{p}_\alpha, \qquad \forall \alpha\in\mathcal{A} \\
& \sum_{\alpha\in\mathcal{A}} \Pi_{\alpha\beta} \le p_\beta, \qquad \forall \beta\in\mathcal{B} \\
& \Pi \ge 0.
\end{aligned} \tag{3.16}$$

Here $\hat{p}_\alpha$ is the fraction of initial population of type α, i.e.,

$$\hat{p}_\alpha \triangleq \frac{x^0_\alpha}{\mathbf{1}^\top x^0} = \frac{x^0_\alpha}{T}.$$

We call this the *time-homogeneous* version of the DLP.

We note in the following proposition that it suffices to use the optimal Π of (3.16) as the time-homogeneous allocation policy for (3.15).

Proposition 5. *Let Π^* be the optimal for (P-DLP-TH). Then a time homogeneous allocation of $\pi^t = \Pi^*$ is optimal for (P-DLP2).*

Proof. First we show that the search for optimal π in (P-DLP2) can be restricted to time-homogeneous allocations. To this end, note that for any feasible π if

$$\bar{\pi}_{\alpha\beta} \triangleq \frac{1}{T}\sum_{t=0}^{T-1} \pi^t_{\alpha\beta},$$

then $\hat{\pi}^t = \bar{\pi}$ is also feasible for (P-DLP2) with the same objective value. Now note that time homogeneity constraints on (P-DLP2) yield the optimization problem (P-DLP-TH). ■

We have showed that (P-DLP1), (P-DLP2) and (P-DLP-TH) are equivalent. Let us denote the initial inventory on the known side as $x^0_{\mathcal{A}}$. For any particular stochastic assignment problem this is deterministic. Let $\nu^{\text{fl}}(x^0_{\mathcal{A}})$ be the optimal value of the (P-DLP1) (or (P-DLP2), (P-DLP-TH)) divided by $\mathbf{1}^\top x^0_{\mathcal{A}} = T$. Further let $\nu^*(x^0_{\mathcal{A}})$ be the optimal value (P-DP) divided by T. Following is a well-known result for many such allocation problems. The proof follows this general reasoning process.

Theorem 4. *We have that $\nu^{\text{fl}}(x^0_{\mathcal{A}}) \geq \nu^*(x^0_{\mathcal{A}})$. Consider a series of problems with initial inventory $mx^0_{\mathcal{A}}$ for $m \in \{1, 2, \dots\}$, then,*

$$\lim_{m\to\infty} \nu^*(mx^0_{\mathcal{A}}) = \nu^{\text{fl}}(x^0_{\mathcal{A}}) = \nu^{\text{fl}}(kx^0_{\mathcal{A}}), \quad \forall k \geq 1.$$

Proof. Let $\pi^{\mu^*,t}_{\alpha\beta}$ be a random variable in $\{0,1\}$ denoting whether an item of type α was matched with something of type β at time t under the measure defined by the optimal policy μ^*. We will show that, $\xi^t_{\alpha\beta} \triangleq \mathsf{E}[\pi^{\mu^*,t}_{\alpha\beta}]$ is feasible for (3.15) and yields $T\nu^*(x^0_{\mathcal{A}})$ as the objective value. Let,

$$AS^{t,\mu^*}_\beta \triangleq \sum_\alpha \pi^{\mu^*,t}_{\alpha\beta},$$

which is a $\{0,1\}$ denoting whether an item of the type β was assigned at time t. Also let $A^t_\beta \triangleq \mathbf{1}_{\beta_t=\beta}$. Clearly,

$$AS^{t,\mu^*}_\beta \leq A^t_\beta.$$

Taking expectations we get,

$$\mathsf{E}[AS^{t,\mu^*}_\beta] \leq \mathsf{E}[A^t_\beta] = p_\beta.$$

Thus,

$$\begin{aligned}\sum_{\alpha}\xi^t_{\alpha\beta} &= \mathsf{E}\left[\textstyle\sum_\alpha \pi^{\mu^*,t}_{\alpha\beta}\right]\\ &= \mathsf{E}[AS^{t,\mu^*}_\beta] \le \mathsf{E}[A^t_\beta]\\ &\le p_\beta.\end{aligned}$$

Similarly let,

$$AS^{\mu^*}_\alpha \triangleq \sum_{t=0}^{T-1}\sum_{\beta}\pi^{\mu^*,t}_{\alpha\beta}.$$

Clearly,

$$AS^{\mu^*}_\alpha \le x^0_\alpha.$$

Taking expectations we get,

$$\sum_{t=0}^{T-1}\sum_{\beta}\xi^t_{\alpha\beta} \le x^0_\alpha.$$

Finally we have that,

$$\begin{aligned}T\nu^*(x^0) &= T\tfrac{1}{T}\mathsf{E}\left[\sum_{t=0}^{T-1}\sum_{\alpha\beta}\pi^{\mu^*,t}_{\alpha\beta}w(\alpha,\beta)\right]\\ &= \mathsf{E}\left[\sum_{t=0}^{T-1}\sum_{\alpha\beta}\xi_{\alpha\beta}w(\alpha,\beta)\right]\\ &\le T\nu^{\mathrm{fl}}(x^0_{\mathcal{A}}),\end{aligned}$$

where the final inequality follows from the feasibility of ξ in (3.15). Thus we prove that $\nu^{\mathrm{fl}}(x^0)$ is an upper-bound on the optimal value of the MDP $\nu^*(x^0_{\mathcal{A}})$.

For the purpose of proving asymptotic optimality let us set up some notation. We will device a policy that achieves a time averaged reward of $\nu^{\mathrm{fl}}(x^0)$. For a problem scaled with $m \ge 1$, we will call this policy $\mu^{\mathrm{fl},m}$. This policy will keep track of the allocations made up to point t. For a problem scaled by m, let $D^{t,m}_{\alpha\beta}$ be the random number of matches of the type α,β made up to time t by the policy $\mu^{\mathrm{fl},m}$. Our policy will make sure that $D^{t,m}_{\alpha\beta} \le mT\Pi^*_{\alpha\beta}$.

Given that an arrival $\beta_t = \beta$, we will draw a matrix $M^{t,m} \in \{0,1\}^{|\mathcal{A}|\times|\mathcal{B}|}$ such that for $\beta' \neq \beta$, $M^{t,m}_{\alpha\beta'} = 0$. As for the column indexed by β, we draw an $\alpha \in \mathcal{A}$ with probability $\Pi^*_{\alpha\beta}/p_\beta$ and choose not to assign with probability $1 - \sum_\alpha \Pi^*_{\alpha\beta}/p_\beta$. By design, $\{M^{t,m}\}_{0\le t<mT}$ is an i.i.d. sequence and,

$$\mathsf{E}M^{t,m}_{\alpha\beta} = \Pi^*_{\alpha\beta}.$$

Finally assignment is made using a matrix $\pi^{\text{fl},t,m}$ such that, $\pi^{\text{fl},t,m}_{\alpha\beta} = 1$ if and only if, $M^{t,m}_{\alpha\beta} = 1$ and $D^{t,m}_{\alpha\beta} \leq mT\Pi^*_{\alpha\beta}$. Now let,

$$\overline{M}^m \triangleq \sum_{t=0}^{mT-1} M^{t,m},$$

and,

$$\overline{\pi}^{\text{fl},m} \triangleq \sum_{t=0}^{mT-1} \pi^{\text{fl},t,m}.$$

By the nature of this policy it is easy to verify that,

$$\overline{\pi}^{\text{fl},m}_{\alpha\beta} = \min\left(\overline{M}^m_{\alpha\beta}, \lfloor mT\Pi^*_{\alpha\beta}\rfloor\right).$$

It is easy to note that the reward of the policy is,

$$\begin{aligned} R^{\mu^{\text{fl}},m} &\triangleq \tfrac{1}{mT} \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} \overline{\pi}^{\text{fl},m}_{\alpha\beta} w(\alpha,\beta) \\ &= \tfrac{1}{mT} \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} w(\alpha,\beta) \min\left(\lfloor mT\Pi^*_{\alpha\beta}\rfloor, M^m_{\alpha\beta}\right) \\ &= \sum_{\alpha\beta} w(\alpha,\beta) \min\left(\frac{\lfloor mT\Pi^*_{\alpha\beta}\rfloor}{mT}, \frac{1}{mT} M^m_{\alpha\beta}\right) \end{aligned}$$

It is clear that $M^{t,m}_{\alpha\beta}$ is an Bernoulli i.i.d. sequence in t with probability of 1 being $p_\beta\Pi^*_{\alpha\beta}/p_\beta = \Pi^*_{\alpha\beta}$. Thus,

$$\frac{1}{mT} M^m_{\alpha\beta} \to \Pi^*_{\alpha\beta}, \quad \text{a.s.}$$

Also, trivially,

$$\frac{\lfloor mT\Pi^*_{\alpha\beta}\rfloor}{mT} \to \Pi^*_{\alpha\beta}, \quad \text{as } m \to \infty$$

Thus,

$$R^{\mu^{\text{fl}},m} \to \sum_{\alpha\in\mathcal{A},\beta\in\mathcal{B}} \overline{\Pi}^*_{\alpha\beta} w(\alpha,\beta) = \nu^{\text{fl}}(x^0_{\mathcal{A}}), \quad \text{a.s.}$$

By bounded convergence we can take expectation, thus,

$$\nu^{\mu^{\text{fl}}}(mx_0) \triangleq \mathsf{E} R^{\mu^{\text{fl}},m} \to \nu^{\text{fl}}(x^0_{\mathcal{A}}).$$

By time homogeneous nature of the DLP (3.16), we have that $\nu^{\text{fl}}(x^0_{\mathcal{A}}) = \nu^{\text{fl}}(kx^0_{\mathcal{A}})$ for all $k \geq 1$. Thus we have that,

$$\nu^{\mu^{\text{fl},m}}(mx^0_{\mathcal{A}}) \leq \nu^*(mx^0_{\mathcal{A}}) \leq \nu^{\text{fl}}(mx^0_{\mathcal{A}}) = \nu^{\text{fl}}(x^0_{\mathcal{A}}).$$

This proves asymptotic optimality. ■

3.5.2.3 SALP formulation

For the stochastic assignment case let us write down our optimization problem. Firstly, we not that for a given initial inventory x^0 the number of possible states that are visited is finite. For this purpose note that $T \leq \sum_{\alpha \in \mathcal{A}} x^t_\alpha$.

We consider the optimization problem (3.9) for the Stochastic Assignment problem by assuming the approximation architecture discussed earlier.

$$
\begin{aligned}
\text{(P-SALP-SA)} \triangleq \text{minimize} \quad & \sum_{\alpha \in \mathcal{A}} x^0_\alpha \kappa^\alpha_t + \sum_{\beta \in \mathcal{B}} p_\beta \kappa^\beta_t + \theta_0 + (1+\gamma) \sum_{t=0}^{T-1} \mathsf{E}_{\mu^*}[s_t(x^t)] \\
\text{subject to} \quad & \sum_{\alpha' \in \mathcal{A}} x^t_{\alpha'} \kappa^{\alpha'}_{t+1} - \kappa^\alpha_{t+1} + \sum_{\beta \in \mathcal{B}} p_\beta \kappa^\beta_{t+1} + \theta_{t+1} + w(\alpha, \beta) \\
& \qquad \leq \sum_{\alpha \in \mathcal{A}} x^t_\alpha \kappa^\alpha_t + \kappa^\beta_t + \theta_t + s_t(x^t), \quad \forall 0 \leq t < T-1, x^t \in \mathcal{X}_t, \alpha \in A(x^t) \\
& \sum_{\alpha' \in \mathcal{A}} x^t_{\alpha'} \kappa^{\alpha'}_{t+1} + \sum_{\beta \in \mathcal{B}} p_\beta \kappa^\beta_{t+1} + \theta_{t+1} \\
& \qquad \leq \sum_{\alpha \in \mathcal{A}} x^t_\alpha \kappa^\alpha_t + \kappa^\beta_t + \theta_t + s_t(x^t), \quad \forall 0 \leq t < T-1, x^t \in \mathcal{X}_t \\
& w(\alpha, \beta) \leq \sum_{\alpha \in \mathcal{A}} x^{T-1}_\alpha \kappa^\alpha_{T-1} + \kappa^\beta_{T-1} + \theta_{T-1} + s_{T-1}(x^{T-1}), \quad x^{T-1} \in \mathcal{X}_{T-1}, \alpha \in A(x^{T-1}) \\
& 0 \leq \sum_{\alpha \in \mathcal{A}} x^{T-1}_\alpha \kappa^{T-1}_\alpha + \kappa^{T-1}_\beta + \theta_{T-1} + s_{T-1}(x^{T-1}), \quad x^{T-1} \in \mathcal{X}_{T-1}.
\end{aligned}
\tag{3.17}
$$

Here $A(x^t) \triangleq \{\alpha \in \mathcal{A} : x^t_\alpha > 0\}$. We label states at time t by the pairs $(x^t_{\mathcal{A}}, \beta)$. This is done to denote which β arrived at t and the inventory on the known side is $x^t_{\mathcal{A}}$. We have a constraint per state action pair for each time period. At each time period t, if the current state is $(x^t_{\mathcal{A}}, \beta)$, we can match the β with some $\alpha \in A(x^t)$ or choose not to match anything. (P-SALP-SA) has a constraint per state-action pair. In the optimization problem the first set of constraints are for the case when $\alpha \in A(x^t)$ is allocated to β at time $t < T-1$. The second set of constraints are for the case when we choose not to allocate. The third and fourth set of constraints are for the boundary case of $t = T-1$. Here the continuation value does not appear in the Bellman constraints.

Let us call the optimal value of the program averaged across time periods (i.e. divided by T) as $\nu^{\text{salp},\gamma}(x^0)$. Further let $\nu^{\text{alp}}(x^0)$ be the value of the program with $\gamma = \infty$, which gives us the ALP formulation.

Theorem 5. *We have that* $\nu^*(x^0) \leq \nu^{\text{salp},\gamma}(x^0) \leq \nu^{\text{alp}}(x^0) \leq \nu^{\text{fl}}(x^0)$ *and,*

$$\lim_{m\to\infty} \nu^{\text{salp},\gamma}(mx^0) = \lim_{m\to\infty} \nu^{\text{alp},\gamma}(mx^0) = \nu^{\text{fl}}(x^0).$$

Proof. The second claim clearly follows from the first claim and Theorem 4. We will prove the first claim by Lagrangian duality. Let $X_{t,x^t_{\mathcal{A}},\alpha,\beta}$ be the corresponding dual variable corresponding to the constraint for state x_t at time t in which we choose α with β. Also let $X^{\text{nm}}_{t,x^t_{\mathcal{A}},\beta}$ be the dual variable corresponding to the constraint for state $(x^t_{\mathcal{A}}, \beta)$ at time t in which we choose to not match anything.

First consider the constraint in the dual problem corresponding to the variable θ_0,

$$1 = \sum_{x^0_{\mathcal{A}},\alpha,\beta} X_{0,x^0_{\mathcal{A}},\alpha,\beta} + \sum_{x^0_{\mathcal{A}},\beta} X^{nm}_{0,x^0_{\mathcal{A}},\beta}.$$

Similarly, for θ_t, with $T-1 > t > 0$ we get constraints of the type,

$$\sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + \sum_{x^t_{\mathcal{A}},\beta} X^{\text{nm}}_{t,x^t_{\mathcal{A}}} = \sum_{x^{t+1}_{\mathcal{A}},\alpha,\beta} X_{t+1,x^{t+1}_{\mathcal{A}},\alpha,\beta} + \sum_{x^{t+1}_{\mathcal{A}},\beta} X^{\text{nm}}_{t+1,x^{t+1}_{\mathcal{A}},\beta}.$$

Thus from the θ's we get constraints,

$$1 = \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + \sum_{x^t,\beta} X^{\text{nm}}_{t,x^t_{\mathcal{A}},\beta},$$

for all $T-1 \geq t \geq 0$.

Now from κ^β_0 we get the constraint,

$$p_\beta = \sum_{x^0_{\mathcal{A}},\alpha,\beta} X_{0,x^0_{\mathcal{A}},\alpha,\beta} + \sum_{x^0_{\mathcal{A}},\beta} X^{\text{nm}}_{0,x^0}$$

Similarly the κ^β_t constraint yields,

$$\left(\sum_{x^{t-1}_{\mathcal{A}},\alpha,\beta} X_{t-1,x^{t-1}_{\mathcal{A}},\alpha,\beta} + \sum_{x^{t-1}_{\mathcal{A}},\beta} X^{\text{nm}}_{t,x^{t-1},\beta}\right) p_\beta = p_\beta = \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + \sum_{x^t_{\mathcal{A}},\beta} X^{\text{nm}}_{t,x^t_{\mathcal{A}}}.$$

As for the $\kappa\alpha_t$s, first note that x^0_α is constant for any x^0 and α. Thus the κ^α_0 gives us a redundant constraint of the type,

$$\left(\sum_{x^0_{\mathcal{A}},\alpha',\beta} X_{0,x^0_{\mathcal{A}},\alpha',\beta} + \sum_{x^0_{\mathcal{A}},\beta} X^{\text{nm}}_{0,x^0_{\mathcal{A}},\beta}\right) x^0_\alpha = x^0_\alpha.$$

κ_1^α yields the constraint,

$$\left(\sum_{x^0_{\mathcal{A}},\alpha',\beta} X_{0,x^0_{\mathcal{A}},\alpha',\beta} + \sum_{x^0_{\mathcal{A}},\beta} X^{\text{nm}}_{0,x^0_{\mathcal{A}},\beta}\right) x^0_\alpha - \sum_{x^0_{\mathcal{A}},\beta} X_{0,x^0_{\mathcal{A}},\alpha,\beta} = x^0_\alpha - \sum_{x^0_{Ascr},\beta} X_{0,x^0_{\mathcal{A}},\alpha,\beta}$$
$$= \sum_{x^1_{\mathcal{A}},\alpha',\beta} X_{1,x^1_{\mathcal{A}},\alpha',\beta} x^1_\alpha + \sum_{x^1_{\mathcal{A}}} X^{\text{nm}}_{1,x^1_{\mathcal{A}},\beta} x^1_\alpha.$$

Similarly, for $T-1 > t > 0$ for any α we get the constraint,

$$\sum_{x^t_{\mathcal{A}},\alpha',\beta} X_{t,x^t_{\mathcal{A}},\alpha',\beta} x^t_\alpha + \sum_{x^t_{\mathcal{A}},\beta} X^{\text{nm}}_{t,x^t,\beta} x^t_\alpha - \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t,\alpha,\beta} = \sum_{x^{t+1}_{\mathcal{A}},\alpha',\beta} X_{t+1,x^{t+1}_{\mathcal{A}},\alpha',\beta} x^{t+1}_\alpha + \sum_{x^{t+1}_{\mathcal{A}},\beta} X^{\text{nm}}_{t+1,x^{t+1}_{\mathcal{A}},\beta} x^{t+1}_\alpha.$$

Let Y^t be a sequence of vectors such that,

$$Y^t_\alpha = \sum_{x^t_{\mathcal{A}},\alpha',\beta} X_{t,x^t_{\mathcal{A}},\alpha',\beta} x^t_\alpha + \sum_{x^t_{\mathcal{A}},\beta} X^{\text{nm}}_{t,x^t_{\mathcal{A}}} x^t_\alpha,$$

then we have the constraints,

$$Y^t_\alpha - \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} = Y^{t+1}_\alpha.$$

for $0 \le t < T-1$, with $Y^0_\alpha = x^0_\alpha$.

For each $s_t(x^t)$ we get the constraint,

$$\sum_\alpha X_{t,x^t_{\mathcal{A}},\alpha,\beta} + X^{\text{nm}}_{t,x^t_{\mathcal{A}},\beta} \le (1+\gamma) p^*_t(x_t).$$

Here $p^*_t(x_t)$ is the probability of ending up in state x^t at time t with the policy μ^*.

Finally the objective is,

$$\sum_{\alpha,\beta} w(\alpha,\beta) \sum_{t=0}^{T-1} \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta}.$$

Thus, the Lagrangian dual of this optimization problem is,

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_{\alpha,\beta} w(\alpha,\beta) \sum_{t=0}^{T-1} \sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} & \\
\text{subject to} & \displaystyle\sum_{x^t_{\mathcal{A}},\alpha,\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} = 1 & \forall 0 \le t < T-1 \\
& p_\beta = \displaystyle\sum_{x^t_{\mathcal{A}},\alpha} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + \sum_{x^t_{\mathcal{A}}} X^{\mathrm{nm}}_{t,x^t_{\mathcal{A}},\beta} & \forall 0 \le t \le T-1, \forall \beta \in \mathcal{B} \\
& Y^t_\alpha = \displaystyle\sum_{x^t,\alpha',\beta} X_{t,x^t_{\mathcal{A}},\alpha',\beta} x^t_\alpha + \sum_{x^t_{\mathcal{A}},\beta} X^{\mathrm{nm}}_{t,x^t_{\mathcal{A}},\beta} x^t_\alpha & \forall 0 < t < T, \forall \alpha \in \mathcal{A} \\
& Y^0_\alpha = x^0_\alpha & \forall \forall \alpha \in \mathcal{A} \\
& Y^{t+1}_\alpha = Y^t_\alpha - \displaystyle\sum_{x^t_{\mathcal{A}},\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta}, & \forall 0 \le t < T-1 \\
& \displaystyle\sum_{\alpha} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + X^{\mathrm{nm}}_{t,x^t_{\mathcal{A}},\beta} \le (1+\gamma) p^*_t(x_t) & \forall 0 \le t < T-1, x^t \in \mathcal{X}_t \\
& X \ge 0, X^{\mathrm{nm}} \ge 0. &
\end{array}
\tag{3.18}
$$

$\nu^* \le \nu^{salp,\gamma}$ follows from Proposition 4. To show that $\nu^{salp,\gamma} \le \nu^{fl}$ and $\nu^{alp} \le \nu^{fl}$, we construct a feasible solution of (3.15) from one of (3.18) which yields the same objective. Let,

$$\pi^t_{\alpha\beta} = \sum_{x^t_{\mathcal{A}}} X_{t,x^t_{\mathcal{A}},\alpha,\beta}.$$

Check that,

$$\sum_{\alpha} \pi^t_{\alpha\beta} \le p_\beta,$$

follows from the second constraint and non-negativity of X^{nm}. Now we sum the constraints

$$Y^{t+1}_\alpha = Y^t_\alpha - \sum_{x^t \in \alpha,\beta} X_{t,x^t,\alpha,\beta},$$

to obtain,

$$
\begin{aligned}
x^0_\alpha &= \sum_{t=0}^{T-1} \sum_{x^t_{\mathcal{A}},\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} + Y^T_\alpha \\
&\ge \sum_{t=0}^{T-1} \sum_{x^t_{\mathcal{A}},\beta} X_{t,x^t_{\mathcal{A}},\alpha,\beta} \\
&= \sum_{t=0}^{T-1} \sum_{\beta} \pi^t_{\alpha\beta}.
\end{aligned}
$$

Thus $\pi^t_{\alpha\beta}$ is feasible for (3.15). It is easy to note that it has yields the same objective value. Also note that increase in γ monotonically relaxes the problem. This means that the objective value is non-decreasing as a function of γ. Thus yielding the inequality $\nu^{\text{salp}}(\gamma) \leq \nu^{\text{alp}}$. ■

3.6 Experiments

This section outlines the numerical experiments performed to assess the performance of our approach. Continuing with the examples in Section 3.3, we consider three different models of kidney allocation problems:

1. **Stochastic Assignment (SA):** A stochastic assignment problem where T patients in need of kidneys must be matched with T arriving kidneys. Assumption is that patients do not perish and kidneys disappear immediately if not matched.

2. **General Bipartite Assignment (GBA):** A generalization of the this kidney allocation model where kidneys and patients follow more general arrival and departure processes. This is a richer model for the problem of allocating cadeveric kidneys to patients.

3. **Cycle Assignment (CA):** Kidney paired donation, a model where incompatible donor recipient pairs must be matched with each other in cyclic transfer of kidneys.

Since the first two models have a matching structure is different from the third we use the term *bipartite assignment* (BA) for the first two cases and *cycle assignment* (CA) for the last one.

3.6.1 Compatibility and Types

When a kidney, or any other organ for that matter, is transplanted from a donor to a recipient, there is a chance of rejection of the organ by recipients immune system. This rejection will have a low chance if the immune system cannot distinguish between its own cells and the foreign cells of the transplant. The chance of rejection, apart from blood types of the donor and the recipient, depends on HLA (Human leukocyte antigen) types of the pair. HLA type of a person is a combination of 6 proteins that are present on every cell. These appear in three pairs denoted by A, B and DR. We will call these proteins $\mathcal{PR} \triangleq \{A1, A2, B1, B2, DR1, DR2\}$.

The graft (transplanted organ) survival depends on factors such match of blood and HLA types and the age of the donor. We will instead consider a simplified model where we will only consider the HLA types of donor and recipient. Let the HLA type of any person be denoted by a 6-dimensional categorical vector $t = (t_{\text{A1}}, t_{\text{A2}}, t_{\text{B1}}, t_{\text{B2}}, t_{\text{DR1}}, t_{\text{DR2}})$. Each dimension of this vector can take one of roughly 30 values. Thus the number of HLA types is approximately $30^6 \approx 7.3 \times 10^8$. Let the set of values a protein p can take be denoted by $\mathcal{L}_p$. We call the space of HLA types of any individual by $\mathcal{L} \triangleq \Pi_{p \in \mathcal{PR}} \mathcal{L}_p$.

Our allocation scheme will try to maximize the number of life-years saved by performing transfer of kidneys. It is known that the match of every HLA protein approximately adds 5 years to the recipient of the kidney [G *et al.*, 1998]. Thus if donor with type $d \in \mathcal{L}$ is matched with the recipient $r \in \mathcal{L}$ then the number of life-years saved is proportional to,

$$l(d, r) \triangleq \sum_{p \in \mathcal{PR}} \mathbf{I}_{d_p = r_p}.$$

This measure of compatibility is consistent with the earlier studies of kidney allocation [Zenios *et al.*, 2000; Su and Zenios, 2005].

In case of the BA problems the space of types $\mathcal{T}_{\text{ba}}$ is merely the space of HLA types, i.e. $\mathcal{L}$. The compatibility function w_{ba} is the same as the function l.

On the other hand for CA problems each item has a donor recipient pair. Thus the type of a pair is the combination of types of the donor and the recipient. The space of types in this case is thus $\mathcal{T}_{\text{ca}} \triangleq \mathcal{L} \times \mathcal{L}$. The compatibility function is given as follows. Let the donor of pair $p_1 = (p_1^d, p_1^r) \in \mathcal{L} \times \mathcal{L}$ be assigned to the recipient of pair $p_2 = (p_2^d, p_2^r) \in \mathcal{L} \times \mathcal{L}$. Then,

$$w_{\text{ca}}(p_1, p_2) \triangleq l(p_1^d, p_2^r).$$

Our model requires us to specify a distribution F on $\mathcal{T}$. We use the empirical distribution in [Zenios *et al.*, 2000] for this purpose. In the BA case, for each new item, each protein has a random type that is independent of the other proteins. For the CA case, the tissue type of the donor and recipient are drawn independently of the other from the distribution independent distribution just described.

3.6.2 Approximation Architecture

We use a separable approximation architecture mentioned in the Section 3.4.2. This requires us to specify the basis functions for the *inner* approximation architecture over the space of types. First consider the BA case. Recall that the space of types in this case is $\mathcal{T}_{\text{ba}} = \mathcal{L}$. In this case we have a basis function for each protein $p \in \mathcal{PR}$ and its type $v \in \mathcal{L}_p$. A particular basis function $\Phi^{\text{ba}}_{pr,v}$ is defined as,

$$\Phi^{\text{ba}}_{p,v}(t) = \mathbf{1}_{t_p=v}.$$

Note that for this case, we'd have roughly $30 \cdot 6 = 180$ basis functions.

As for the CA case for each protein $p \in \mathcal{PR}$ and its type $v \in \mathcal{L}_p$, we'd have two basis function $\Phi^{\text{ca}}_{p,v,d}$ for the donor and $\Phi^{\text{ca}}_{p,v,d}$ for the recipient. They are defined as,

$$\Phi^{\text{ca}}_{p,v,d}(pr) = \mathbf{1}_{pr^d_p=v},$$

and,

$$\Phi^{\text{ca}}_{p,v,r}(pr) = \mathbf{1}_{pr^r_p=v}.$$

Thus in this case we have roughly 360 basis functions.

3.6.3 Arrival and Departure Dynamics

Here we elaborate on the arrival and departure processes in each of the cases we consider.

- **Stochastic Assignment:** In this case, the arrival and departure process are specified by the model. For a problem with T decision epochs we have T patients waiting for assignment of kidneys. The types of these kidneys are known to the algorithm. Thus the initial population, which consists of these patients is known. We sample this initial population of types as i.i.d. with distribution F, the empirical distribution from data given in [Zenios *et al.*, 2000]. Cadaveric kindeys arrive one at a time and perish if not matched. Types of the arriving kidneys follows a i.i.d. process with distribution F.

- **General Bipartite Assignment:** The the number of the initial population in this model comes from a geometric distribution. We call this parameter the *initial population mean* (IPM). For our experiments, this parameter is 10.

Each of these items has a tissue type that comes from the distribution F on $\mathcal{L}$. Further, whether the item is a donor or a recipient is decided by flipping a biased coin. We call the probability of a particular item being a patient as *patient bias* (PB). In practice, it is observed that the number of patients waiting for the allocation vastly outnumber the number of potential donors in the pool. This is reflected in the by our choice of PB as 0.85.

Arrival process in each time-period follows similar dynamics as the sampling of the initial population. The number of arrival follows a geometric distribution. The mean of the distribution is the called the *mean arrival rate* (MAR). We set the MAR to 5. Again each of the items has a tissue type sampled from F and is a patient with probability PB.

The departure rate of a particular item depends on whether it is a patient or a donor. We call the rate of departure of a donor as *donor departure rate* (DDR) and *patient departure rate* (PDR). Donors are considered cadaveric and must be matched immediately upon arrival. Thus DDR is set to 1.0. On the other hands patients are have a geometric timespan of 20 time-periods. Thus the PDR is set at 0.05.

- **Cycle Assignment:** The arrival and departure dynamics of the cycle assignment case are similar to the GBA example. The initial population is unknown and the total number of it is sampled according to a geometric distribution. The IPM is again set at 10. At each point of time a geometric number of items are sampled. The MAR is set at 5.

  Each item has a donor and a recipient associated with it. The tissue type of each one of them is sampled from F and is independent of the other.

  Each item has a constant departure rate and it is set at 0.05 which is the same as PDR from the GBA case. As opposed to the GBA case where the donors are cadaveric, CA assumes that donors are healthy and remain in the system forever if not matched.

3.6.4 Policies

Having described the model we now describe the policies we use to perform the dynamic matchings.

- **Myopic**: Myopic is the most natural policy to consider. This policy ignores the future and takes decisions that maximize the immediate reward that can be obtained. As mentioned

earlier, the myopic policy in general requires us to solve a linear program at each step. In the SA case this LP is a trivial maximization problem where if β_t is the arriving kidney and $\mathcal{A}_t$ is the set of remaining patients, the patient to be matched is given by $\text{argmax}_{\alpha \in \mathcal{A}_t} w(\alpha, \beta)$. For the GBA model we must solve a weighted bipartite matching problem with weights $w(\alpha, \beta)$ for $\alpha \in \mathcal{A}_t$ and $\beta \in \mathcal{B}_t$. We use the Hungarian algorithm to perform this matching. In the CA case we use a commercial LP solver to find the assignment that maximizes the matching reward.

- **Offline:** This type of allocation refers to the optimal allocation when all the randomness is known apriori. In this way, offline allocation gives a perfect information upper-bound on the value of the problem.

  For the SA case if all the arrivals are known apriori, then we would solve a size T perfect bipartite matching problem to find the allocation of patients with all the arriving kidneys. At each point of time we match the arriving type with the patient specified by this matching.

  For the GBA we use the model of arrivals and departures to sample the arrival and the departure time periods of each item. Then we produce a bipartite graph where all the patient items are on one side and the donors on the other. We connect two items, a patient and a donor, if both ever exist in the same time-period in that sample path. We solve the bipartite matching problem to find the optimal allocation.

  Finally for the CA case let $\hat{\mathcal{S}}$ be the all the items that are encountered in a particular sample path. Further let a_σ and d_σ be the arrival and departure (if unmatched) epoch of each of the items. To offline assignment we must solve the following integer program:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{t} \sum_{\sigma, \sigma' \in \mathcal{S}} \pi^t_{\sigma\sigma'} w(\sigma, \sigma') \\
\text{subject to} \quad & \sum_{\sigma' \in \hat{\mathcal{S}}/\sigma} \pi^t_{\sigma\sigma'} = \sum_{\sigma' \in \hat{\mathcal{S}}/\sigma} \pi^t_{\sigma'\sigma} && \forall 0 \le t \le T-1,\ \sigma \in \hat{\mathcal{S}} \\
& \pi^t_{\sigma\sigma'} = \pi^t_{\sigma'\sigma} = 0 && \forall t \notin [a_\sigma, d_\sigma],\ \sigma' \in \hat{\mathcal{S}} \\
& \sum_{t=0}^{T-1} \sum_{\sigma' \in \mathcal{S}/\sigma} \pi^t_{\sigma\sigma'} \le 1 && \forall \sigma \in \hat{\mathcal{S}} \\
& \pi^t_{\sigma\sigma'} \in \{0, 1\} && \forall 0 \le t \le T-1,\ \sigma, \sigma' \in \hat{\mathcal{S}}.
\end{aligned}
\tag{3.19}
$$

Note that relaxing the $\{0,1\}$ constraint still gives an upper bound to the problem. We use this relaxed problem for the upper bound.

- **Dual Value Regression:** We consider an alternative approach to estimate the *value* of each of the items. This approach applies to the bipartite allocation problems. Let $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ be the patients and donors, respectively, that are encountered in a particular sample path. Also let $\hat{w}(\alpha,\beta)$ be the function that equals $w(\alpha,\beta)$ if α and β ever co-exist in a time-period, or else it is $-\infty$. The offline policy computes:

$$\begin{aligned} \text{maximize} \quad & \sum_{\alpha\in\hat{\mathcal{A}},\beta\in\hat{\mathcal{B}}} \pi_{\alpha\beta}\hat{w}(\alpha,\beta) & \\ \text{subject to} \quad & \sum_{\beta\in\hat{\mathcal{B}}} \pi_{\alpha\beta} \le 1 & \forall\alpha\in\hat{\mathcal{A}} \\ & \sum_{\alpha\in\hat{\mathcal{B}}} \pi_{\alpha\beta} \le 1 & \forall\beta\in\hat{\mathcal{B}} \\ & \pi_{\alpha\beta}\in[0,1] & \forall\alpha\in\hat{\mathcal{A}},\ \beta\in\hat{\mathcal{B}}. \end{aligned} \tag{3.20}$$

This has the following Lagrangian dual:

$$\begin{aligned} \text{minimize} \quad & \sum_{\alpha\in\hat{\mathcal{A}}} \lambda_\alpha + \sum_{\beta\in\hat{\mathcal{B}}} \lambda_\beta & \\ \text{subject to} \quad & \lambda_\alpha + \lambda_\beta \ge w(\alpha,\beta) & \forall\alpha\in\hat{\mathcal{A}},\ \beta\in\hat{\mathcal{B}} \\ & \lambda \ge 0. & \end{aligned} \tag{3.21}$$

Now the optimal dual values can be interpreted as the contribution of a particular item α or β to the objective. Thus by solving a dual problem for each sample path, we can have an estimate of the contribution of each item in the sample path to the total reward obtained in the offline policy. We use this estimate to perform a regression on the space of types. In particular fit a function that is a linear combination of some basis functions. We use a 2-norm error criterion which gives us a least-squares estimate of the value function.

After we obtain the estimate we use this approximation of the value, say $v(\sigma)$, in the approximation architecture outlined in Section 3.4.2 in place of $\langle\kappa_t,\sigma\rangle$. This gives us a tractable policy whose evaluation is similar to the one used by our approach.

- **Our Approach**: Finally we consider our approach described in the paper. As described in Section 3.4 our approach requires us to solve an optimization problem. We will use the

Problem Size	**greedy**	**regression**	**SALP-γ**				**Offline**
			greedy	γ	offline	γ	
25	2.058	2.138 (0.001)	2.134 (0.002)	1	2.143 (0.002)	0.01	2.593
50	2.326	2.427 (0.002)	2.426 (0.002)	1	2.424 (0.002)	0.01	2.708
75	2.496	2.607 (0.001)	2.604 (0.001)	1	2.586 (0.001)	1	2.839
100	2.641	2.741 (0.001)	2.734 (0.002)	1	2.730 (0.002)	1	2.967
125	2.712	2.829 (0.001)	2.828 (0.002)	1	2.825 (0.001)	1	3.095
150	2.772	2.911 (0.001)	2.912 (0.001)	1	2.910 (0.001)	1	3.208

Table 3.1: Performance results for the stochastic assignment case for various problem size and algorithms.

stochastic subgradient descent (SSGD) to solve the optimization problem. The input required the method are: the approximation architecture, the sampling distribution, the step-size selection and γ. We use the separable approximation architecture with indicators of tissue types as the basis functions. To be consistent with the theory we pick $\gamma = 1$ for all the examples we consider.

A sampling distribution gives us a way to sample a sequence of state trajectories that the stochastic gradient descent will take as an input. For simplicity we consider the greedy distribution and the offine distribution. For the bipartite assignment problems we use the offline policy to sample states and for the cycles assignment we use the greedy policy to sample states.

For the stochastic subgradient descent we use the step-size of the form $A/(B+\omega)$, where ω is the index of the sample. With crude trial-and-error search we use the parameter $A = 0.1$ and $B = 1000.0$. As mentioned earlier, we keep tract of the *best* set of parameters encountered for any path of the SSGD algorithm. Our criterion to keep track of this is the evaluation of the policy obtained by these weights. This performance is evaluated by Monte Carlo simulation on 300 sample paths sampled with the same seed. Since this evaluation is expensive, we perform this after each 20 updates.

Problem Size	**greedy**	**regression**	**SALP-γ**				**Offline**
			greedy	γ	offline	γ	
25	1.200	1.430 (0.002)	1.439 (0.006)	1	1.429 (0.007)	1	1.641
50	1.216	1.498 (0.002)	1.510 (0.005)	1	1.513 (0.004)	1	1.752
75	1.212	1.509 (0.001)	1.557 (0.005)	1	1.561 (0.005)	1	1.825
100	1.207	1.426 (0.001)	1.570 (0.005)	1	1.570 (0.006)	1	1.865
125	1.205	1.313 (0.002)	1.563 (0.005)	1	1.561 (0.004)	1	1.877
150	1.207	1.278 (0.002)	1.496 (0.004)	1	1.498 (0.004)	1	1.895

Table 3.2: Performance results for the general bipartite matching case for various problem size and algorithms.

Problem Size	**greedy**	**SALP-γ**		**Offline**
		greedy	γ	
25	5.624	6.043 (0.006)	1	7.852
50	5.277	5.869 (0.004)	1	7.598
75	5.157	5.805 (0.003)	1	7.510
100	5.170	5.832 (0.001)	1	7.558
125	5.153	5.830 (0.003)	1	7.568
150	5.158	5.835 (0.002)	1	7.566

Table 3.3: Performance results for the the kidney cycles case for various problem size and algorithms.

Chapter 4

Optimal A-B Testing

4.1 Introduction

The prototypical example of an 'A-B' test is the design of a clinical trial where one must judge the efficacy of a treatment or drug relative to some control. In a different realm, A-B testing today plays an increasingly pivotal role in e-commerce, ranging from the optimization of content and graphics for online advertising, to the design of optimal layouts and product assortments for webpages. E-commerce properties will even use A-B testing as a means of finding the best third party vendor for a specific service on their website (such as, say, recommendations or enterprise search).

A natural approach to A-B testing is to independently, and with equal probability, assign each subject (or sample) to either the treatment or control groups. Following such a randomized allocation, the benefit of the treatment relative to the control can be estimated from the outcomes of subjects in the two groups. The notion of a sample here can range from a patient in the clinical trial setting to a web-surfer or impression in the e-commerce setting. The notion of a treatment can range from an actual medical treatment in the clinical trial setting to an action such as showing a specific ad in the e-commerce setting. While randomized allocation is simple and can easily be shown to yield unbiased estimates of the treatment effect under a minimal set of assumptions, the *efficiency* of this procedure (or the number of samples needed to get a statistically significant estimate of the treatment effect) can prove onerous in practice. To see why consider the following challenges:

1. Limited Samples: In the clinical trial setting, the size of the sample set is limited for a number of reasons. As an example, the cost of managing a single subject through a clinical trial is tens of thousands of dollars, see e.g. [Steensma and Kantarjian, 2014]. In the e-commerce setting, one may need to conduct many thousands of A-B tests in an ongoing fashion. As an example, consider an advertising firm that uses A-B testing on live impressions (i.e., web-surfers) to mechanically decide the appropriate messaging, text size, font, color etc. for the creatives it generates for an online advertising campaign. In this domain, a reduction in the number of 'samples' needed to learn can, due to scale, result in dramatic, continual, cost savings.

2. Confounding Effects: Running counter to the need for quick inference, the impact of a particular treatment (or design decision) may be marred by a potentially large number of covariates, or alternative explanatory variables that are unrelated to the choice of treatment. The presence of these covariates makes the inference of the treatment effect more challenging, since the difference in outcome of the treatment and control groups might be on the account of a lack of 'balance' in the covariates in the two groups. While the law of large numbers assures us that a large enough sample size will 'wash out' the impact of this imbalance of covariates, this number of samples may grow exceedingly large when the number of covariates is large and/ or the treatment effect is small.

3. 'Small' Treatment Effects: Similar to the covariate imbalance issue above, the incremental impact of the treatment under study may be relatively 'small'. More precisely, if one imagined a model where the outcome is additively impacted by the treatment and exogenous noise, we expect the number of samples required to discern the treatment from noise to grow quadratically with the ratio of the standard deviation of the exogenous noise to the treatment effect. To (heuristically) see why, observe that if S_n is the sum of n independent, zero mean random variables, with standard deviation σ, and $\theta > 0$ is some constant, then by the CLT, we expect
$$\mathbb{P}\left(\left|\frac{S_n}{n}\right| \geq \theta\right) \sim 2\Phi\left(\frac{\theta\sqrt{n}}{\sigma}\right)$$
suggesting that in order to differentiate a treatment effect θ from exogenous noise with standard deviation σ, we needs on the order of σ^2/θ^2 samples.

Addressing theses challenges, motivates considering the careful design of such A-B tests. In particular, given a collection of subjects, some of whom must be chosen for treatment, and others

assigned to a control, we would like an assignment that 'cancels' the impact of covariates. This in turn would conceptually address the two challenges discussed above and yield an 'efficient' test.

Given the broad applicability of an efficient A-B test, it is perhaps not surprising that a large body of literature within the statistical theory of the design of experiments has considered this very problem, starting with the nearly century old work of [Fisher, 1935]. While we defer a review of this substantial literature to Section 4.1.2, a very popular approach to dealing with the problem of achieving balance to cancel the impact of covariates is by using 'stratification'. In this approach, the subjects are divided into a number of groups based on the covariates. In other words, the covariate space is divided into a number of regions and subjects whose covariates lie in a certain region are grouped together. Further, each of the groups is randomly split to be allocated to the treatment or the control. Unfortunately, stratification does not scale gracefully with the number of covariates since the number of groups required in stratification will grow exponentially with the dimension. Another natural idea would be to 'match' subjects with similar covariates, followed by assigning one member of a match to the treatment and the other to the control. Such a design would try to mimic an idealistic scenario in which, for n subjects under the experiment, we have $n/2$ pairs of 'twins'. If the matched subjects are indeed close to each other in the space of covariates, we would have that the distribution of covariates in the treatment and control is close to each other, which would cancel out the effect of these covariates. While this latter approach does allow us to consider a large number of covariates, the literature only appears to present heuristics motivated by these ideas.

To add a further challenge beyond those already discussed, an additional (and very important) requirement apparent from the applications above is that the process of allocating subjects (or impressions) to a particular treatment (or creative) must be made *sequentially, in an online fashion.* Again, there is a literature on dynamic allocation starting with seminal work by [Efron, 1971]. This *online* variant of the A-B testing problem is also a well studied one and while the literature pertaining to the problem abounds with heuristics, an efficient to implement, optimal 'solution' remains unknown. In fact, the literature surprisingly does not consider the design of an 'optimal' online allocation of subjects to treatments – or online A-B testing in our parlance – as a principled dynamic optimization problem.

The principle contribution of this paper is to present the first provably near-optimal algorithm

for online A-B testing that applies to a canonical class of treatment-effect models. As a secondary contribution, we also show that that the important 'offline' variant of the problem also admits an efficient optimal algorithm for the same canonical class of treatment-effect models and tightly characterize the value of optimization in that setting.

4.1.1 This Paper

Our approach, in a nutshell, is to formulate online A-B testing as a (computationally challenging) dynamic optimization problem and develop approximation and exact algorithms for the same. In particular, the present paper considers the setting where a subject's response is linear in the treatment and covariates; as we discuss later, this is a canonical model and ubiquitous in the literature on experiment design. We consider the problem of minimizing the variance of our estimate of the treatment effect by optimally allocating subjects to either the treatment or control group, and designing an appropriate estimator. We formulate this problem as a dynamic optimization problem and make the following contributions:

1. We characterize the value of optimized allocations relative to randomized allocations and show that this value grows large as the number of covariates grows. In particular, we show that there is a lot to be gained from 'optimizing' the process of online A-B testing relative to the simple randomized trials that are the mainstay of modern A-B testing. We show that the gains achieved using optimization are more dramatic when the number of covariates is close to the number of trials, a scenario quite pertinent to the e-commerce setting, for example.

2. In the offline setting, i.e., where the allocation can be made after observing all subjects, we show that the problem can be solved efficiently by using as a subroutine a generalization of the MAX-CUT SDP relaxation of [Goemans and Williamson, 1995]. While not our main result, this result shows that the problem of *offline* A-B testing (which is still valuable in some traditional applications) can surprisingly be solved efficiently.

3. In the online setting — which is the algorithmic focal point of our work — our optimization problem is, not surprisingly, a high dimensional dynamic optimization problem with dimension that grows like the number of covariates. *We show how to break the curse of dimensionality here.* In particular, we show that the state space of this dynamic optimization

problem collapses if covariates come from an elliptical family of distributions (a family that includes, for example, the multivariate Gaussian). This yields an *efficient* algorithm that is provably optimal in the elliptical distribution setting and that can nonetheless be employed when covariates are not from an elliptical family. Using real data on user impressions on Yahoo.com, we show that our algorithm does indeed yield impressive gains in efficiency even when covariates do not arise from an elliptical family.

Thus, our main contribution is providing an algorithm for the challenging problem of online A-B testing that can be shown to be near-optimal when covariates are drawn from an elliptical family. The algorithm is applicable to a canonical family of treatment models. Given the vast extant literature on this problem, it is a pleasant surprise that such an algorithm exists.

4.1.2 Related Literature

The theory of optimal experiment design (which, in a sense, subsumes the problems we consider here) starts with the seminal work of [Fisher, 1935]. Important textbook expositions of this mature topic include that of [Pukelsheim, 2006] and [Cook *et al.*, 1979], the latter of which discusses the notion of covariate matching as it applies to practice. While not our primary focus, the 'offline' problem we discuss in this paper is of practical relevance in the social sciences; see [Raudenbush *et al.*, 2007], for an application and heuristics. [Kallus, 2013] studies an approach to this problem based on linear mixed integer optimization with an application to clinical trials. Finally, [Kallus, 2012] presents a robust optimization framework for the offline problem with an emphasis on allocations of treatments that are robust to the specific form of the subjects response as a function of the treatments and subject covariates (we merely consider linear functions here). For the linear response functions we consider here, our offline problem may be viewed as a special case of the problem of D_a optimal experiment design for which, unlike the general case, we can present an efficient algorithm.

The problem that is of greatest algorithmic interest to us is the 'online' allocation problem, where treatments must be assigned to subjects as they arrive. With regard to this problem, [Efron, 1971] proposed a seminal strategy that sought to 'balance' the number of subjects in each trial while minimizing certain types of selection bias. This approach was generalized to problems where a number of known covariates for each subject were available by a number of authors, of which [Pocock and Simon, 1975] present a widely cited and used heuristic. [Kapelner and Krieger, 2013] provide

a sophisticated and recently proposed heuristic intended for the same setting. [Atkinson, 1982; Atkinson, 1999] made the first attempt to root the design of dynamic allocation methods in theory by connecting them with the notion of D_a optimality in experiment design. Atkinson, however, made no attempt to formulate a dynamic optimization problem and the heuristics he proposed seem difficult to justify from a dynamic optimization perspective. [Smith, 1984a; Smith, 1984b] considers a similar dynamic optimization formulation, which is also addressed via heuristics. Finally, [Rosenberger and Sverdlov, 2008] provides an excellent overview of the variety of heuristics that exist for the dynamic allocation problem. The model we consider is narrower than that required for some of the work above in its requirement that the dependence on covariates be linear; this distinction is not substantive as our work permits easy extensions to more complicated dependencies. A second distinction is that in order to formulate a rigorous optimization problem we focus on a single objective — the variance in our estimate of the treatment effect — whereas some of the aforementioned literature considers issues such as manipulation (selection bias) by experiment designers, etc.

Related but Distinct Problems.

It is important to distinguish the experiment design problems considered here from 'bandit' problems, particularly those with side information [Woodroofe, 1979] as both classes of problems frequently find application in very related applications. Here we simply note that the experiment design setting is appropriate when an irrevocable decision of what treatment is appropriate must be made (e.g., use creative A as opposed to B for a display campaign), whereas the bandit setting is appropriate in a setting where the decision can be changed over time to optimize the (say) long-run average value of some objective (e.g., maximizing revenues by finding the best audience for a specific campaign). From a methodology perspective, an important difference is that solution methods for bandit problems need to address an 'exploitation-exploration' trade-off between learning the best alternative and collecting rewards to optimize the objective, while there is no such trade-off in our experiment design setting.

Another closely related class of problems are ranking and selection problems where the task is to pick the best of a set of alternatives with a budget on samples (for an overview see [Kim and Nelson, 2006]). In our lexicon, the emphasis in such problems is choosing from multiple (typically, greater than two) treatments in the *absence* of observable covariates on a sample. Interestingly,

recent progress on this class of problems has also heavily employed dynamic optimization techniques [Chick and Gans, 2009] [Chick and Frazier, 2012].

As a final note, the emphasis in our work is on A-B testing with a fixed budget on samples. It is interesting to consider A-B tests that can be 'stopped' with continuous monitoring. Doing so can introduce a significant bias towards false discovery; [Pekelis *et al.*, 2015] have recently made exciting progress on this problem.

4.2 Model

In this section we describe the model in more detail. Given the model assumptions in Section 4.2.1, our problem is to find an estimator of the treatment effect that minimizes the variance of the estimate. In Section 4.2.2 we pose the two optimization problem that are of interest. One of them is the offline problem where all subjects can be observed before making the decisions and the other is the sequential problem where subjects must be allocated without knowing the future arrivals.

In Section 4.2.3 we give an interpretation to our objective and explain why the allocations prescribed by the optimization problems make intuitive sense. Finally, in Section 4.2.4 we give a simple upper bound to both the optimization problems. This shows that the efficiency (inverse variance) of the optimal allocation is $O(n)$.

4.2.1 Setup

We must learn the efficacy of a treatment by observing its effect on n subjects. The kth subject is assigned a treatment $x_k \in \{\pm 1\}$. The kth subject is associated with a covariate vector (i.e., side information or context) $Z_k \in \mathbb{R}^p$. We assume that impact of the treatment on the kth subject is given by:

$$y_k = x_k\theta + Z_k^\top \kappa + \epsilon_k.$$

This assumes a linear dependence of the covariates and treatment decision on the outcome. The treatment effect $\theta \in \mathbb{R}$ and the weights on the covariates $\kappa \in \mathbb{R}^p$ are unknown. Our aim is to estimate θ. The $\{\epsilon_k\}$ are i.i.d. zero mean random variables with variance σ^2. The key restriction imposed by this model is that the impact of treatment is additive, an assumption that is ubiquitous in all of the related literature on the topic. Further, we assume that there is no endogeneity. In

other words, there are no unobserved covariates.

Letting $Z \in \mathbb{R}^{n\times p}$ be the matrix whose kth row is $Z_k^\top$, throughout this paper, we will assume that:

Assumption 3. *The first column of Z is a vector of all ones. Further, Z is full rank and $p \leq n-1$.*

The requirement that one of the covariates be a constant ensures that θ is interpreted as a treatment effect, otherwise it could be learned from the assignment of a single treatment. The crucial assumption is that $p \leq n-1$, which nonetheless allows for a large number of covariates. In fact the scenario where $p \sim n$ is particularly relevant. For a particular allocation of treatments, x, let us denote by $\hat{\theta}_x$ the least squares estimator for θ.

4.2.2 Optimization Problem

We are interested in finding an estimator with minimal variance or, equivalently, maximal efficiency. A standard calculation yields that the estimator $\hat{\theta}_x$ has efficiency

$$\mathrm{Eff}(\hat{\theta}_x) \triangleq \frac{1}{\mathrm{Var}(\hat{\theta}_x)} = \frac{x^\top P_{Z^\perp} x}{\sigma^2}, \tag{4.1}$$

where $P_{Z^\perp} \triangleq I - Z(Z^\top Z)^{-1}Z^\top$. Details are presented in Section C.1 of the appendix.

We can now immediately state the *offline experiment design problem*:

$$\begin{aligned} \text{(P1)} \triangleq \text{maximize} \quad & x^\top P_{Z^\perp} x \\ \text{subject to} \quad & x \in \{\pm 1\}^n. \end{aligned}$$

Here, given the collection of covariates Z, we seek to find the allocation x which yields the least squares estimator with maximal efficiency.

In many real world applications the assignments need to be made in a sequential fashion. Subjects arrive one at a time and the assignment must be made without the knowledge of subjects in the future. We formulate this as a dynamic optimization problem. To this end we must now assume the existence of a measure on the covariate process $\{Z_k\}$. We define a filtration $\{\mathcal{F}_k\}$ by setting, for each time k, $\mathcal{F}_k$ to be the sigma algebra generated by the first k covariates $(Z_1, \ldots, Z_k)$ and the first $k-1$ allocations $(x_1, \ldots, x_{k-1})$. The *online experiment design problem* is then given

by:

$$\begin{aligned} \text{(P2)} \triangleq \text{maximize} \quad & \mathsf{E}\left[x^\top P_{Z^\perp} x\right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\, 1 \le k \le n, \end{aligned}$$

where the expectation is over the distribution of the covariate process.

4.2.3 Problem Interpretation

Before moving on to algorithm design, we pause to interpret the offline and online problems present above. First we begin with an intuitive interpretation of the objective. Define the imbalance in covariate effects between the test and control groups, $\overline{\Delta}_n \in \mathbb{R}^p$, according to $\overline{\Delta}_n \triangleq \sum_{k=1}^n x_k Z_k = Z^\top x$. Notice that the empirical second moment matrix for the covariates is given by $\Gamma_n \triangleq Z^\top Z/n$. Then, it is easy to see that the objective of the offline problem (P1) reduces to

$$x^\top P_{Z^\perp} x = x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) x = n\left(1 - \overline{\Delta}_n^\top \Gamma_n^{-1} \overline{\Delta}_n\right).$$

Therefore, the offline problem (P1) is equivalent to minimizing the square of the weighted euclidean norm of $\overline{\Delta}$,

$$\left\|\overline{\Delta}_n\right\|_{\Gamma_n^{-1}}^2 \triangleq \overline{\Delta}_n^\top \Gamma_n^{-1} \overline{\Delta}_n,$$

while (P2) seeks to minimize the expected value of this quantity where the expectation is over the covariate process and our allocations. Put simply, both problems seek to minimize the aggregate imbalance of covariates between the treatment and control groups, measured according to this norm.

We next seek a statistical interpretation of the two problems. To this end, we note the Cramér-Rao bound dictates that, provided x and Z are independent of ϵ, and further if ϵ is normally distributed, then for *any* unbiased estimator of the treatment effect $\tilde{\theta}_x$, we have that

$$\mathrm{Eff}(\tilde{\theta}_x) \le \mathrm{Eff}(\hat{\theta}_x)$$

where the right hand side quantity is the efficient of the least square estimator. Now both problems (P1) and (P2) seek to find an allocation x to maximize the latter quantity, or its expected value, respectively. Consequently, both problems may be interpreted as seeking an allocation of samples to the test and control group with a view to maximizing the efficiency of our estimate of the treatment effect among *all* unbiased estimators of the treatment effect.

4.2.4 Upper Bound on Efficiency

We end this section with an upper bound on the efficiency of any unbiased estimator that is a straightforward consequence of the Cramér-Rao bound:

Proposition 6. *If $\epsilon \sim N(0, \sigma^2 I)$, then for any covariate matrix Z and any unbiased estimator $(\hat{\theta}, \hat{\kappa})$, including non-least squares estimators, we have:*

$$\text{Eff}(\hat{\theta}) \leq \frac{n}{\sigma^2},$$

an upper bound on the optimal value of both problems (P1) and (P2). For non-Gaussian noise ϵ, this upper bound still holds for all least squares estimators.

This proposition shows that the efficiency of the optimal estimator is $O(n)$. Consider the case when subjects are identical, i.e., $p = 1$ and $Z_k = 1$ for all k. It is easy to note that, in this case assuming n is even, the optimal design allocates half of the subjects to either treatments. Further, the efficiency of such a design is σ^2/n, the optimal achievable efficiency. For $p > 1$ this efficiency is less than this value. Thus the presence of covariates only makes the inference challenging.

Proof of Proposition 6. By Cramér-Rao we have that,

$$\text{Cov}\left(\begin{bmatrix}\hat{\theta}\\ \hat{\kappa}\end{bmatrix}\right) \succeq I(\theta, \kappa)^{-1},$$

where $I(\theta, \kappa)$ is the Fisher information matrix. Under the Gaussian assumption for ϵ it is easy to see that,

$$I(\theta, \kappa)^{-1} = \sigma^2 \begin{bmatrix} x^\top x & x^\top Z \\ Z^\top x & Z^\top Z \end{bmatrix}^{-1}.$$

If e_1 is the unit vector along the first coordinate then,

$$\text{Var}(\hat{\theta}) \geq e_1^\top I(\theta, \kappa)^{-1} e_1 = \frac{\sigma^2}{x^\top (I - Z(Z^\top Z)^{-1} Z^\top) x}.$$

Thus,

$$\text{Eff}(\hat{\theta}) \leq \frac{x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) x}{\sigma^2} = \frac{n - x^\top Z(Z^\top Z)^{-1} Z^\top x}{\sigma^2} \leq \frac{n}{\sigma^2}.$$

The inequality follows since $Z(Z^\top Z)^{-1} Z^\top$ is positive semidefinite.

The last statement is consequence of the fact that $x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) x/\sigma^2$ is the efficiency of the optimal least squares estimator (see Section C.1 of the appendix). ■

4.2.5 Hypothesis Tests

A common statistical practice given $\hat{\theta}$ is to do perform a hypothesis test on whether $\theta = 0$ vs $\theta \neq 0$ ($\theta > 0$ or $\theta < 0$ for one sided tests.) This way the practitioner can decide whether or not the there is a treatment effect.

Given the model assumptions it is fairly straghtforward to perform a t-test similar routinely performed to check whether a coeffient in a linear regression is significant.

There is a broader class of hypothesis tests that have a somewhat lesser dependence on the model. Fisher's permutation test and Wilcox's rank sum test are some examples of these 'non-parametric' hypothesis tests. These tests rely heavily on the assumption that the allocations to treatment and control are done in a randomized fashion. We don't tackle the issue of perfmorming non-parametric hypothesis tests with 'optimal' allocations and leave it to further research.

4.3 Offline Problem

In this section, we consider the offline optimization problem (P1). We show that this combinatorial problem permits a tractable, constant factor approximation using an SDP-based randomized rounding algorithm. Moreover, in this setting, we can analyze the effect optimization has on the efficiency of the estimator of the treatment effect, as compared to randomization. To this end, we first obtain the mean efficiency of the randomized design. Surprisingly, efficiency is a simple function of n and p and does not depend on the data matrix Z. We show that when $p \sim n$, the randomization is rather inefficient and the efficiency is $O(1)$. This can be contrasted with the upper bound on efficiency given by Proposition 6 which is $\Omega(n)$. To conclude the section, we analyze the performance of the optimal allocation assuming a distribution on Z. We show that for any p, the efficiency of optimal allocation is $\Omega(n)$. Thus concluding that when $p \sim n$, randomization can be arbitrarily bad as compared to the optimal design.

4.3.1 Approximation Algorithm for (P1)

First, we observe that there is a tractable approximation algorithm to solve the combinatorial optimization problem (P1). In particular, consider the semidefinite program (SDP) over symmetric

positive semidefinite matrices $Y \in \mathbb{R}^{n \times n}$ given by

$$\begin{aligned} \text{(P1-SDP)} \triangleq \quad & \text{maximize} && \operatorname{tr}(P_{Z^\perp} Y) \\ & \text{subject to} && Y_{kk} = 1, \qquad \forall\ 1 \le k \le n, \\ & && Y \succeq 0, \\ & && Y \in \mathbb{R}^{n \times n}. \end{aligned}$$

It is straight forward to see that (P1-SDP) is a relaxation of (P1): given $x \in \{\pm 1\}^n$ feasible for (P1), define the symmetric positive definite matrix $Y \triangleq xx^\top \in \mathbb{R}^{n \times n}$. Then, clearly Y is feasible for (P1-SDP), and the objective values for (P1) and (P1-SDP) coincide. Moreover, because it is an SDP, (P1-SDP) can be efficiently solved in polynomial time.

Building upon prior work on the MAX-CUT problem [Goemans and Williamson, 1995], the following result, due to [Nesterov, 1997], establishes that (P1-SDP) can be used as the basis of a randomized algorithm to solve (P1) with a constant factor guarantee with respect to the optimal design:

Theorem 6. *Let Y be an optimal solution to (P1-SDP), with a matrix factorization $Y = V^\top V$, for some matrix $V \in \mathbb{R}^{n \times n}$ with columns $v_1, \ldots, v_n \in \mathbb{R}^n$. Let $u \in \mathbb{R}^n$ be a vector chosen at random uniformly over the unit sphere, and define the random vector $\tilde{x} \in \{\pm 1\}^n$ by setting*

$$\tilde{x}_k \triangleq \begin{cases} +1 & \text{if } u^\top v_k \ge 0, \\ -1 & \text{if } u^\top v_k < 0, \end{cases}$$

for each $1 \le k \le n$. Then,

$$\mathsf{E}_u \left[\tilde{x}^\top P_{Z^\perp} \tilde{x} \right] \ge \frac{2}{\pi} \max_{x \in \{\pm 1\}^n} x^\top P_{Z^\perp} x,$$

where the expectation is taken over the choice of random vector u. In order words, the expected value achieved by the vector $\tilde{x}$ in the offline experiment design problem (P1) is within a constant factor $2/\pi$ of the best possible.

Proof. This theorem is a direct consequence of Theorem 3.4.2 of [Ben-Tal and Nemirovski, 2001]. That result states that any quadratic integer optimization problem with objective $x^\top Q x$, such that $x \in \{\pm 1\}^n$, can be approximated within a relative error of $\pi/2$ using the prescribed algorithm, provided Q is positive semidefinite. Since $P_{Z^\perp}$ is positive semidefinite (indeed, it is a projection matrix), the result follows. ■

4.3.2 Optimal Allocations vs. Randomized Allocations

Randomization is the most popular technique used for A-B testing. In what follows, we will compare the performance of randomization to what can be achieved by the optimal offline allocation of (P1).

In its most basic variation, simple randomization partitions the population into two equally sized groups, each assigned a different treatment, where the partition is chosen uniformly at random over all such partitions (for simplicity, we will assume that the population is of even size). Denote by $X_{\text{rand}} \in \{\pm 1\}^n$ the random allocation generated by simple randomization, and denote by $\hat{\theta}_{X_{\text{rand}}}$ the resulting unbiased least squares estimator for θ.

Theorem 7. *If n is even, given a covariate matrix Z, define the expected efficiency of simple randomization*

$$\text{Eff}_{\text{rand}} \triangleq \mathsf{E}_{X_{\text{rand}}}\left[\text{Var}\left(\hat{\theta}_{X_{\text{rand}}}\right)^{-1}\right],$$

where the expectation is taken over the random allocation X_{rand}. Then,

$$\text{Eff}_{\text{rand}} = \frac{n}{\sigma^2}\left(1 - \frac{p-1}{n-1}\right).$$

The proof relies on simple probabilistic arguments and is presented in Section C.2 of the appendix. Surprisingly the efficiency of the randomized allocation *does not* depend on the data matrix Z at all, as long as it is full rank and has a constant column.

Comparing with the upper bound of Proposition 6, we notice that in the large sample size regime where $n \to \infty$, simple randomization is asymptotically order optimal in the sense that it achieves efficiency that grows with order n — the maximum permitted by the upper bound of Proposition 6 — when $p \ll n$. This may not be the case when p is close to n, however. For example, if $p = n - 1$, which is the maximum value p can take under Assumption 3, then $\text{Eff}_{\text{rand}} \approx 1/\sigma^2$, which is of *constant order*. In such a case, the least squares estimator $\hat{\theta}_{X_{\text{rand}}}$ will not asymptotically converge to θ as $n \to \infty$.

Now we consider the performance of the optimal estimator that would be obtained by solving the offline experiment design problem (P1). By construction, the optimal estimator will clearly have efficiency that is at least that of the randomized procedure. We would like to understand the magnitude of the possible improvement, however, and to see if it is material. Unlike in the simple randomized case, however, the efficiency of the optimal estimator depends on the covariate matrix Z. Moreover, it is difficult to obtain a closed-form expression for this efficiency as a function of Z.

We can illustrate this with a simple example. Consider the case where $p = n-1$. The efficiency of the optimal estimator is given by

$$\sup_{x \in \{\pm 1\}^n} \frac{x^\top P_{Z^\perp} x}{\sigma^2}.$$

Since $p = n-1$, the null space of $Z^\top$ is a one dimensional subspace of $\mathbb{R}^n$. Let $y \in \mathbb{R}^n$ be a non-zero vector such that $Z^\top y = 0$ and $\|y\|_2^2 = 1$. That is, y is a unit vector in the null space of $Z^\top$. It is easy to see that $P_{Z^\perp} = yy^\top$. Thus, the efficiency of the optimal estimator is

$$\sup_{x \in \{\pm 1\}^n} \frac{x^\top y y^\top x}{\sigma^2} = \sup_{x \in \{\pm 1\}^n} \frac{\left(y^\top x\right)^2}{\sigma^2} = \frac{\|y\|_1^2}{\sigma^2}. \tag{4.2}$$

Now, consider the following two cases:

1. y has only two non-zero components given by $1/\sqrt{2}$ and $-1/\sqrt{2}$. In this case, the optimal efficiency is $2/\sigma^2$. Thus, in this case, randomization is within a constant factor of optimal.

2. y has entries such that $|y_i| = 1/\sqrt{n}$ and $\mathbf{1}^\top y = 0$. In this case, the efficiency is n/σ^2. Thus, in this case, the optimal design achieves the Cramér-Rao upper bound and the performance is a significant improvement over the randomized design.

The preceding two cases show, that depending on the covariate matrix Z (which determines the vector y in the discussion above), the performance of the optimal design may be a drastic improvement over that of the randomized design. In order to study the performance of the optimal design, we proceed by making a certain probabilistic assumption on Z. Under this assumption, we will then analyze the distribution of performance of the optimal design. For this purpose, we will assume a distribution on the covariate matrix Z as follows:

Assumption 4. *Given (n, p) with $1 \leq p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n \times p}$ has independent and identically distributed rows. Further, assume that for each $1 \leq k \leq n$, the kth row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector of all components except the first satisfies $Z_{k,2:p} \sim N(0, \Sigma)$, i.e., it is distributed according to a multivariate normal distribution with zero mean and covariance matrix $\Sigma \in \mathbb{R}^{p-1 \times p-1}$.*

Is it easy to check that, under Assumption 4, the covariate matrix Z will satisfy the full rank condition of Assumption 3 almost surely. Consider a sequence of problems indexed by the sample

size n, and where the dimension of the covariates is given by $1 \leq p_n < n$. For each n, let $Z^{n,p_n} \in \mathbb{R}^{n \times p_n}$ be the data matrix satisfying Assumption 4. We have that:

Theorem 8. *Suppose that Assumption 4 holds with $\Sigma = \rho^2 I$. Let x^* be an optimal design obtained by solving (P1) with covariance matrix $Z = Z^{n,p_n}$, and let $\hat{\theta}_{x^*,Z^{n,p_n}}$ be the corresponding least squares estimator of θ. Define the efficiency of this estimator by*

$$\mathrm{Eff}_*^{n,p_n} \triangleq \mathrm{Var}\left(\hat{\theta}_{x^*,Z^{n,p_n}}\right)^{-1}.$$

Then, we have that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathsf{P}\left(\frac{\mathrm{Eff}_*^{n,p_n}}{n} < \frac{1}{8\pi\sigma^2} - \epsilon\right) = 0,$$

where the probability is taken over the distribution of the covariance matrix.

Theorem 8 states that, with high probability, the optimal offline optimization-based design always yields $\Omega(n)$ efficiently under Assumption 4. Comparing to Theorem 7, the optimal efficiency is a $\Theta(n)$ relative improvement over that of simple randomization when p is large (i.e., $p = \Omega(n)$). In other words, if the number of covariates is comparable to the number of samples, we might expect dramatic improvements over simple randomization through optimization. Moreover, while the optimal design requires solution of (P1), which may not be tractable, Theorem 6 suggests a tractable approximation which is guaranteed to achieve the same efficiency as the optimal design up to a constant factor.

The proof of Theorem 8 is presented in Section C.3. Here we provide a proof sketch. Let $Z^{n,p} \in \mathbb{R}^{n \times p}$ and $Z^{n,n-1} \in \mathbb{R}^{n \times p}$ be two covariate matrices defined on the same probability space (under the Assumption 4 with $\Sigma = \rho^2 I$) such that they are identical on the first p columns. We show that $\mathrm{Eff}_*^{n,p} \geq \mathrm{Eff}_*^{n,n-1}$. This establishes that $p = n - 1$ corresponds to the worst case efficiency and allows us to focus on the sequence $\mathrm{Eff}_*^{n,n-1}$. We then analyze the distribution of $Z^{n,n-1}$. We show that $\mathrm{Eff}_*^{n,n-1}$ can be written down as a function of a unit vector in the null space of $(Z^{n,n-1})^\top$, say $y_n \in \mathbb{R}^n$. Further, y_n describes a random one-dimensional subspace of $\mathbb{R}^n$ that is invariant to orthonormal transformations that leave the constant vector unchanged. There is a unique distribution that has this property. We then identify the distribution and compute the efficiency in the case in closed-form using this distribution. In particular, we show that,

$$\frac{\mathrm{Eff}_*^{n,n-1}}{n} \to \frac{1}{8\pi\sigma^2},$$

where the convergence is in distribution.

4.4 Sequential Problem

We now consider the online experiment design problem (P2). Here, decisions must be made sequentially. At each time k, an allocation $x_k \in \{\pm 1\}$ must be made based only on the first k covariates and any prior allocations. In other words, x_k is $\mathcal{F}_k$-measurable.

In this section we show that the optimization problem is tractable. First, we pose a surrogate problem in which the objective of (P2) is simplified. The details of this simplification are provided in Section 4.4.1. In Section 4.4.2, we show that loss in performance when the surrogate problem is used to device an assignment policy is negligible. Focusing on the surrogate problem, we show that the surrogate problem is a p-dimensional dynamic program in Section 4.4.3. Surprisingly, if we assume that the data generating distribution for the covariates comes from the so-called *elliptical family* then the state space collapses to two dimensions, making the dynamic program tractable. This state space collapse is presented in Section 4.4.4.

4.4.1 Formulation and Surrogate Problem

In order to formulate the sequential problem with an expected value objective, a probabilistic model for covariates is necessary. We will start by making the following assumption:

Assumption 5. *Given (n, p) with $1 \leq p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n \times p}$ has independent and identically distributed rows. Further, assume that for each $1 \leq k \leq n$, the kth row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector $Z_{k,2:p} \in \mathbb{R}^{p-1}$ of all components except the first has zero mean and covariance matrix $\Sigma \in \mathbb{R}^{p-1 \times p-1}$.*

Assumption 5 requires that the sequentially arriving covariates are i.i.d. with first and second moments. Assumption 4, by comparison, in addition imposes a Gaussian distribution.

Problem (P2) can be viewed as maximizing the expectation of terminal reward that is given by

$$x^\top P_{Z^\perp} x = x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top \right) x = n - \frac{1}{n} \left(\sum_{k=1}^{n} x_k Z_k \right)^\top \Gamma_n^{-1} \left(\sum_{k=1}^{n} x_k Z_k \right), \tag{4.3}$$

where

$$\Gamma_n \triangleq \frac{1}{n} \sum_{k=1}^{n} Z_k Z_k^\top.$$

We write this matrix in block form as

$$\Gamma_n = \begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix},$$

where,

$$\Sigma_n \triangleq \frac{1}{n}\sum_{k=1}^{n} Z_{k,2:p} Z_{k,2:p}^\top, \quad M_n \triangleq \frac{1}{n}\sum_{k=1}^{n} Z_{k,2:p}.$$

Here, M_n and Σ_n correspond to sample estimates of the covariate mean and covariance structure, respectively.

We define, for each k, the scalar $\delta_k \in \mathbb{R}$ and the vector $\Delta_k \in \mathbb{R}^{p-1}$ by

$$\delta_k \triangleq \sum_{\ell=1}^{k} x_\ell, \quad \Delta_k \triangleq \sum_{\ell=1}^{k} x_\ell Z_{\ell,2:p}.$$

The terminal reward (4.3) is equal to

$$x^\top P_{Z^\perp} x = n - \frac{1}{n}\begin{bmatrix}\delta_n & \Delta_n^\top\end{bmatrix}\begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix}^{-1}\begin{bmatrix}\delta_n \\ \Delta_n\end{bmatrix}.$$

Problem (P2) is then equivalent to

$$\begin{aligned} \text{(P3)} \triangleq \text{minimize} \quad & \mathsf{E}\left[\begin{bmatrix}\delta_n & \Delta_n^\top\end{bmatrix}\begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix}^{-1}\begin{bmatrix}\delta_n \\ \Delta_n\end{bmatrix}\right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\ 1 \le k \le n. \end{aligned}$$

Observe that, as $n \to \infty$, by the strong law of large numbers (under mild additional technical assumptions), $\Sigma_n \to \Sigma$ and $M_n \to 0$ almost surely. Motivated by this fact, in developing an efficient algorithm for (P3), our first move will be to consider a surrogate problem that replaces the sample covariance matrix Σ_n with the exact covariance matrix Σ and sets the sample mean M_n to the exact mean 0:

$$\begin{aligned} \text{(P3}') \triangleq \text{minimize} \quad & \mathsf{E}\left[\delta_n^2 + \|\Delta_n\|_{\Sigma^{-1}}^2\right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\ 1 \le k \le n. \end{aligned}$$

Here, given an arbitrary covariance matrix $\hat{\Sigma} \in \mathbb{R}^{p-1\times p-1}$, we find it convenient to introduce the norm $\|\cdot\|_{\hat{\Sigma}^{-1}}$ on $\mathbb{R}^{p-1}$ defined by $\|z\|_{\hat{\Sigma}^{-1}} \triangleq (z^\top \hat{\Sigma}^{-1} z)^{1/2}$. In the present context, this norm is typically referred to as a Mahalanobis distance.

The roles of δ_n and Δ_n in the surrogate problem (P3′) are intuitive: requiring δ_n to be small balances the number of assignments between the two treatments (the focus of the so-called biased-coin designs). Requiring the same of Δ_n will tend to 'balance' covariates — when Δ_n is small, the empirical moments of the covariates across the two treatments are close. As discussed in the introduction, heuristics developed in the literature on the design of optimal trials tend to be driven by precisely these two forces.

For the rest of this section we will focus on the surrogate problem. We want to first justify the use of the surrogate objective. We do this by providing an approximation guarantee in Section 4.4.2. We then turn our attention on how to solve the surrogate problem via dynamic programming in the subsequent sections.

4.4.2 Approximation Guarantee for the Surrogate Problem

First, we show that the policy obtained by solving (P3′) is near optimal. Denote by $\hat{\mu}$ the measure over the sequence x_k induced by an optimal solution for the surrogate control problem (P3′), and let μ^* denote the measure induced by an optimal policy for our original dynamic optimization problem (P3). Now, δ_n and Δ_n are random variables given an allocation policy. Given a allocation policy μ, define

$$D_{\mu}^{n,p} \triangleq \mathsf{E}_{\mu}\left[\begin{bmatrix}\delta_n & \Delta_n^{\top}\end{bmatrix}\Gamma_n^{-1}\begin{bmatrix}\delta_n \\ \Delta_n\end{bmatrix}\right]$$

to be the objective value of (P3) under the allocation policy μ with sample size n and covariate dimension p. The following result is demonstrated, without loss of generality, under the assumption that Σ is the identity (otherwise, we simply consider setting $Z_{k,2:p}$ to $\Sigma^{-1/2}Z_{k,2:p}$):

Theorem 9. *Suppose that Assumption 4 holds with $\Sigma = I$ and let $\epsilon > 0$ be any positive real number. Consider a sequence of problems indexed by the sample size n, where the dimension of the covariates is given by $1 \leq p_n < n$ and $\rho_n > 0$ are real numbers such that, for n sufficiently large, $n \geq L\max(p_n, l\log 2/\rho_n)/\epsilon^2$. Then, as $n \to \infty$*

$$D_{\hat{\mu}}^{n,p_n} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 D_{\mu^*}^{n,p_n} + \rho_n n^2 + \rho_n n^2 p_n + O\left(\sqrt{\frac{n}{p_n - 1}}\right).$$

Here, L and l are universal constants. In particular, selecting $\rho_n \propto 1/n^4$ yields

$$D_{\hat{\mu}}^{n,p_n} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 D_{\mu^*}^{n,p_n} + O\left(\sqrt{\frac{n}{p_n - 1}}\right). \tag{4.4}$$

The result above relies on the use of non-asymptotic guarantees on the spectra of random matrices with sub-Gaussian entries and can be found in Section C.5 of the appendix.

The preceding result bounds the objective of the problem (P3) when (P3′) is used to devise an allocation policy. However, we are interested in the objective of the problem problem (P2), which is the efficiency or inverse variance of the design corresponding to the policy used. In particular, denote by $\text{Eff}_{\mu}^{n,p}$ the expected efficiency of the estimator when allocations are made with a policy μ, for a problem with sample size n and covariate dimension p, i.e.,

$$\text{Eff}_{\mu}^{n,p_n} = \frac{\mathsf{E}_{\mu}\left[x^{\top} P_{Z^{\perp}} x\right]}{\sigma^2} = \frac{n - D_{\mu}^{n,p_n}/n}{\sigma^2}. \tag{4.5}$$

Then, we have the following:

Corollary 2. *Suppose that Assumption 4 holds with $\Sigma = I$. Consider a sequence of problems indexed by the sample size n, where the dimension of the covariates is given by $1 \leq p_n < n$, and a fixed positive real number $\epsilon > 0$ such that*

$$\epsilon > \sqrt{L \limsup_{n\to\infty} p_n/n},$$

for a universal constant L. Then, as $n \to \infty$,

$$\frac{\text{Eff}_{\hat{\mu}}^{n,p_n}}{\text{Eff}_{\mu^*}^{n,p_n}} \geq 1 - \frac{4\epsilon^3}{(L-\epsilon^2)(1-\epsilon^2)} + o(1).$$

Corollary 2 gives the multiplicative loss in the efficiency by using an allocation derived from the surrogate problem (P3′). The multiplicative loss depends on the ratio p/n, which is captured in the choice of ϵ. For small values of ϵ the ratio of efficiency obtained by solving (P3′) and (P2) approaches 1. Note that this result holds in an asymptotic regime where p and n both increase to infinity, as long as p/n remains small.

Proof of Corollary 2. Consider (4.4) in Theorem 9. This holds when

$$n \geq \frac{L \max(p_n, l \log 2/\rho_n)}{\epsilon^2}$$

with $\rho_n = b/n^4$ for some constant b. Equivalently,

$$n \geq \frac{L \max(p_n, 4l \log n + 2l \log b)}{\epsilon^2}.$$

For n sufficiently large, clearly the constraint that $n \geq L(4l \log n + 2l \log b)/\epsilon^2$ will be satisfied. Therefore, combined with the lower bound hypothesized for ϵ, (4.4) holds as $n \to \infty$.

Using (4.5),

$$\begin{aligned} \mathrm{Eff}_{\mu^*}^{n,p_n} - \mathrm{Eff}_{\hat{\mu}}^{n,p_n} &= \frac{D_{\hat{\mu}}^{n,p_n} - D_{\mu^*}^{n,p_n}}{n\sigma^2} \\ &\leq \frac{\dfrac{(1+\epsilon)^2}{(1-\epsilon)^2} D_{\mu^*}^{n,p_n} - D_{\mu^*} + O\left(\sqrt{\dfrac{n}{p_n - 1}}\right)}{n\sigma^2} \\ &= \frac{4\epsilon D_{\mu^*}^{n,p_n}}{n\sigma^2(1-\epsilon)^2} + o(1) \\ &= \frac{4\epsilon}{(1-\epsilon)^2}\left(\frac{n}{\sigma^2} - \mathrm{Eff}_{\mu^*}^{n,p_n}\right) + o(1). \end{aligned} \tag{4.6}$$

The first inequality follows from Theorem 9 and the last equality from (4.5).

Let $\mathrm{Eff}_{\mathrm{rand}}^{n,p_n}$ denote efficiency of the randomized policy. Using Theorem 7 and the optimality of μ^*, we have that

$$\frac{n}{\sigma^2} - \mathrm{Eff}_{\mu^*}^{n,p_n} \leq \frac{n}{\sigma^2} - \mathrm{Eff}_{\mathrm{rand}}^{n,p_n} = \frac{n}{\sigma^2}\frac{p_n - 1}{n-1} \leq \frac{n}{\sigma^2}\frac{p_n}{n} \leq \frac{\epsilon^2 n}{L\sigma^2}, \tag{4.7}$$

where the last inequality uses the fact that, by hypothesis, $p_n/n \leq \epsilon^2/L$. Substituting this into (4.6) we get that

$$\mathrm{Eff}_{\mu^*}^{n,p_n} - \mathrm{Eff}_{\hat{\mu}}^{n,p_n} \leq \frac{4\epsilon^3 n}{(1-\epsilon)^2 L\sigma^2} + o(1).$$

Now, using (4.7) we get that,

$$\mathrm{Eff}_{\mu^*}^{n,p_n} \geq \frac{n}{\sigma^2}\left(1 - \frac{\epsilon^2}{L}\right).$$

Thus, we have that,

$$\begin{aligned} 1 - \frac{\mathrm{Eff}_{\hat{\mu}}^{n,p_n}}{\mathrm{Eff}_{\mu^*}^{n,p_n}} &\leq \frac{4\epsilon n}{\mathrm{Eff}_{\mu^*}^{n,p_n}(1-\epsilon)^2 L\sigma^2} + o(1) \\ &\leq \frac{4\epsilon^3}{(L-\epsilon^2)(1-\epsilon^2)} + o(1). \end{aligned}$$

This yields the result. ■

4.4.3 Dynamic Programming Decomposition

It is not difficult to see that (P3′) is a terminal cost dynamic program with state $(\delta_{k-1}, \Delta_{k-1}) \in \mathbb{R}^p$ at each time k. The pair (δ_k, Δ_k) can be interpreted as the state of the dynamic decision problem.

In other words, given the past arrival sequence and actions, (δ_k, Δ_k) summarizes the the impact of this 'past' on the future objective. This is formally stated in the following proposition:

Proposition 7. *Suppose that Assumption 5 holds. For each $1 \leq k \leq n$, define the function $Q^k : \mathbb{R} \times \mathbb{R}^{p-1} \to \mathbb{R}$ by the Bellman equation*

$$Q^k(\delta_k, \Delta_k) \triangleq \begin{cases} \delta_n^2 + \|\Delta_n\|^2_{\Sigma^{-1}}, & \text{if } k = n, \\ \mathsf{E}\left[\min_{u \in \{\pm 1\}} \; Q^{k+1}(\delta_k + u, \Delta_k + Z_{k+1,2:p})\right], & \text{if } 1 \leq k < n. \end{cases} \tag{4.8}$$

Then,

1. *At each time k, the optimal continuation cost for the dynamic program (P3′) is given by $Q^k(\delta_k, \Delta_k)$. In other words, this is the expected terminal cost, given then covariates observed and the allocations made up to and including time k, assuming optimal decisions are made at all future times.*

2. *Suppose the allocation x_k^* at each time k is made according to*

$$x_k^* \in \operatorname*{argmin}_{u \in \{\pm 1\}} \; Q^k\left(\delta_{k-1} + u, \Delta_{k-1} + uZ_k\right).$$

   Then, the sequence of allocations x^* is optimal for the online experiment design problem (P3′).

Proposition 7, whose proof is presented in Section C.4 of the appendix, suggests a standard dynamic programming line of attack for the surrogate problem (P3′): optimal continuation cost functions $\{Q^k\}_{1 \leq k \leq n}$ can be computed via backward induction, and these can then be applied to determine an optimal policy. However, the dimension of this dynamic program is given by the number of covariates p. In general, the computational effort required by this approach will be exponential in p — this is the so-called curse of dimensionality. Thus, outside of very small numbers of covariates, say, $p \leq 3$, the standard dynamic programming approach is intractable. However, as we will now see, that the surrogate problem surprisingly admits an alternative, low dimensional dynamic programming representation.

4.4.4 State Space Collapse

Proposition 7 yields a dynamic programming approach for the surrogate problem (P3′) that is intractable for all but very small values of p. What is remarkable, however, is that if the covariate

data is assumed to have an *elliptical distribution*, then (P3′) can be solved via a tractable *two*-dimensional dynamic program. We first present the technical definition.

Definition 1. *A random variable X taking values in $\mathbb{R}^m$ has an elliptical distribution if the characteristic function $\varphi\colon \mathbb{C}^m \to \mathbb{C}$ has the form*

$$\varphi(t) \triangleq \mathsf{E}\left[\exp(it^\top X)\right] = \exp(i\mu^\top t)\Psi(t^\top \Sigma t),$$

for all $t \in \mathbb{C}^m$, given some $\mu \in \mathbb{R}^m$, $\Sigma \in \mathbb{R}^{m\times m}$, and a characteristic function $\Psi\colon \mathbb{C} \to \mathbb{C}$.

Elliptical distributions, studied extensively, for example, by [Cambanis *et al.*, 1981], are a generalization of the multivariate Gaussian distribution. The name derives from the fact that if an elliptical distribution has a density, then the contours of the density are ellipsoids in $\mathbb{R}^m$ parameterized by μ and Σ. A useful standard result for us [Cambanis *et al.*, 1981] is that these distributions can be generated by independently generating the direction and the length of the deviation (in $\|\cdot\|_{\Sigma^{-1}}$-norm) from the center μ:

Proposition 8. *If X has an elliptical distribution with parameters μ, Σ, and Ψ, then there exists a non-negative random variable R such that,*

$$X \overset{d}{=} \mu + R\Sigma^{1/2}U,$$

where U is distributed uniformly on the unit sphere $\{x \in \mathbb{R}^{p-1} \mid \|x\|_2^2 = 1\}$ and U and R are independent.

Thus, any elliptical distribution can be identified with a vector $\mu \in \mathbb{R}^m$, a positive semi-definite matrix $\Sigma \in \mathbb{R}^{m\times m}$, and random variable R taking values on the non-negative real line. We denote such a distribution by $\mathrm{Ell}(\mu, \Sigma, R)$. It can be shown that if $R^2 \sim \chi^2_m$ is a chi-squared distribution with m degrees of freedom, then $\mathrm{Ell}(\mu, \Sigma, R)$ is a Gaussian distribution with mean μ and covariance Σ. Well-known distributions such as the multivariate t-distribution, Cauchy distribution, and logistic distribution also fall in the elliptical family.

We state the assumption needed for the state space collapse.

Assumption 6. *Given (n, p) with $1 \le p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n\times p}$ has independent and identically distributed rows. Further, assume that for each $1 \le k \le n$, the kth*

row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector $Z_{k,2:p} \in \mathbb{R}^{p-1}$ of all components except the first is distributed according to $\mathrm{Ell}(0, \Sigma, R)$*, where it is assumed that the random variable R has finite second moment, and further that, without loss of generality,*[1] $\mathsf{E}[R^2] = p - 1$.

The following theorem shows how the p-dimensional dynamic program is reduced to a 2-dimensional one with Assumption 6.

Theorem 10. *Suppose that Assumption 6 holds. For each $\ell \geq 1$, define the function $q^{\ell,p} \colon \mathbb{Z} \times \mathbb{R}_+ \to \mathbb{R}$ according to*

$$q^{\ell,p}(m, \lambda) \triangleq \begin{cases} m^2 + \lambda, & \text{if } \ell = 1, \\ \mathsf{E}\left[\min\limits_{u \in \{\pm 1\}} \; q^{\ell-1,p}\left(m + u, \lambda + 2uRU_1\sqrt{\lambda} + R^2\right)\right], & \text{if } \ell > 1. \end{cases} \tag{4.9}$$

Here, when $\ell > 1$, the expectation is taken over independent random variables U and R that are the random variables in the stochastic decomposition of $Z_{1,2:p}$ from Assumption 6. Then,

1. *At each time k, the optimal continuation cost for the dynamic program (P3′) is given by*

$$Q^k(\delta_k, \Delta_k) = q^{n-k+1,p}\left(\delta_k, \|\Delta_k\|^2_{\Sigma^{-1}}\right).$$

   In other words, this is the expected terminal cost, given then covariates observed and the allocations made up to and including time k, assuming optimal decisions are made at all future times.

2. *Suppose the allocation x_k^* at each time k is made according to*

$$x_k^* \in \operatorname*{argmin}_{u \in \{\pm 1\}} \; q^{n-k+1,p}\left(\delta_{k-1} + u, \|\Delta_{k-1} + uZ_k\|^2_{\Sigma^{-1}}\right). \tag{4.10}$$

   Then, the sequence of allocations x^* is optimal for the online experiment design problem (P3′).

For the case of Gaussian distribution, the recursion (4.9) for solving the DP can be simplified according to the following corollary:

[1] Note that under our assumption, it is easy to verify that each covariate vector $Z_{k,2:p}$ is zero mean. Our choice of normalization $\mathsf{E}[R^2] = p - 1$ ensures that the covariance matrix of $Z_{k,2:p}$ is given by Σ.

Corollary 3. *If Assumption 4 holds, then, for $\ell \geq 1$ and $p \geq 2$, the functions $q^{\ell,p}_{\text{gauss}} \colon \mathbb{Z} \times \mathbb{R}_+ \to \mathbb{R}$ are given by*

$$q^{\ell,p}_{\text{gauss}}(m,\lambda) \triangleq \begin{cases} m^2 + \lambda, & \text{if } \ell = 1, \\ \mathsf{E}\left[\min\limits_{u \in \{\pm 1\}} q^{\ell-1,p}_{\text{gauss}}\left(m+u, \left(\sqrt{\lambda} + u\eta\right)^2 + \xi\right)\right], & \text{if } \ell > 1. \end{cases} \tag{4.11}$$

Here, when $\ell > 1$, the expectation is taken over independent random variables $(\eta, \xi) \in \mathbb{R}^2$, where $\eta \sim N(0,1)$ is a standard normal random variable, and $\xi \sim \chi^2_{p-2}$ is chi-squared random variable with $p-2$ degrees of freedom.[2]

We defer the proof of Theorem 10 and Corollary 3 until Section 4.4.5 in order to make several remarks:

1. A key point is that, unlike the standard dynamic programming decomposition of Proposition 7, Theorem 10 provides a *tractable* way to solve the surrogate problem (P3′), independent of the covariate dimension p. This is because the recursion (4.9) yields a two-dimensional dynamic program. One of the state variables of this program, m, is discrete, taking values on the integers from $-n$ to n. Further, one can show that, with high probability, the second state variable λ is $O(n^2)$ thereby allowing us to discretize the state-space on a two-dimensional mesh. The functions $\{q^{\ell,p}\}_{\ell \geq 1}$ can be numerically evaluated on this grid via backward induction. Note that since the expectation in (4.9) is over a two-dimensional random variable, it can be computed via numerical integration. Further details of this procedure are given in Section 4.5.

2. Moreover, the functions $\{q^{\ell,p}\}_{\ell \geq 1}$ do not depend on the matrix Σ or the time horizon n. They only depend on the covariate dimension p. For example, in the Gaussian case, this means that if these functions are computed offline, they can subsequently be applied to *all* p-dimensional problem with a Gaussian data distribution.

3. Finally, the algorithm assumes that the covariance matrix Σ is known. This is needed to compute the $\|\cdot\|_{\Sigma^{-1}}$-norm of Δ_k. In practice, Σ may not be known, and may need to be estimated from data. However, observe that Σ depends only on the distribution of covariates

[2] If $p = 2$, we take $\xi \triangleq 0$.

across the subject population, not on the outcome of experiments. In the applications we have in mind, there is typically a wealth of information about this population known in advance of the experimental trials. Hence, Σ can be estimated offline even if the number of covariates p is large and the number of experimental subjects n is small.

For example, in an online advertising setting, and advertiser may want to compare two creatives using A-B testing with a limited number of experimental subjects. In advance of any experiments, the advertiser can use historical data from other trials or market surveys over the same population of subjects to estimate Σ.

4.4.5 Proof of Theorem 10

In essence, the proof of Theorem 10 relies on the symmetry of the elliptical distribution for each covariate vector $Z_{k,2:p}$. In particular, for orthonormal matrix $Q \in \mathbb{R}^{p-1\times p-1}$, $\Sigma^{-1/2}Z_{k,2:p}$ has the same distribution as $Q\Sigma^{-1/2}Z_{k,2:p}$. As a result of this spherical symmetry, under any non-anticipating policy, the distribution of the Mahalanobis distance $\|\Delta_{k+1}\|_{\Sigma^{-1}}$ at time $k+1$ is invariant across all Δ_k of a fixed Mahalanobis distance $\|\Delta_k\|_{\Sigma^{-1}}$ at time k. Thus, as opposed to having to maintain the p-dimensional state variable (δ_k, Δ_k), one merely needs to maintain the two-dimensional state variable $(\delta_k, \|\Delta_k\|_{\Sigma^{-1}})$.

To make this argument formal, we first define an inner product $\langle\cdot,\cdot\rangle_{\Sigma^{-1}}$ on $\mathbb{R}^{p-1}$ by

$$\langle \Delta, \Delta' \rangle_{\Sigma^{-1}} \triangleq \Delta^\top \Sigma^{-1} \Delta',$$

for $\Delta, \Delta' \in \mathbb{R}^{p-1}$. Using the symmetry of elliptical distribution, we can establish that:

Lemma 5. *Suppose $\Delta \in \mathbb{R}^{p-1}$ is a fixed $p-1$-dimensional vector and $X \sim \mathrm{Ell}(0, \Sigma, R)$ is an elliptically distributed $p-1$-dimensional random vector. Then,*

$$(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}}) \stackrel{d}{=} \left(R^2, R\|\Delta\|_{\Sigma^{-1}} U_1\right).$$

In particular, when $X \sim N(0, \Sigma)$ has a Gaussian distribution, then,

$$(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}}) \stackrel{d}{=} \left(\zeta^\top \zeta, \|\Delta\|_{\Sigma^{-1}} \zeta_1\right),$$

for an independent and normally distributed $p-1$-dimensional random vector $\zeta \sim N(0, I)$.

Proof. Since X follows the elliptical distribution,

$$X \stackrel{d}{=} R\Sigma^{1/2}U.$$

Thus,

$$\langle X, X\rangle_{\Sigma^{-1}} \stackrel{d}{=} R^2U^\top\Sigma^{1/2}\Sigma^{-1}\Sigma^{1/2}U = R^2.$$

Also,

$$\langle X, \Delta\rangle_{\Sigma^{-1}} \stackrel{d}{=} R\Delta^\top\Sigma^{-1/2}U.$$

But, by the symmetry of the distribution of U, for any $h \in \mathbb{R}^{p-1}$, $h^\top U$ is has the same distribution as $\|h\|_2 U_1$. Due to independence of U and R, $(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}})$ is distributed as $(R^2, R\|\Delta\|_{\Sigma^{-1}}U_1)$.

To prove the last statement, note that for the Gaussian case (R, U) is distributed as $(\|\zeta\|_2, \zeta/\|\zeta\|_2)$, if $\zeta \sim N(0, I)$. Thus,

$$(R^2, R\|\Delta\|_{\Sigma^{-1}}U_1) = \left(\|\zeta\|_2^2, \|\zeta\|_2\|\Delta\|_{\Sigma^{-1}}e_1^\top\frac{\zeta}{\|\zeta\|_2}\right) = (\zeta^\top\zeta, \|\Delta\|_{\Sigma^{-1}}\zeta_1).$$

■

Now we are ready to prove the theorem.

Proof of Theorem 10. We will prove, by backward induction over $1 \le k \le n$, that

$$Q^k(\delta_k, \Delta_k) = q^{n-k+1,p}\left(\delta_k, \|\Delta_k\|_{\Sigma^{-1}}^2\right) \tag{4.12}$$

holds for all $\delta_k \in \mathbb{Z}$, $\Delta_k \in \mathbb{R}^{p-1}$. The result will then follow from Proposition 7.

Comparing (4.8) and (4.9), (4.12) clearly holds for $k = n$.

Now, assume that (4.12) holds for $k+1$. Then, from (4.8),

$$\begin{aligned}
Q^k(\delta_k, \Delta_k) &= \mathsf{E}\left[\min_{u\in\{\pm1\}} q^{n-k,p}\left(\delta_k + u, \|\Delta_k + uZ_{k+1,2:p}\|_{\Sigma^{-1}}^2\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm1\}} q^{n-k,p}\left(\delta_k + u, \|\Delta_k\|_{\Sigma^{-1}}^2 + \|Z_{k+1,2:p}\|_{\Sigma^{-1}}^2 + 2u\langle Z_{k+1,2:p}, \Delta_{k+1}\rangle_{\Sigma^{-1}}\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm1\}} q^{n-k,p}\left(\delta_k + u, \|\Delta_k\|_{\Sigma^{-1}}^2 + R^2 + 2uRe_1^\top U\|\Delta\|_{\Sigma^{-1}}\right)\right] \\
&\triangleq q^{n-k+1,p}\left(\delta_k, \|\Delta_k\|_{\Sigma^{-1}}^2\right).
\end{aligned} \tag{4.13}$$

The third equality follows from Lemma 5. ■

Finally, we prove Corollary 3.

Proof of Corollary 3. Following the proof of Theorem 10, we will simplify the expression for (4.13). In particular, using the final part of Lemma 5,

$$\begin{aligned}
Q^k(\delta_k, \Delta_k) &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{n-k,p}_{\text{gauss}}\left(\delta_k + u, \|\Delta_k + uZ_{k+1,2:p}\|^2_{\Sigma^{-1}}\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{n-k,p}_{\text{gauss}}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + R^2 + 2uRe_1^\top U\|\Delta\|_{\Sigma^{-1}}\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{n-k,p}_{\text{gauss}}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + \zeta^\top\zeta + 2u\zeta_1\|\Delta\|_{\Sigma^{-1}}\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{n-k,p}_{\text{gauss}}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + \xi + \eta^2 + 2u\eta\|\Delta\|_{\Sigma^{-1}}\right)\right] \\
&= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{n-k,p}_{\text{gauss}}\left(\delta_k + u, (\|\Delta_k\|_{\Sigma^{-1}} + u\eta)^2 + \xi\right)\right].
\end{aligned}$$

Here, $\xi \sim \chi^2_{p-2}$ if $p > 2$ and $\xi \triangleq 0$ if $p = 2$, and $\eta \sim N(0,1)$ are independent of each other. ■

4.5 Experiments

This section focuses on numerical experiments with data. Our goal is to show the gain our algorithm for online allocation enjoys over randomization as one varies the two key parameters of interest — the number of covariates p and the number of samples n; as such this complements our theoretical analysis on the value of optimization in the offline setting.

In Section 4.4 we showed that if the covariates follow a distribution that falls in the elliptical family then then sequential optimization problem can be solved using a 2-dimensional dynamic program. In real applications this might not be true. For example, some of the covariates might be discrete or binary which rules out elliptical distributions. In this section we show that policy obtained by solving the 2-dimensional dynamic program gives a large improvement in efficiency over randomization in realistic scenarios.

4.5.1 Efficiency Gain

As will be discussed shortly in Section 4.5.3, we are principally be interested in comparing our dynamic programming allocation policy with randomization. Our performance metric of interest is the efficiency of these procedures. In the present setting, from (4.1), the expected efficiency of

the dynamic programming policy is given by

$$\mathrm{Eff}_{\mathrm{dp}} = \frac{\mathsf{E}\left[x_{\mathrm{dp}}{}^{\top} P_{Z^{\perp}} x_{\mathrm{dp}}\right]}{\sigma^2},$$

where the allocations x_{dp} are determined by the policy, and the expectation is taken over realizations of the covariate matrix Z and the resulting allocations x_{dp}. On the other hand, the efficiency of randomization is known from Theorem 7 to be

$$\mathrm{Eff}_{\mathrm{rand}} = \frac{n}{\sigma^2}\left(1 - \frac{p-1}{n-1}\right).$$

The results in a elative efficiency of

$$\frac{\mathrm{Eff}_{\mathrm{dp}}}{\mathrm{Eff}_{\mathrm{rand}}} = \frac{\mathsf{E}\left[x_{\mathrm{dp}}{}^{\top} P_{Z^{\perp}} x_{\mathrm{dp}}\right]}{n\left(1 - \frac{p-1}{n-1}\right)}. \tag{4.14}$$

We call this quantity the *efficiency gain* of the dynamic programming approach.

Observe that the efficiency gain depends only on the distribution of covariates. In particular, estimating the efficiency gain *does not* require observations of experimental outcomes y_k. From an empirical perspective, this is helpful since it means that we can assess the allocation policies given only access to covariate distributions, as opposed to requiring observations of outcomes which would necessitate running actual experimential trials under different allocation policies. Indeed, the efficiency gain provides a relative assessment that holds for *any* outcome which can be modelled via linear dependencies on the covariates and treatments.

4.5.2 Data

We run our experiments on two different data distributions for the covariates. Assumption 5 holds in both cases. Thus $\{Z_k\}$ are i.i.d. and $Z_{k,1}$ is assumed to be 1. We run our experiments with the following sampling distributions for $Z_{2:p}$:

Synthetic Data. In this case we consider the ideal case scenario where $Z_{2:p}$ follow the Gaussian distribution which falls in the elliptical. In other words Assumption 4 is satisfied. For the covariance matrix Σ, we set $\Sigma_{ii} = 1.0$ and $\Sigma_{ij} = 0.1$ for a $j \neq i$.

Yahoo User Data. To experiment on data from a more realistic setting, we use a dataset on user click log on the Yahoo![3] The users here are visitors to 'Featured Tab of the Today Module' on tha

[3]This dataset is obtained from the Yahoo! Labs repository of datasets available for academic research, and can be

Yahoo! front page. In the dataset, each user has 136 features associated with it such as age and gender. Each feature is binary, taking the values in $\{0, 1\}$. Some of these features are either 0 or 1 throughout the data set, and we discard these since they do not provide any additional information. The data also seems to have duplicate columns, which we again discard since these are redundant.

Our algorithm as an input requires the covariance matrix of the data. For this purpose, we estimate the covariance matrix from a portion of the dataset. This estimate is obtained by simply taking a sample average across roughly 1 million data points kept aside.

Finally, for evaluation purposes, we need a generative model for the data. To this end, from a set of 1 million data points we sample users, with replacement. In other words, as the sampling distribution we use the empirical distribution of the 1 million data points used for testing. Such a sampling procedure is intended to mimic arrival of users on Yahoo! front page.

4.5.3 Algorithms

Dynamic Programming (Our Approach). We approach this sequential problem using our dynamic programming based approach from Section 4.4. As the approximation for the p-dimensional value functions, we use the 2-dimensional value functions given by $\{q^{\ell,p}_{\text{gauss}}\}_{\ell\geq 1}$. These functions depend only on p and can be computed offline numerically. Here we provide the details of how these are evaluated. $\{q^{\ell,p}_{\text{gauss}}\}_{\ell\geq 1}$ are computed using backward induction. In particular, given $q^{\ell-1,p}_{\text{gauss}}$ we compute $q^{\ell,p}_{\text{gauss}}$ as follows:

1. Discretization: The first state variable m is discrete and can take values from $-n$ to n. We consider values for the second state variable λ on a mesh of of the interval $[0, M]$. M is conservatively set at a high value, i.e., a value such that $\|\Delta_k\|^2_{\Sigma^{-1}}$ has a low probability of exceeding M. In our case, we choose $M = 2 \times 10^3$ and a mesh step-size of 0.2. Using this procedure we have a 2-dimensional grid on the state space of the dynamic program.

2. Sampling: At each point (m, λ) such that $-n \leq m \leq n$ and λ is the mesh of $[0, M]$, we estimate $q^{\ell,p}_{\text{gauss}}(m, \lambda)$ via Monte Carlo sampling, In particular, N pairs (ξ, η) are sampled from the given distribution and $q^{\ell,p}_{\text{gauss}}(m, \lambda)$ is estimated according to (4.9) using the corresponding

downloaded as "R6B — Yahoo! Front Page Today Module User Click Log Dataset, version 2.0" at `http://webscope.sandbox.yahoo.com/catalog.php?datatype=r`.

empirical measure. We use the same sample set for all (m, λ) at which this is evaluated. The number of sample points (N) is chosen to be 10,000.

3. Interpolation: In the right side of (4.9), estimates of the value of the function $q^{\ell-1,p}_{\text{gauss}}(m', \lambda')$ for various values of (m', λ') are obtained by linearly interpolating the closest values of the 'quantized' function described in the previous step; i.e., $q^{\ell-1,p}_{\text{gauss}}(m', \lambda')$ linearly interpolates $q^{\ell-1,p}_{\text{gauss}}(m', \lambda_l)$ and $q^{\ell-1,p}_{\text{gauss}}(m', \lambda_r)$ where $\lambda_l \leq \lambda' \leq \lambda_r$ are closest to λ' on the mesh used for computing $q^{\ell-1,p}_{\text{gauss}}$.

Randomization. This is the algorithm described in Section 4.3.2, where half of the subjects are assigned to the control group, uniformly at random. In randomization we can perform all allocations up front at time $t = 0$. This way it can be considered as a sequential procedure since we do not use information from future arrivals to make decisions.

4.5.4 Results

As an evaluation criteria, we use the relative efficiency gain, given by (4.14). Here, the expectation in the numerator of (4.14) is approximated through Monte Carlo simulation, which requires sampling covariates Z from the aforementioned Gaussian and Yahoo! user data generative models, and computing the resulting dynamic programming allocations.

The aim of the experiments is to explore the gains of optimization for various values of the number of trials n and covariate dimension p. To this end, we plot the efficiency gain for values of n in the range $[p + 1, 100]$. We repeat this for various values p between 10 and 40. We perform 10,000 Monte Carlo trials for each combination of p and n to make sure that confidence intervals are narrow (standard errors for the efficiency gain are uniformly below 0.01).

The results are plotted in Figure 4.1 and Figure 4.2. We see that there are significant gains to be obtained by optimization, both for the Gaussian modeland Yahoo! use data model. Further, this improvement grows as the ratio of p/n gets closer to 1. The efficiency ratio is close to 3 in some cases. This is consistent with the theory of the offline optimal design studied in Section 4.3, where we saw that randomization gives $O(1)$ performance when $p \sim n$ and optimized designs are $\Omega(n)$. As n increases the efficiency gain decreases. However, even for the extreme case of $p = 10$ and $n = 100$, there is close to 10% improvement in efficiency for both data sets.

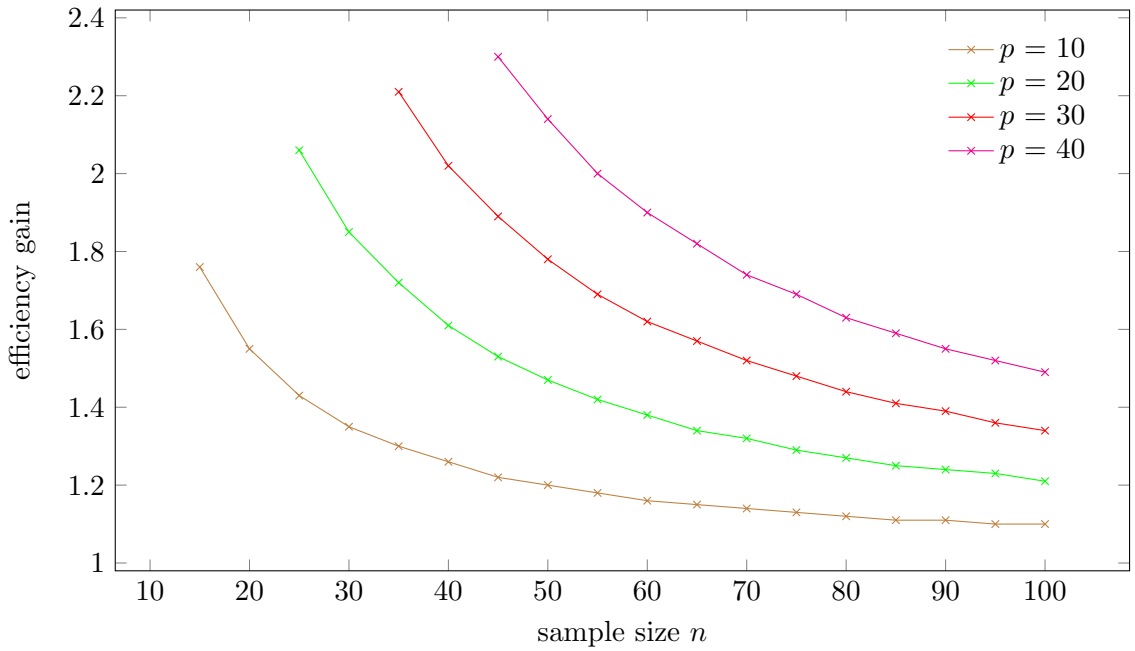

Figure 4.1: Efficiency gain (the ratio of the efficiencies of the optimal and the randomized design) as a function of the sample size n and the covariate dimension p for the Gaussian dataset.

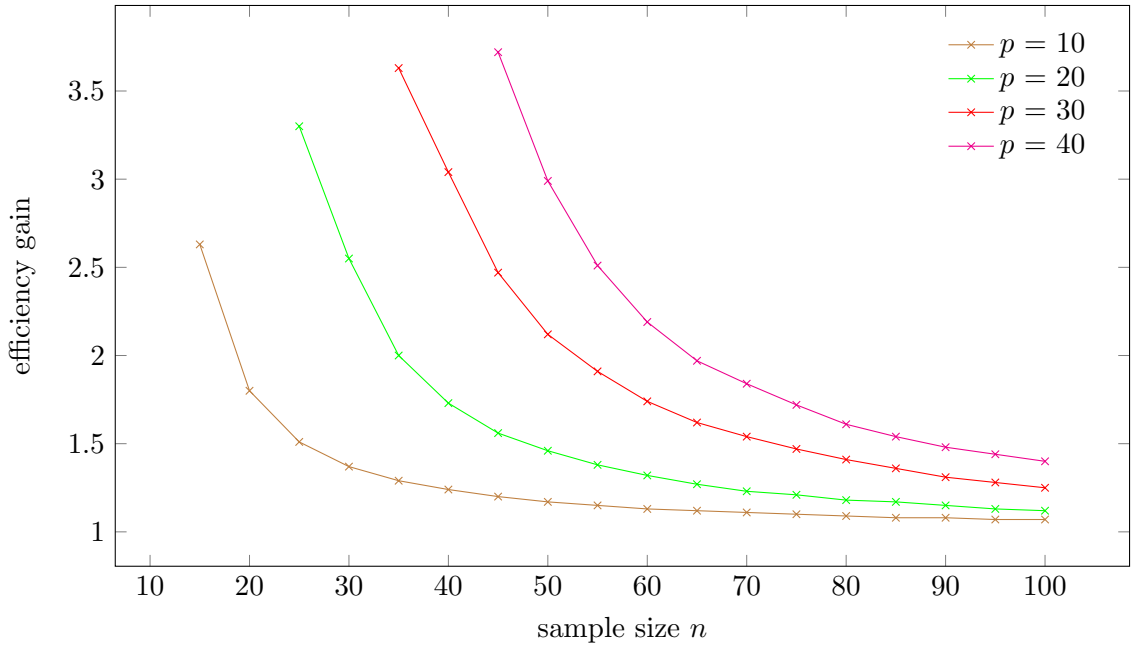

Figure 4.2: Efficiency gain (the ratio of the efficiencies of the optimal and the randomized design) as a function of the sample size n and the covariate dimension p for the Yahoo! dataset.

4.6 Conclusions

We conclude with a summary of what we have accomplished and what we view as key directions for further research. At a conceptual level, this paper illustrates the power of the 'optimization' viewpoint in what are inherently statistical problems: we have presented a provably optimal solution to a problem for which a plethora of heuristics were available. In addition to establishing the appropriate approach to this problem, the algorithms we have developed are eminently practical and easy to implement — a property that is crucial for the sorts of applications that motivated this work. On a more pragmatic note, we have quantified the *value* of these sorts of optimization approaches establishing precise estimates of the benefits optimization approaches provide over straightforward randomization. These estimates illustrate that in so-called high dimensional setting — i.e., in settings where the number of covariates is large, such approaches can provide order of magnitude improvements in sampling efficiency.

A number of directions remain for future research. We highlight several here in parting:

1. Normality: To what extent can our assumption on the normality of covariates be relaxed? Can we develop approximation guarantees for the situation when covariates are not normally distributed?

2. Non-linear models: Can we allow for a nonlinear dependence on covariates? One direction to accomplish this is perhaps a reliance of some manner of non-parametric 'kernel' approach. The good news here is that the value of optimization is likely to be even higher in such an infinite-dimensional setting.

3. More than two alternatives: The present paper considers only the two alternative setting, an important direction for future work would be to consider settings where there is a larger number of choices.

Part I

Bibliography

Bibliography

[Adelman, 2007] Daniel Adelman. Dynamic bid prices in revenue management. *Operations Research*, 55(4):647–661, 2007.

[Atkinson, 1982] A. C. Atkinson. Optimum biased coin designs for sequential clinical trials with prognostic factors. *Biometrika*, 69(1):61–67, 1982.

[Atkinson, 1999] A. C. Atkinson. Optimum biased-coin designs for sequential treatment allocation with covariate information. *Statistics in Medicine*, 18(14):1741–1752, 1999.

[Barreto *et al.*, 2011] A. M. S. Barreto, D. Precup, and J. Pineau. Reinforcement learning using kernel-based stochastic factorization. In *Advances in Neural Information Processing Systems*, volume 24, pages 720–728. MIT Press, 2011.

[Bartlett and Mendelson, 2002] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[Ben-Tal and Nemirovski, 2001] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics, 2001.

[Bertsekas, 2013] Dimitri P Bertsekas. Abstract dynamic programming. *Athena Scientific, Belmont, MA*, 2013.

[Bethke *et al.*, 2008] B. Bethke, J. P. How, and A. Ozdaglar. Kernel-based reinforcement learning using bellman residual elimination. *Journal of Machine Learning Research (to appear)*, 2008.

[Buchbinder *et al.*, 2007] Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Algorithms–ESA 2007*, pages 253–264. Springer, 2007.

[Cambanis *et al.*, 1981] Stamatis Cambanis, Steel Huang, and Gordon Simons. On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis*, 11(3):368–385, 1981.

[Chen and Meyn, 1998] R. R. Chen and S. Meyn. Value iteration and optimization of multiclass queueing networks. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 1, pages 50 –55 vol.1, 1998.

[Chick and Frazier, 2012] S. E. Chick and P. Frazier. Sequential sampling with economics of selection procedures. *Management Science*, 58(3):550–569, 2012.

[Chick and Gans, 2009] S. E. Chick and N. Gans. Economic analysis of simulation selection problems. *Management Science*, 55(3):421–437, 2009.

[Cook *et al.*, 1979] T. D. Cook, D. T. Campbell, and A. Day. *Quasi-Experimentation: Design & Analysis Issues for Field Settings*. Houghton Mifflin Boston, 1979.

[Dai and Lin, 2005] J. G. Dai and W. Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2):197–218, 2005.

[de Farias and Van Roy, 2003] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[de Farias and Van Roy, 2004] D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29:462–478, 2004.

[Derman *et al.*, 1972] Cyrus Derman, Gerald J Lieberman, and Sheldon M Ross. A sequential stochastic assignment problem. *Management Science*, 18(7):349–355, 1972.

[Derman *et al.*, 1975] Cyrus Derman, Gerald J Lieberman, and Sheldon M Ross. A stochastic sequential allocation model. *Operations Research*, 23(6):1120–1130, 1975.

[Desai *et al.*, 2011] V. V. Desai, V. F. Farias, and C. C. Moallemi. Approximate dynamic programming via a smoothed linear program. To appear in *Operations Research*, 2011.

[Dietterich and Wang, 2002] T. G. Dietterich and X. Wang. Batch value function approximation via support vectors. In *Advances in Neural Information Processing Systems*, volume 14, pages 1491–1498. MIT Press, 2002.

[Efron, 1971] B. Efron. Forcing a sequential experiment to be balanced. *Biometrika*, 58(3):403–417, 1971.

[Engel *et al.*, 2003] Y. Engel, S. Mannor, and R. Meir. Bayes meets Bellman: The Gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning*, pages 154–161. AAAI Press, 2003.

[Ernst *et al.*, 2005] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, April 2005.

[Farahmand *et al.*, 2009] A. M. Farahmand, M. Ghavamzadeh, S. Mannor, and C. Szepesvári. Regularized policy iteration. In *Advances in Neural Information Processing Systems*, pages 441–448, 2009.

[Fisher, 1935] R. A. Fisher. *The Design of Experiments*. Oliver & Boyd, 1935.

[G *et al.*, 1998] Opelz G, Wujciak T, and Ritz E. Association of chronic kidney graft failure with recipient blood pressure. collaborative transplant study. *Kidney Int.*, 53(1), 1998.

[Goel and Mehta, 2008] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 982–991, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[Goemans and Williamson, 1995] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[Harrison and Wein, 1989] J. M. Harrison and L. M. Wein. Scheduling network of queues: Heavy traffic analysis of a simple open network. *Queueing Systems*, 5:265–280, 1989.

[James, 1954] A. T. James. Normal multivariate analysis and the orthogonal group. *The Annals of Mathematical Statistics*, pages 40–75, 1954.

[Joachims, 1999] T. Joachims. *Making large-scale support vector machine learning practical*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[Kallus, 2012] N. Kallus. The power of optimization over randomization in designing experiments involving small samples. Working paper, 2012.

[Kallus, 2013] N. Kallus. Regression-robust designs of controlled experiments. Working paper, 2013.

[Kapelner and Krieger, 2013] A. Kapelner and A. Krieger. Matching on-the-fly in sequential experiments for higher power and efficiency. Working paper, 2013.

[Karp *et al.*, 1990] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[Kim and Nelson, 2006] S.-H. Kim and B. L. Nelson. Selecting the best system. *Handbooks in operations research and management science*, 13:501–534, 2006.

[Kolter and Ng, 2009] J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 521–528. ACM, 2009.

[Kumar and Muthuraman, 2004] S. Kumar and K. Muthuraman. A numerical method for solving singular stochastic control problems. *Operations Research*, 52(4):563–582, 2004.

[Kumar and Seidman, 1990] P. R. Kumar and T. I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control*, 35(3):289 –298, March 1990.

[Kushner and Martins, 1996] H. J. Kushner and L. F. Martins. Heavy traffic analysis of a controlled multiclass queueing network via weak convergence methods. *SIAM J. Control Optim.*, 34(5):1781–1797, 1996.

[Lawler, 2001] Eugene L Lawler. *Combinatorial optimization: networks and matroids*. Courier Dover Publications, 2001.

[Luenberger, 1997] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, Inc., 1997.

[Manne, 1960] A. S. Manne. Linear programming and sequential decisions. *Management Science*, 6(3):pp. 259–267, 1960.

[Martins *et al.*, 1996] L. F. Martins, S. E. Shreve, and H. M. Soner. Heavy traffic convergence of a controlled multiclass queueing network. *SIAM J. Control Optim.*, 34(6):2133–2171, 1996.

[Mehta *et al.*, 2005] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized on-line matching. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 264–273. IEEE, 2005.

[Moallemi *et al.*, 2008] C. C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Working Paper, 2008.

[Nesterov, 1997] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. Technical report, Université Catholique de Louvain, Center for Operations Research and Econometrics, 1997.

[Ormoneit and Glynn, 2002] D. Ormoneit and P. Glynn. Kernel-based reinforcement learning in average cost problems. *IEEE Transactions on Automatic Control*, 47(10):1624–1636, 2002.

[Ormoneit and Sen, 2002] D. Ormoneit and S. Sen. Tree-based batch mode reinforcement learning. *Machine Learning*, 49(2):161–178, 2002.

[Osuna *et al.*, 1997] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Neural Networks for Signal Processing, Proceedings of the 1997 IEEE Workshop*, pages 276 –285, sep 1997.

[Patrick *et al.*, 2008] J. Patrick, M. L. Puterman, and M. Queyranne. Dynamic multipriority patient scheduling for a diagnostic resource. *Operations Research*, 56(6):1507–1525, 2008.

[Pazis and Parr, 2011] J. Pazis and R. Parr. Non-parametric approximate linear programming for MDPs. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[Pekelis *et al.*, 2015] Leo Pekelis, David Walsh, and Ramesh Johari. The new stats engine, 2015. Available at `http://pages.optimizely.com/rs/optimizely/images/stats_engine_technical_paper.pdf`.

[Petrik *et al.*, 2010] M. Petrik, G. Taylor, R. Parr, and S. Zilberstein. Feature selection using regularization in approximate linear programs for Markov decision processes. In *Proceedings of the 27th Annual International Conference on Machine Learning*, pages 871–879. ACM, 2010.

[Pocock and Simon, 1975] S. J. Pocock and R. Simon. Sequential treatment assignment with balancing for prognostic factors in the controlled clinical trial. *Biometrics*, pages 103–115, 1975.

[Pukelsheim, 2006] F. Pukelsheim. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics, 2006.

[Raudenbush *et al.*, 2007] S. W. Raudenbush, A. Martinez, and J. Spybrook. Strategies for improving precision in group-randomized experiments. *Educational Evaluation and Policy Analysis*, 29(1):5–29, 2007.

[Rosenberger and Sverdlov, 2008] W. F. Rosenberger and O. Sverdlov. Handling covariates in the design of clinical trials. *Statistical Science*, pages 404–419, 2008.

[Roth *et al.*, 2004] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.

[Scholkopf and Smola, 2001] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.

[Schrijver, 2000] A. Schrijver. *A course in combinatorial optimization*. TU Delft, 2000.

[Schweitzer and Seidman, 1985] P. Schweitzer and A. Seidman. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.

[Smith, 1984a] R. L. Smith. Properties of biased coin designs in sequential clinical trials. *The Annals of Statistics*, pages 1018–1034, 1984.

[Smith, 1984b] R. L. Smith. Sequential treatment allocation using biased coin designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 519–543, 1984.

[Steensma and Kantarjian, 2014] D. P. Steensma and H. M. Kantarjian. Impact of cancer research bureaucracy on innovation, costs, and patient care. *Journal of Clinical Oncology*, 32(5):376–378, 2014.

[Stolyar, 2004] A. L. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14:1–53, 2004.

[Su and Zenios, 2005] Xuanming Su and Stefanos A Zenios. Patient choice in kidney allocation: A sequential stochastic assignment model. *Operations research*, 53(3):443–455, 2005.

[Tassiulas and Ephremides, 1992] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

[Van Roy, 2006] B. Van Roy. Performance loss bounds for approximate value iteration with state aggregation. *Mathematics of Operations Research*, 31(2):234–244, 2006.

[Vershynin, 2012] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, pages 210–268. Cambridge University Press, 2012.

[Woodroofe, 1979] M. Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.

[Xu *et al.*, 2007] X. Xu, D. Hu, and X. Lu. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 18(4):973–992, 2007.

[Zenios *et al.*, 2000] Stefanos A Zenios, Glenn M Chertow, and Lawrence M Wein. Dynamic allocation of kidneys to candidates on the transplant waiting list. *Operations Research*, 48(4):549–569, 2000.

Part II

Appendices

Appendix A

Non-parametric ADP

A.1 Duality of the Sampled RSALP

Proof of Proposition 1. We begin with a few observations about the primal program, (2.5):

1. Because the objective function is coercive,[1] weight vector $\mathbf{z}$ can be restricted without loss of generality to some finite ball in $\mathcal{H}$. The optimal value of the primal is consequently finite.

2. The primal has a feasible interior point: consider setting

$$\mathbf{z} \triangleq \mathbf{0}, \quad b \triangleq 0, \quad s_x \triangleq \max(-\min_a g_{x,a}, \epsilon),$$

   for some $\epsilon > 0$.

3. The optimal value of the primal is achieved. To see this, we note that it suffices to restrict $\mathbf{z}$ to the finite dimensional space spanned by the vectors $\{\mathbf{x} : x \in \hat{\mathcal{S}} \cup \mathcal{N}(\hat{\mathcal{S}})\}$, where $\mathcal{N}(\hat{\mathcal{S}})$ denotes the set of states that can be reached from the sampled states of $\hat{\mathcal{S}}$ in a single transition. Then, the feasible region of the primal can be restricted, without loss of optimality, to a compact subset of $\mathcal{H} \times \mathbb{R}^{\hat{\mathcal{S}}} \times \mathbb{R}$. Since the objective function of the primal is continuous, we know that its optimal value must be achieved by the Weierstrass theorem.

[1] As $\|\mathbf{z}\|_{\mathcal{H}} \to \infty$, the objective value goes to $-\infty$.

We next derive the dual to (2.5). As in [Chapter 8 [Luenberger, 1997]], we define the Lagrangian:

$$\mathcal{L}(\mathbf{z}, b, s, \lambda) \triangleq \left\langle -\frac{1}{N}\sum_{x\in\hat{\mathcal{S}}} w_x \mathbf{x} + \sum_{(x,a)\in\hat{\mathcal{S}}\times\mathcal{A}} \lambda_{x,a}\left(\mathbf{x} - \alpha \mathsf{E}_{x,a}[\mathbf{X}']\right), \mathbf{z}\right\rangle + \frac{\Gamma}{2}||\mathbf{z}||^2_{\mathcal{H}}$$
$$+ \sum_{x\in\hat{\mathcal{S}}} s_x \left(\frac{\kappa}{N} - \sum_{a\in\mathcal{A})} \lambda_{x,a}\right) - b\left(1 - (1-\alpha)\sum_{(x,a)\in\hat{\mathcal{S}}\times\mathcal{A}} \lambda_{x,a}\right) - \sum_{(x,a)\in\hat{\mathcal{P}}} g_{x,a}\lambda_{x,a}.$$

and define the dual function $G(\lambda) \triangleq \inf_{(\mathbf{z},b,s)\in\mathcal{D}} \mathcal{L}(\mathbf{z}, b, s, \lambda)$ where we denote by $\mathcal{D}$ the feasible region of the primal problem. Now, observe that for any given λ, $\mathcal{L}(\mathbf{z}, b, s, \lambda)$ is (uniquely) minimized at

$$\mathbf{z}^*(\lambda) = \frac{1}{\Gamma}\left[\frac{1}{N}\sum_{x\in\hat{\mathcal{S}}} w_x \mathbf{x} - \sum_{x\in\hat{\mathcal{S}}, a\in\mathcal{A}} \lambda_{x,a}\Big(\mathbf{x} - \alpha \mathsf{E}_{x,a}[\mathbf{X}']\Big)\right], \tag{A.1}$$

for any finite b, s. This follows from the observation that for any $\overline{\mathbf{z}} \in \mathcal{H}$, $\langle \mathbf{z}, \mathbf{z}\rangle - \langle \bar{\mathbf{z}}, \mathbf{z}\rangle$ is minimized at $-\frac{1}{2}\overline{\mathbf{z}}$ by the Cauchy-Schwartz inequality. It follows immediately that on the set defining the feasible region of the program (2.8), we must have that

$$G(\lambda) = \frac{1}{2}\lambda^\top Q\lambda + R^\top\lambda + S$$

and moreover that $G(\lambda) = +\infty$ outside that set. This suffices to establish that the dual problem $\inf_{\lambda\geq 0} G(\lambda)$ is precisely program (2.8).

The first conclusion of [**?**, Theorem 1, pp. 224–225 [Luenberger, 1997]]nd the first and second observations we made at the outset of our proof then suffice to establish that programs (2.5) and (2.8) have equal optimal values (i.e. strong duality holds) and that the optimal value of the dual program is achieved at some λ^*. Our third observation, (A.1), and the second conclusion of [Theorem 1, pp. 224–225[Luenberger, 1997]] then suffice to establish our second claim. ■

A.2 Proof of Lemma 1

Proof of Lemma 1. We begin with some preliminaries: Define

$$Z(X_1, X_2, \ldots, X_n) \triangleq \sup_{f\in\mathcal{F}} \mathsf{E}f(X) - \hat{E}_n f(X).$$

Notice that

$$\left|Z(X_1, X_2, \ldots, X_i, \ldots, X_n) - Z(X_1, X_2, \ldots, X_i', \ldots, X_n)\right| \leq \frac{2\overline{B}}{n}.$$

McDiarmid's inequality (or equivalently, Azuma's inequality) then implies:

$$\mathsf{P}\left(Z - \mathsf{E}Z \geq \sqrt{\frac{2\overline{B}^2 \ln(1/\delta)}{n}}\right) \leq \delta. \tag{A.2}$$

Now,

$$\begin{aligned}
\mathsf{E}Z &= \mathsf{E}\left[\sup_{f\in\mathcal{F}} \mathsf{E}f(X) - \hat{\mathsf{E}}_n f(X)\right] \\
&= \mathsf{E}\left[\sup_{f\in\mathcal{F}} \mathsf{E}\hat{\mathsf{E}}_n f(X) - \hat{\mathsf{E}}_n f(X)\right] \\
&\leq \mathsf{E}\left[\sup_{f\in\mathcal{F}} \hat{\mathsf{E}}_n f(X') - \hat{\mathsf{E}}_n f(X)\right] \\
&= \mathsf{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \left(f(X_i') - f(X_i)\right)\right] \\
&\leq \mathsf{E}\left[\sup_{f\in\mathcal{F}} \frac{2}{n}\sum_{i=1}^{n} \sigma_i \left(f(X_i)\right)\right] \\
&= R_n(\mathcal{F})
\end{aligned}$$

With (A.2), this immediately yields the result. ■

A.3 Justification of Average Cost Objective

In our experiments we use the average cost associated a policy as our evaluation criterion. However, since our algorithm uses the discounted cost objective, on the surface, there seems to be some inconsistency. Here, using a straightforward proposition, we show that the average cost objective can be used as a proxy for discounted cost objective. The discounted cost objective J^μ is a function over the state space rather than a number. To scalarize this and use it as an objective we might want to take a weighted average of the function over all states. Thus, to be consistent, we would use, $\mathsf{E}_\pi J^\mu(X)$ as our evaluation criterion, for some distribution π.

For the purpose of this section assume that for a policy μ the associated Markov chain is irreducible. For a stationary policy μ, let π_μ be the (unique) limiting stationary distribution of the Markov chain induced by the policy. Further let,

$$\lambda_\mu \triangleq \mathsf{E}_{\pi_\mu} g(X, \mu(X)).$$

By the ergodic theorem,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} g(X_t, \mu(X_t)) \to \lambda_\mu.$$

Proposition 9. *Given that under the policy μ the Markov chain is irreducible,*

$$\lambda_\mu = (1-\alpha)\mathsf{E}_{\pi_\mu} J^\mu(X).$$

Proof.

$$J^\mu(x) = \mathsf{E}_{X_0=x} \sum_{t=0}^{\infty} \alpha^t g(X_t, \mu(X_t)).$$

Here, $\{X_t\}_{0\leq t}$ follows the distribution of the Markov chain induced by the policy μ. It follows that,

$$\begin{aligned}
\mathsf{E}_{X\sim\pi_\mu} J^\mu(X) &= \mathsf{E}_{X_0\sim\pi_\mu} \sum_{t=0}^{\infty} \alpha^t g(X_t, \mu(X_t)) \\
&= \sum_{t=0}^{\infty} \alpha^t \mathsf{E}_{X\sim\pi_\mu} g(X, \mu(X)) \\
&= \sum_{t=0}^{\infty} \alpha^t \lambda_\mu \\
&= \frac{1}{1-\alpha}\lambda_\mu.
\end{aligned}$$

The limit sum interchange is justified by bounded convergence since $|\alpha^t g(X_t, \mu(X_t))| \leq \|g\|_\infty$. Since π_μ is the invariant distribution, if $X_0 \sim \pi_\mu$, then $X_t \sim \pi_\mu$ for all t. ■

In our experiments, for a particular policy μ, we estimate λ_μ by Monte Carlo. It follows from the preceding result that this is a valid proxy for the discounted cost-to-go function J^μ, averaged over the state space with the weights π_μ.

Appendix B

Dynamic Matching Problems

B.1 SALP Proof

Before we begin the proof let us set up some more definitions. For any $0 \leq t < T$, let V^t be an operator defined by,

$$(V^t J_{t+1})(\mathcal{S}_t) \triangleq \max_{\pi \in \mathcal{P}(\mathcal{S}_t)} \left(w(\pi) + E_\pi J_{t+1}(\mathcal{S}_t)\right).$$

Here we set $J_T(\mathcal{S}_T) = 0$, for all $\mathcal{S}_T \in \mathcal{X}_T$. This operator maps functions of the type $\mathcal{X}_{t+1} \mapsto \mathbb{R}$ to $\mathcal{X}_t \mapsto \mathbb{R}$, for any $0 \leq t < T$. This is commonly known as the Bellman operator.

Similarly, we define V_μ^t, the Bellman operator for a particular policy μ, as,

$$(V_\mu^t J_{t+1})(\mathcal{S}_t) = w(\mu^t(\mathcal{S}_t)) + E_\mu J_{t+1}(\mathcal{S}_t).$$

We state some facts about V^t (and V_μ^t) without proof since these are standard results.

Fact 1. *V^t (and V_μ^t) are monotonic, for each $0 \leq t < T$. Thus if $J_{t+1} \geq J'_{t+1}$, then $V^t J_{t+1} \geq V^t J'_{t+1}$ ($V_\mu^t J_{t+1} \geq V_\mu^t J'_{t+1}$).*

Fact 2. *V^t (and V_μ^t), for each $0 \leq t < T$, are non-expansive with respect to the ∞ norm, i.e.*

$$\|V^t J_{t+1} - V^t J'_{t+1}\|_\infty \leq \|J_{t+1} - J'_{t+1}\|_\infty,$$

and,

$$\|V_\mu^t J_{t+1} - V_\mu^t J'_{t+1}\|_\infty \leq \|J_{t+1} - J'_{t+1}\|_\infty.$$

We now present a simple lemma that follows from the monotonicity of T_{μ^*}.

Lemma 6. *Let $s_t : \mathcal{X}_t \to \mathbb{R}$, for each $0 \leq t < T$, be such that,*

$$J_t + s_t \geq V_{\mu^*}^t J_{t+1}, \quad \forall\ 0 \leq t < T,$$

with $J_T = 0$. Then, for $0 \leq t < T$,

$$J_t(\mathcal{S}'_t) + \sum_{\tau=t}^{T-1} \mathsf{E}_{\mu^*}[s_\tau(\mathcal{S}_\tau)|\mathcal{S}_t = \mathcal{S}'_t] \geq J_t^*(\mathcal{S}'_t)$$

Proof. We prove this by backward induction. The condition is trivial for $t = T-1$ is the same as the assumption:

$$J_{T-1}(\mathcal{S}'_{T-1}) + s_{T-1}(\mathcal{S}'_T) \geq V_{\mu^*}^{T-1} J_T(\mathcal{S}'_{T-1}) = J_{T-1}^*(\mathcal{S}'_{T-1})$$

Now assume that the result is true for $t+1$ for some $t \in \{0, \ldots, T-1\}$. Thus,

$$J_{t+1}(\mathcal{S}'_{t+1}) + \sum_{\tau=t+1}^{T-1} \mathsf{E}_{\mu^*}[s_\tau(\mathcal{S}_\tau)|\mathcal{S}_{t+1} = \mathcal{S}'_{t+1}] \geq J_{t+1}^*(\mathcal{S}'_{t+1})$$

Using monotonicity of $V_{\mu^*}^t$ we get,

$$V_{\mu^*}^t J_{t+1}(\mathcal{S}'_t) \geq J_t^*(\mathcal{S}'_t) - \sum_{\tau=t+1}^{T-1} \mathsf{E}_{\mu^*}[s_\tau(\mathcal{S}_\tau)|\mathcal{S}_t = \mathcal{S}'_t].$$

Finally, we use the fact that $J_t + s_t \geq V_{\mu^*}^t J_{t+1}$. Thus,

$$J_t(\mathcal{S}'_t) + s_t(\mathcal{S}'_t) + \sum_{\tau=t+1}^{T-1} \mathsf{E}_{\mu^*}[s_\tau(\mathcal{S}_\tau)|\mathcal{S}_t = \mathcal{S}'_t] \geq J_t^*(\mathcal{S}'_t).$$

■

Finally we present the proof of Theorem 1.

Proof. **[Theorem 1]**

Given that J^{SALP} is the solution of the optimization problem of interest, augmented with $J_T^{SALP} = 0$, let us define s^{SALP} as the vectors of one-sided Bellman errors, i.e.,

$$s_t^{SALP}(\mathcal{S}_t) \triangleq \left(V^t J_{t+1}(\mathcal{S}_t) - J_t(\mathcal{S}_t)\right)^+,$$

for $0 \leq t < T$. Further, let P_{μ^*} be the transition kernel of the Markov chain of the MDP under the policy μ^*. We use this to compactly express $\mathsf{E}_{\mu^*}[s_\tau(\mathcal{S})|\mathcal{S}_t = \mathcal{S}'_t] = P_{\mu^*}^{\tau-t} s_\tau(\mathcal{S}'_t)$, for $\tau \geq t$. It is easy

to verify that J^{SALP} and s^{SALP} pair satisfies the conditions of the Lemma 6. Thus, using the new notation we can express the result of Lemma 6 as,

$$J_t + \sum_{\tau=t}^{T-1} P_{\mu^*}^{\tau-t} s_\tau \geq J_t^*.$$

Specifically, for $t = 0$,

$$J_0 + \sum_{\tau=0}^{T-1} P_{\mu^*}^{\tau} s_\tau \geq J_0^*.$$

Now it follows that,

$$\begin{aligned}
\|J_0^{SALP} - J_0^*\|_{1,\nu} &= \left\| J_0^{LP} - J_0^* + \sum_{t=0}^{T-1} P_{\mu^*}^t s_t^{SALP} - \sum_{t=0}^{T-1} P_{\mu^*}^t s_t^{SALP} \right\|_{1,\nu} \\
&\leq \left\| J_0^{SALP} - J_0^* + \sum_{t=0}^{T-1} P_{\mu^*}^t s_t^{SALP} \right\|_{1,\nu} + \left\| \sum_{t=0}^{T-1} P_{\mu^*}^t s_t^{SALP} \right\|_{1,\nu} \\
&= \mathsf{E}[J_0^{SALP}(\mathcal{S}_0) - J_0^*(\mathcal{S}_0)] + 2\mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[\sum_{t=0}^{T} s_t^{SALP}(\mathcal{S}_t) \right] \\
&= \mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[J_0^{SALP}(\mathcal{S}_0) + 2 \sum_{t=0}^{T} s_t^{SALP}(\mathcal{S}_t) \right] - \mathsf{E}[J_0^*(\mathcal{S}_0)].
\end{aligned}$$

Here the expectation is with a distribution sampled from policy μ^* with an initial distribution ν.

Now note that on the right side we have the objective function of the optimization problem. This is minimized at (J^{SALP}, s^{SALP}). Thus for any other pair (J, s^J), with s^J being the one-sided Bellman error corresponding to J, this objective must be greater that the value for (J^{SALP}, s^{SALP}). Thus we get,

$$\begin{aligned}
\|J_0^{SALP} - J_0^*\|_{1,\nu} &\leq \mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[J_0^{SALP}(\mathcal{S}_0) + 2 \sum_{t=0}^{T} s_t^{SALP}(\mathcal{S}_t) \right] - \mathsf{E}[J_0^*(\mathcal{S}_0)] \\
&\leq \mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[J_0(\mathcal{S}_0) + 2 \sum_{t=0}^{T} s_t^{J}(\mathcal{S}_t) \right] - \mathsf{E}[J_0^*(\mathcal{S}_0)] \\
&= \mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[J_0(\mathcal{S}_0) + 2 \sum_{t=0}^{T} (V^t J_{t+1}(\mathcal{S}_t) - J_t(\mathcal{S}_t))^+ \right] - \mathsf{E}[J_0^*(\mathcal{S}_0)]
\end{aligned}$$

Now we use the fact that $\mathsf{E}J(\mathcal{S}) \leq \|J\|_\infty$ for any vector J. Thus,

$$\begin{aligned}
\|J_0^{SALP} - J_0^*\|_{1,\nu} &\leq \mathsf{E}_{\mathcal{S}_0 \sim \nu, \mu^*} \left[J_0(\mathcal{S}_0) + 2 \sum_{t=0}^{T} (V^t J_{t+1}(\mathcal{S}_t) - J_t(\mathcal{S}_t))^+ \right] - \mathsf{E}[J_0^*(\mathcal{S}_0)] \\
&\leq \|J_0 - J_0*\|_\infty + 2 \sum_{t=0}^{T-1} \|V^t J_{t+1} - J_t\|_\infty
\end{aligned}$$

Finally note that,

$$\begin{aligned}\|V^t J_{t+1} - J_t\|_\infty &= \|V^t J_{t+1} - V^t J^*_{t+1} + J^*_t - J_t\|_\infty \\ &\leq \|V^t J_{t+1} - V^t J_{t+1}*\|_\infty + \|J^*_t - J_t\|_\infty \\ &\leq \|J_{t+1} - J_{t+1}*\|_\infty + \|J^*_t - J_t\|_\infty,\end{aligned}$$

due to the non-expansive nature of V^t.

Thus we have that,

$$\begin{aligned}\|J_0^{SALP} - J_0^*\|_{1,\nu} &\leq \mathsf{E}_{\mathcal{S}_0\sim\nu,\mu^*}\left[J_0(\mathcal{S}_0) + 2\sum_{t=0}^{T}(TJ_{t+1}(\mathcal{S}_t) - J_t(\mathcal{S}_t))^+\right] - \mathsf{E}[J_0^*(\mathcal{S}_0)] \\ &\leq 3\|J_0 - J_0*\|_\infty + 4\sum_{t=1}^{T-1}\|J_t - J_t\|_\infty\end{aligned}$$

The result then follows after taking the infimum over each J^t, since this choice is arbitrary over all $\tilde{\mathcal{J}}$. ■

Appendix C

Optimal A-B Testing

C.1 Derivation of the Optimization Problem

Here derive the expression for efficiency used in Section **??**. Denote the matrix $X \triangleq [x\ Z]$ and $\beta \triangleq [\theta\ \kappa^\top]^\top$. Thus our model is

$$y = X\beta + \epsilon.$$

The least squares estimate $\hat{\beta}$ of β is given by

$$\hat{\beta} = (X^\top X)^{-1}X^\top y = (X^\top X)^{-1}X^\top(X\beta + \epsilon) = \beta + (X^\top X)^{-1}X^\top\epsilon.$$

Then,

$$\mathrm{Var}(\hat{\beta}) = (X^\top X)^{-1}X^\top \mathrm{Var}(\epsilon\epsilon^\top)X(X^\top X)^{-1} = \sigma^2(X^\top X)^{-1}.$$

Thus variance of $\hat{\theta} = \hat{\beta}_1$ is

$$\mathrm{Var}(\hat{\theta}) = \sigma^2 e_1^\top (X^\top X)^{-1} e_1 = \sigma^2 e_1^\top \begin{bmatrix} x^\top x & x^\top Z \\ Z^\top x & Z^\top Z \end{bmatrix}^{-1} e_1 = \frac{\sigma^2}{x^\top(I - Z(Z^\top Z)^{-1}Z^\top)x}.$$

Here, $e_1 \triangleq (1, 0, \ldots)$ is the first coordinate vector, and for the last equality we apply the block matrix inversion formula.

C.2 Performance of the Randomized Algorithm

We begin with a lemma that relies on some linear algebra arguments.

Lemma 7. *Consider a vector $a \in \mathbb{R}^{p-1}$ and an invertible $Q \in \mathbb{R}^{p-1 \times p-1}$ such that the matrix,*

$$\begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix},$$

is invertible. Then,

$$\begin{bmatrix} 1 & a^\top \end{bmatrix} \begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ a \end{bmatrix} = 1.$$

Proof. By matrix inversion lemma,

$$\begin{aligned} \begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} &= \begin{bmatrix} \frac{1}{\rho} & \frac{-a^\top Q^{-1}}{\rho} \\ \frac{-Q^{-1}a}{\rho} & (Q - aa^\top)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\rho} & \frac{-a^\top Q^{-1}}{\rho} \\ \frac{-Q^{-1}a}{\rho} & Q^{-1} + \frac{Q^{-1}aa^\top Q^{-1}}{1+a^\top Q^{-1}a} \end{bmatrix}, \end{aligned}$$

where,

$$\rho \triangleq 1 - a^\top Q^{-1} a.$$

Thus,

$$\begin{aligned} \begin{bmatrix} 1 & a^\top \end{bmatrix} \begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ a \end{bmatrix} &= \begin{bmatrix} 1 & a^\top \end{bmatrix} \begin{bmatrix} \frac{1}{\rho} & \frac{-a^\top Q^{-1}}{\rho} \\ \frac{-Q^{-1}a}{\rho} & Q^{-1} + \frac{Q^{-1}aa^\top Q^{-1}}{\rho} \end{bmatrix} \begin{bmatrix} 1 \\ a \end{bmatrix} \\ &= \frac{1}{\rho} - 2\frac{a^\top Q^{-1}a}{\rho} + a^\top Q^{-1}a + \frac{\left(a^\top Q^{-1}a\right)^2}{\rho} \\ &= \frac{1}{\rho} - 2\frac{1-\rho}{\rho} + 1 - \rho + \frac{(1-\rho)^2}{\rho} \\ &= \frac{1-2(1-\rho)+(1-\rho)\rho+1+\rho^2-2\rho}{\rho} \\ &= 1. \end{aligned}$$

■

Now we turn our attention to quantifying the performance of the randomized design.

Proof of Theorem 7. Let X_{rand} be the allocation vector of the randomized allocation. Let $S \subset \{1, \ldots, n\}$ a random subset such that $S = \{i \leq n | X_{\text{rand},i} = 1\}$.

The efficiency of the estimator of θ is,

$$\begin{aligned} \text{Var}\left(\hat{\theta}_{X_{\text{rand}}}\right)^{-1} &\triangleq \sigma^{-2} X_{\text{rand}}^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) X_{\text{rand}} \\ &= \sigma^{-2} \left(n - X_{\text{rand}}^\top Z(Z^\top Z)^{-1} Z^\top X_{\text{rand}}\right). \end{aligned}$$

Let us define,

$$\bar{Z_S} = \frac{2}{n}\sum_{k\in S} Z_k,$$

and,

$$\bar{Z_{S^c}} = \frac{2}{n}\sum_{k\notin S} Z_k,$$

the mean covariates in S and S^c respectively. Also let,

$$\bar{V} \triangleq \frac{1}{n}\sum_{1\le i\le n} Z_i Z_i^\top,$$

and

$$\bar{Z} \triangleq \frac{1}{n}\sum_{1\le i\le n} Z_i.$$

Thus,

$$Z^\top Z = n\bar{V}.$$

We have that,

$$\begin{aligned}
\mathrm{Var}\left(\hat{\theta}_{X_{\mathrm{rand}}}\right)^{-1} &= \sigma^{-2}\left(n - X_{\mathrm{rand}}^\top Z(Z^\top Z)Z^\top X_{\mathrm{rand}}\right)\\
&= \sigma^{-2}\left(n - \left(\sum_{i\in S} Z_i - \sum_{i\notin S} Z_i\right)^\top (n\bar{V})^{-1}\left(\sum_{i\in S} Z_i - \sum_{i\notin S} Z_i)\right)\right)\\
&= \sigma^{-2}\left(n - \left(\tfrac{n}{2}(\bar{Z_S} - \bar{Z_{S^c}})\right)^\top (n\bar{V})^{-1}\left(\tfrac{n}{2}(\bar{Z_S} - \bar{Z_{S^c}})\right)\right)\\
&= \tfrac{n}{\sigma^2}\left(1 - \tfrac{1}{4}\left(\bar{Z_S} - \bar{Z_{S^c}}\right)^\top \bar{V}^{-1}\left(\bar{Z_S} - \bar{Z_{S^c}}\right)\right)
\end{aligned}$$

Now note that,

$$\bar{Z_S} - \bar{Z_{S^c}} = 2(\bar{Z_S} - \bar{Z}),$$

thus,

$$\begin{aligned}
\mathrm{Var}\left(\hat{\theta}_{X_{\mathrm{rand}}}\right)^{-1} &= \tfrac{n}{\sigma^2}\left(1 - \left(\bar{Z_S} - \bar{Z}\right)^\top \bar{V}^{-1}\left(\bar{Z_S} - \bar{Z}\right)\right)\\
&= \frac{n}{\sigma^2}\left(1 - \left(\frac{2}{n}\sum_{k\in S}(Z_k - \bar{Z})\right)^\top \bar{V}^{-1}\left(\frac{2}{n}\sum_{k\in S}(Z_k - \bar{Z})\right)\right)\\
&= \tfrac{n}{\sigma^2}\left(1 - \tfrac{4}{n^2}\sum_{k,l\in S}(Z_k - \bar{Z})^\top \bar{V^{-1}}(Z_l - \bar{Z})\right)\\
&= \tfrac{n}{\sigma^2}\left(1 - \tfrac{4}{n^2}\sum_{k,l=1}^{n}(Z_k - \bar{Z})^\top \bar{V}^{-1}(Z_l - \bar{Z})\mathbf{1}_{k,l\in S}\right)
\end{aligned}$$

Thus,

$$\begin{aligned}\mathsf{E}\left[\mathrm{Var}\left(\hat{\theta}_{X_{\mathrm{rand}}}\right)^{-1}\right] &= \tfrac{n}{\sigma^2}\left(1-\tfrac{4}{n^2}\mathsf{E}\left[\sum_{k,l=1}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_l-\bar{Z})\mathbf{1}_{k,l\in S}\right]\right)\\ &= \tfrac{n}{\sigma^2}\Bigg(1-\tfrac{4}{n^2}\Bigg[\sum_{k=1}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_k-\bar{Z})\mathsf{P}(k\in S)\\ &\qquad+\sum_{k=1}^{n}\sum_{l=1,l\neq k}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_l-\bar{Z})\mathsf{P}(k,l\in S)\Bigg]\Bigg)\end{aligned}$$

But we observe that,

$$\sum_{\substack{l=1\\l\neq k}}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_l-\bar{Z}) = -(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_k-\bar{Z}).$$

Also,

$$\mathsf{P}(k\in S)=\frac{1}{2},$$

and,

$$\mathsf{P}(k,l\in S)=\frac{n/2-1}{2(n-1)}=\frac{n-2}{4(n-1)}.$$

Let us define,

$$\bar{C}\triangleq\frac{1}{n}\sum_{1\leq i\leq n}(Z_i-\bar{Z})(Z_i-\bar{Z})^\top.$$

Substituting in the earlier equation we get,

$$\begin{aligned}\mathrm{Var}\left(\hat{\theta}_{X_{\mathrm{rand}}}\right)^{-1} &= \tfrac{n}{\sigma^2}\left(1-\tfrac{4}{n^2}\left[\left(\tfrac{1}{2}-\tfrac{n-2}{4(n-1)}\right)\sum_{k=1}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_k-\bar{Z})\right]\right)\\ &= \tfrac{n}{\sigma^2}\left(1-\tfrac{1}{n(n-1)}\left[\sum_{k=1}^{n}(Z_k-\bar{Z})^\top\bar{V}^{-1}(Z_k-\bar{Z})\right]\right)\\ &= \tfrac{n}{\sigma^2}\left(1-\tfrac{1}{n(n-1)}\left[\sum_{k=1}^{n}\mathrm{tr}(\bar{V}^{-1}(Z_k-\bar{Z})(Z_k-\bar{Z})^\top)\right]\right)\\ &= \tfrac{n}{\sigma^2}\left(1-\tfrac{\mathrm{tr}(\bar{V}^{-1}\bar{C})}{n-1}\right)\\ &= \tfrac{n}{\sigma^2}\left(1-\tfrac{\mathrm{tr}(\bar{V}^{-1}(\bar{V}-\bar{Z}\bar{Z}^\top))}{n-1}\right)\\ &= \tfrac{n}{\sigma^2}\left(1-\tfrac{p-\bar{Z}^\top\bar{V}^{-1}\bar{Z}}{n-1}\right)\end{aligned}$$

Finally using Lemma 7, we get that

$$\bar{Z}^\top\bar{V}^{-1}\bar{Z}=1.$$

Thus we obtain the result. ■

C.3 Asymptotic Performance of the Optimal Design

In this section, we will prove Theorem 8. The theorem relies on Assumption 4 with $\Sigma = \rho^2 I$. In particular, we assume that $Z_{i1} = 1$ and $Z_{ij} \sim N(0, \rho^2)$ for $j > 1$. Further it is assumed that all entries of Z are independent.

We will place a sequence of problems of dimensions $1 \leq p < n$ on the same probability space $(\Omega, \mathcal{F}, \mathbf{P})$. To make the dependence on the dimension clear, we will denote the data matrix by $Z^{n,p}$. In this sequence of data matrices, $Z^{n,p}$ is formed by adding a column to $Z^{n,p}$. The additional column has the distribution $N(0, \rho^2 I_n)$. Let $\{Z^{n,n-1}\}_n$ be an independent sequence. Note that the sequence of matrices $\{Z^{n,p_n}\}_n$ defined using this generative model satisfy the assumptions laid out in Theorem 8.

Before we proceed let us set up some notation. Let $\mathrm{Gr}(k, \mathbb{R}^n)$ be Grassmannian of dimension k in the vector space $\mathbb{R}^n$. In other words, it is the set of all subspaces of dimension k in $\mathbb{R}^n$. Let $\mathcal{S}^{n,p} \in \bigcup_{k=n-p}^{n} \mathrm{Gr}(k, \mathbb{R}^n)$ be the null space of $Z^{n,p\top}$. In other words, it the the orthogonal complement of the span of $Z^{n,p}$. In the following Lemma we show that the $Z^{n,p}$ is full rank.

Lemma 8. *The rank of $Z^{n,p}$ is p with probability 1. Thus, $\mathcal{S}^{n,p} \in Gr(n-p, \mathbb{R}^n)$ almost surely.*

Proof. We can prove this inductively. Since $Z^{n,1} = \mathbf{1}$, the statement is trivially true for $p = 1$. Assume that $Z^{n,p-1}$ is rank $p-1$. It implies that the span of $Z^{n,p-1}$ is a $p-1$ dimensional subspace, let us call it $\mathrm{span}(Z^{n,p-1})$. The pth column of $Z^{n,p}$ is non-degenerate Gaussian vector independent of $\mathrm{span}(Z^{n,p-1})$, call it $Z^{,p}$. $\mathsf{P}(Z^{,p} \in \mathrm{span}(Z^{n,p-1}) = 0$. Thus, almost surely, $Z^{n,p}$ is of rank p. ■

From the preceding lemma we can conclude that $\mathcal{S}^{n,n-1}$ is a 1 dimensional subspace, with probability 1. Now we derive an expression for the efficiency of the optimal estimator for $p = n-1$ in terms of $\mathcal{S}^{n,n-1}$. Let $A = \{\omega \in \Omega : \mathcal{S}^{n-1}(\omega) \in \mathrm{Gr}(1, \mathbb{R}^n)\}$. From now on, we assume $\Omega = A$ and all subsequent statements hold with probability one.

Consider a function $h : \mathrm{Gr}(1, \mathbb{R}^n) \to \mathbb{R}_+$, such that $h(\mathcal{S}) \triangleq \frac{\|y\|_1}{\|y\|_2}$ for some non-zero $y \in \mathcal{S}$. It is trivial to check that this value is unique for any non-zero y in $\mathcal{S} \in \mathrm{Gr}(1, \mathbb{R}^n)$.

Lemma 9. *Then efficiency of the optimal estimator for $p = n-1$ is given by $\frac{h(\mathcal{S}^{n,n-1})^2}{\sigma^2}$, almost surely.*

Proof. We know that the optimal efficiency for $n = p-1$ is given by $\frac{x^{*\top} P_{Z^{n,n-1\perp}} x^*}{\sigma^2}$, where x^* is the assignment that maximizes (P1). Now not that, $P_{Z^{n,n-1\perp}}$ can be given by $\frac{yy^\top}{\|y\|_2^2}$, for any non-zero $y \in \mathcal{S}^{n-1}$. Thus the optimization problem (P1) is,

$$\begin{aligned} \text{maximize} \quad & x^\top \tfrac{yy^\top}{\|y\|_2^2} x = \tfrac{(x^\top y)2}{\|y\|_2^2} \\ \text{subject to} \quad & x \in \{-1,+1\}^n. \end{aligned}$$

But the optimal x is such that $x_i = sgn(y_i^n)$. With this assignment, the optimal value is $\frac{\|y\|_1^2}{\|y\|_2^2}$. Thus the optimal efficiency for a given ω is given by $\frac{\|y\|_1^2}{\|y\|_2^2\sigma^2} = \frac{h(\mathcal{S}^{n,n-1})^2}{\sigma^2}$. Thus,

$$\text{Eff}_*^{n,n-1} = \frac{h(\mathcal{S}^{n,n-1})^2}{\sigma^2},$$

almost surely. ■

Using the fact that we have all the $Z^{n,p}$s on the same probability space, it is easy to show that the efficiency monotonically decreases as p grows, for a fixed n.

Lemma 10. *For a fixed n, $\text{Eff}_*^{n,p}$ is a decreasing sequence in p. Thus,*

$$\inf_{1\le p<n} \frac{\text{Eff}_*^{n,p}}{n} = \frac{\text{Eff}_*^{n,n-1}}{n}$$

Proof. We will prove that $\text{Eff}_*^{n,p}(\omega)$ is a decreasing sequence in p for a fixed n. By construction, $\mathcal{S}^{n,p}(\omega) \subset \mathcal{S}^{n,p-1}(\omega)$. Note that objective value of (P1) can be written as $x^\top P_{\mathcal{S}^{n,p}} x$, where $P_{\mathcal{S}^{n,p}}$ is the projection matrix for the subspace $\mathcal{S}^{n,p}$. For each $x \in \{-1,1\}^n$ in the constraint set this value will monotonically decrease in p. Thus the optimal value will also decrease with p. This proves that $\text{Eff}_*^{n,p}$ is monotonically decreasing in p. ■

In the light of the preceding lemma we have that,

$$\inf_{1\le p<n} \frac{\text{Eff}_*^{n,p}}{n} = \frac{\text{Eff}_*^{n,n-1}}{n} = \frac{h(\mathcal{S}^{n,n-1})^2}{n\sigma^2}. \tag{C.1}$$

In the last step we find the distribution of $\mathcal{S}^{n,n-1}$. For this purpose let us setup some more notation. Let $\mathcal{Q}^1 \subset \mathbb{R}^{n\times n}$ be the group of orthonormal matrices that leaves $\mathbf{1}$ invariant. In other words, it is a collection of matrices $Q \in \mathbb{R}^{n\times n}$ that satisfy,

$$QQ^\top = Q^\top Q = I,$$

and

$$Q\mathbf{1} = Q^\top \mathbf{1} = \mathbf{1}.$$

For any $\mathcal{S} \in \mathrm{Gr}(k, \mathbb{R}^n)$, let $Q\mathcal{S} = \{Qx | x \in \mathcal{S}\}$. Let us also define $\mathcal{G}^1 = \{g \in \mathrm{Gr}(1, \mathbb{R}^n) | \mathbf{1}^\top P_g \mathbf{1} = 0\}$.

Lemma 11. *$Q\mathcal{S}^{n,n-1}$ is distributed as $\mathcal{S}^{n,n-1}$, for any $Q \in \mathcal{Q}^1$. There is a unique distribution on $\mathcal{G}^1$ that has this invariance property. Further it has the same distribution as span$(\eta^n - \mathbf{1}\bar{\eta^n})$ with $\eta^n \sim N(0, I_n)$ and $\bar{\eta^n} = \frac{1}{n}\mathbf{1}^\top \eta^n$. Finally $h(\mathcal{S}^{n,n-1})$ has the same distribution as $\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1}{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2}$*

Proof. We first show that there is a unique probability distribution on $\mathcal{G}^1$, say μ, such that $\mathcal{S}$ has the same distribution as $Q\mathcal{S}$ for any $Q \in \mathcal{Q}^1$, if $\mathcal{S}$ is distributed as μ. For this purpose we use Theorem 4.1 of [James, 1954]. $\mathcal{Q}^1$ is a transitive compact topological transformations of $\mathcal{G}^1$ to itself. Thus by the mentioned theorem, there exists a unique measure that is invariant under transformations by $Q \in \mathcal{Q}^1$.

Now we prove that span$(\eta^n - \mathbf{1}\bar{\eta^n})$ has the specified invariance property. First note that the covariance matrix of $\eta^n - \mathbf{1}\bar{\eta^n}$ is of the form $cI + d\mathbf{1}\mathbf{1}^\top$ for some $c, d \in \mathbb{R}$. Thus the covariance matrix of $Q(\eta^n - \frac{1}{n}\mathbf{1}^\top \eta^n)$ is $Q(cI + d\mathbf{1}\mathbf{1}^\top)Q^\top = cI + d\mathbf{1}\mathbf{1}^\top$. Since both of them are mean 0 and the same covariance matrix, span$(\eta^n - \mathbf{1}\bar{\eta^n})$ and span$(Q(\eta^n - \mathbf{1}\bar{\eta^n}))$ have the same distribution.

By the uniqueness of this distribution μ, we have that span$(\eta^n - \mathbf{1}\bar{\eta^n})$ is indeed distributed as $\mathcal{S}^{n,n-1}$. ■

The previous lemma explicitly gives the distribution of $h(\mathcal{S}^{n,n-1})$. Using this, we prove an asymptotic property about $\frac{h(\mathcal{S}^{n,n-1})^2}{n}$.

Lemma 12.

$$\frac{h(\mathcal{S}^{n,n-1})^2}{n} \to \frac{1}{8\pi},$$

in distribution.

Proof. From Lemma 11 we have that, $h(\mathcal{S}^{n,n-1})^2$ has the same distribution as $\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1^2}{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2}$, with

$\eta^n \sim N(0, I_n)$. Further,

$$\begin{aligned}
\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2}{n} &= \frac{1}{n}\sum_{i=1}^{n}(\eta_i^n - \bar{\eta^n})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}({\eta^n}_i^2 - 2\eta_i^n\bar{\eta^n} + \bar{\eta^n}^2) \\
&= \frac{1}{n}\sum_{i=1}^{n}{\eta^n}_i^2 - \frac{2}{n}\sum_{i=1}^{n}\eta_i^n\bar{\eta^n} + \bar{\eta^n}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}{\eta^n}_i^2 - \bar{\eta^n}^2
\end{aligned}$$

By strong law of large numbers we have that,

$$\frac{1}{n}\sum_{i=1}^{n}{\eta^n}_i^2 \to 1, \quad \text{a.s.}$$

and,

$$\bar{\eta^n}^2 \to 0, \quad \text{a.s.}$$

Thus,

$$\frac{1}{\sqrt{n}}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2 \to 1, \quad a.s. \tag{C.2}$$

Now we look at $\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1$. By triangle inequality,

$$\frac{1}{n}\|\eta^n\| + \frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1 \geq \frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1 \geq \frac{1}{n}\|\eta^n\|_1 - \frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1$$

Now by the strong law of large numbers,

$$\frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1 = |\bar{\eta^n}| \to 0, \quad \text{a.s.}$$

Thus, $\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1$ and $\frac{1}{n}\|\eta^n\|_1$ must have the same limit (if it exists). Again by, strong law of large numbers that,

$$\frac{1}{n}\sum_{i=1}^{n}|\eta_i^n| \to \mathsf{E}|\xi| = \frac{1}{2\sqrt{2\pi}},$$

where ξ is standard normal. Thus,

$$\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1 \to \frac{1}{2\sqrt{2\pi}}. \tag{C.3}$$

From (C.2) and (C.3) and using Slutsky's lemma we have that,

$$\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1}{\sqrt{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2} \to \frac{1}{2\sqrt{2\pi}}, \quad \text{a.s.}$$

By continuity of $x \mapsto x^2$,

$$\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1^2}{n\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2} \to \frac{1}{8\pi}, \quad \text{a.s.} \tag{C.4}$$

Finally by Equation (C.4) and the fact that $h(\mathcal{S}^{n,n-1})^2$ has the same distribution as $\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1^2}{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2}$,

$$\frac{h(\mathcal{S}^{n,n-1})^2}{n} \to \frac{1}{8\pi},$$

in distribution. ■

Proof of Theorem 8. Using Lemmas 9 and 10, we have,

$$\frac{\text{Eff}_*^{n,p_n}}{n} \geq \frac{\text{Eff}_*^{n,n-1}}{n} = \frac{h(\mathcal{S}^{n-1})^2}{n\sigma^2}.$$

Finally using Lemma 12 we have,

$$\frac{h(\mathcal{S}^{n-1})^2}{n\sigma^2} \to_D \frac{1}{8\pi\sigma^2}.$$

Thus for any $\epsilon > 0$

$$\mathsf{P}\left(\left|\frac{\text{Eff}_*^{n,n-1}}{n} - \frac{1}{8\pi\sigma^2}\right| > \epsilon\right) \to 0.$$

Subsequently,

$$\mathsf{P}\left(\frac{\text{Eff}_*^{n,n-1}}{n} - \frac{1}{8\pi\sigma^2} < \epsilon\right) \to 0.$$

Finally,

$$\mathsf{P}\left(\frac{\text{Eff}_*^{n,p_n}}{n} - \frac{1}{8\pi\sigma^2} < \epsilon\right) \to 0.$$

■

C.4 Dynamic Programming Formulation

Proof of Proposition 7. Consider the following n step Markov decision process (MDP):

1. The state at time k, $S_k = (\delta_{k-1}, \Delta_{k-1}, Z_k)$. The terminal state $S_n = (\delta_n, \Delta_n)$. The state space is $\mathcal{X}_k = \mathbb{R}^{2p}$ for non terminal time periods and is $\mathcal{X}_n = \mathbb{R}^p$ for the terminal time period.

2. The set of actions available to us is $\{-1, +1\}$.

3. At state S_k if action a_k is chosen, the state S_{k+1} is given by $(\delta_{k-1} + a_k, \Delta_{k-1} + a_k Z_{k,2:p}, Z_{k+1})$. After n actions, the terminal state is $S_{n+1} = (\delta_n, \Delta_n)$.

4. There is no per step reward and the terminal reward is $S_{n+1} \mapsto \delta_n^2 + \|\Delta_n\|^2_{\Sigma^{-1}}$.

Note that the MDP is finite horizon and the set of actions available at any point of time is finite, in particular 2. The problem (P3′) is just a terminal cost minimization MDP. It follows from Proposition 4.2.1 in [Bertsekas, 2013] that a policy x^* that achieves the minimum expected cost. Further there exists a set of functions $J^{*,k} : \mathcal{X}_k \to \mathbb{R}$ such that $J^{*,k}(s_k)$ is the cost conditioned on $S_k = s_k$. Trivially,

$$J^{*,n+1}(\delta_n, \Delta_n) = \delta_n^2 + \|\Delta_n\|^2_{\Sigma^{-1}}.$$

These functions follow the recursion,

$$J^{*,k}(\delta_{k-1}, \Delta_{k-1}, Z_k) = \min_{u \in \{-1,+1\}} \mathsf{E}[J^{*,k+1}(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p}, Z_{k+1})]. \tag{C.5}$$

Further x_k^*, the optimal policy, has the property that,

$$x_k^* \in \operatorname*{argmin}_{u \in \{-1,+1\}} \mathsf{E}[J^{*,k+1}(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p}, Z_{k+1})]. \tag{C.6}$$

Let,

$$Q^k(\delta_k, \Delta_k) \triangleq \mathsf{E}\left[J^{*,k+1}(\delta_{k-1}, \Delta_k, Z_{k+1})\right]. \tag{C.7}$$

Using (C.5) and (C.7),

$$Q^k(\delta_k, \Delta_k) = \mathsf{E}\left[\min_{u \in \{-1,+1\}} \mathsf{E}[Q^{k+1}(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p})\right].$$

Further using (C.6) and (C.7),

$$x_k^* \in \operatorname*{argmin}_{u \in \{-1,+1\}} \mathsf{E}\left[Q^{k+1}(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p})\right].$$

This proves the dynamic programming proposition. ■

C.5 Approximation Guarantee for the Surrogate Problem

We assume without loss that $\Sigma = I$ and begin by establishing a corollary to a basic theorem from the non-asymptotic analysis of random matrices. Let us denote by Γ_n the matrix $\frac{1}{n}Z^\top Z$. Then we have the following approximation result:

Lemma 13. *Provided $n \geq \frac{L}{\epsilon^2}\max(p, l\log\frac{2}{\delta})$, then with probability at least $1-\delta$, we have*

$$\|\Gamma_n - I\| \leq \epsilon$$

where L and l are universal constants.

Proof. Let $Z_i^\top$ be a generic row of Z. We first observe that for any x satisfying $\|x\|_2^2 = 1$, we have

$$\mathsf{E}\left(x^\top Z_i\right)^2 = 1$$

so that the rows of Z are isotropic. Moreover, the sub-Gaussian norm of $x^\top Z_i$ is bounded, uniformly over all x of unit norm, by a universal constant (say, K). This fact follows from a calculation identical to that in equation 5.6 of [Vershynin, 2012]. Consequently, we may apply Theorem 5.39 (specifically see equation 5.23) in [Vershynin, 2012], so that we have that with probability at least $1-2\exp(-c_K s^2)$,

$$\|\Gamma_n - I\| \leq C_K\sqrt{\frac{p}{n}} + \frac{s}{\sqrt{n}}$$

where $C_K(=C)$ and $c_K(=c)$ depend only on K, and can thus be taken as universal constants. Consequently, if $n \geq \max\left(\frac{4C^2p}{\epsilon^2}, \frac{4\log 2/\delta}{c\epsilon^2}\right)$, then we immediately have the result of the Lemma by taking $s = \sqrt{\frac{\log 2/\delta}{c}}$, $L = 4C^2$ and $l = \frac{1}{C^2 c}$. ■

Lemma 13 implies using Lemma 5.36 of [Vershynin, 2012] (or basic linear algebraic manipiulations) that

$$1-\epsilon \leq \sigma_{\min}\left(\frac{Z}{\sqrt{n}}\right) \leq \sigma_{\max}\left(\frac{Z}{\sqrt{n}}\right) \leq 1+\epsilon \tag{C.8}$$

Now, let us denote by $\hat{\mu}$ the measure over the sequence x_k induced by an optimal solution for the control problem (P3′) and let μ^* denote the measure induced by an optimal policy for our original dynamic optimization problem, (P3). We will demonstrate that an optimal solution to (P3′) is a near optimal solution to (P3). Before doing so, we establish some convenient notation: Denote

$$\bar{\Delta}_n = \begin{bmatrix} \delta_n \\ \Delta_n \end{bmatrix}$$

and recall

$$\Sigma_n \triangleq \frac{1}{n}\sum_{k=1}^{n} Z_{k,2:p} Z_{k,2:p}^\top$$

Proof. Now, (C.8) is equivalently stated as:

$$1-\epsilon \leq \sqrt{\lambda_{\min}\left(\Gamma_n\right)} \leq \sqrt{\lambda_{\max}\left(\Gamma_n\right)} \leq 1+\epsilon$$

which in turn implies that

$$\frac{1}{1+\epsilon} \leq \sqrt{\lambda_{\min}\left(\Gamma_n^{-1}\right)} \leq \sqrt{\lambda_{\max}\left(\Gamma_n^{-1}\right)} \leq \frac{1}{1-\epsilon}$$

By the Courant-Fisher theorem we consequently have that

$$\frac{\|\Delta\|_2^2}{(1+\epsilon)^2} \leq \Delta^\top \Gamma_n^{-1} \Delta \leq \frac{\|\Delta\|_2^2}{(1-\epsilon)^2} \ \forall \Delta \in \mathbb{R}^p. \tag{C.9}$$

Comparing sample paths yields:

$$\begin{aligned}\mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|_{\Gamma_n^{-1}}^2\right] &\leq \frac{\mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|_2^2\right]}{(1-\epsilon)^2} + \delta n^2 \\ &\leq \frac{\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2\right]}{(1-\epsilon)^2} + \delta n^2\end{aligned}$$

where the first inequality follows from the right hand side of (C.9) applied to each sample path followed by taking an expectation over sample paths. The additive factor of δn^2 corresponds to an upper bound on $\left\|\bar{\Delta}_n\right\|_{\Sigma_n^{-1}}^2$ on the fraction of sample paths where (C.9) does not hold. On those sample paths we use the fact that

$$\left\|\bar{\Delta}_n\right\|_{\Gamma_n^{-1}}^2 \leq n^2.$$

The second inequality follows from the optimality of $\hat{\mu}$ for (P3′). We will now show that

$$\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2\right] \leq (1+\epsilon)^2 \mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_{\Gamma_n^{-1}}^2\right] + n^2 p\delta + O\left(\sqrt{\frac{n}{p-1}}\right);$$

together with the inequality above, this will yield the theorem. To prove this inequality, we first observe (as we did earlier) that on the set where (C.9) holds, $\left\|\bar{\Delta}_n\right\|_2^2 \leq (1+\epsilon)^2 \left\|\bar{\Delta}_n\right\|_{\Gamma_n^{-1}}^2$. Further, we can upper bound the expectation of $\left\|\bar{\Delta}_n\right\|_2^2$ over the set where (C.9) does not hold by the quantity:

$$n^2\delta + \mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha(\delta)}\right]$$

where $\alpha(\delta)$ satisfies $\mathbb{P}_{\mu^*}\left(\|\Delta_n\|^2 \geq \alpha(\delta)\right) = \delta$. Applying Lemma 16 yields

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha(\delta)}\right] \leq n^2(p-1)\delta + O\left(\sqrt{\frac{n}{p-1}}\right).$$

which yields the result. ■

To complete our proof of the theorem above, we must provide an upper bound on the quantity

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|^2 \mathbf{1}_{\|\Delta_n\|^2 \geq \alpha(\delta)}\right]$$

where $\alpha(\delta)$ satisfies $\mathbb{P}_{\mu^*}\left(\|\Delta_n\|^2 \geq \alpha(\delta)\right) = \delta$. Let $\bar{Z}$ be a Gamma$(n(p-1)/2, 1)$ random variable, and let $\hat{\alpha}(\delta)$ satisfy

$$\mathsf{P}\left(\bar{Z} \geq \hat{\alpha}(\delta)\right) = \delta.$$

We have

Lemma 14.

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha(\delta)}\right] \leq 2n\mathsf{E}\left[\bar{Z}\mathbf{1}_{\bar{Z} \geq \hat{\alpha}(\delta)}\right]$$

Proof. Observe that

$$\begin{aligned}
\|\Delta_n\|_2^2 &= \left\|\sum_{k=1}^{p} x_k Z_k\right\|_2^2 \\
&\leq \left(\sum_{k=1}^{p} \|Z_k\|_2\right)^2 \\
&\leq n \sum_{k=1}^{p} \|Z_k\|_2^2.
\end{aligned}$$

where the first inequality follows from the triangle inequality and the second from Cauchy-Schwartz. We then immediately have that

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha(\delta)}\right] \leq n\mathsf{E}\left[\left(\sum_k \|Z_k\|_2^2\right) \mathbf{1}_{\sum_{k=1}^{p} \|Z_k\|_2^2 \geq \tilde{\alpha}(\delta)}\right].$$

But $\frac{1}{2}\sum_{k=1}^{p} \|Z_k\|_2^2 \triangleq \bar{Z}$ is distributed as a Gamma$(n(p-1)/2, 1)$ random variable and the claim follows. ■

Now Gamma random variables enjoy the following property on their tails:

Lemma 15. *If* $\bar{Z} \sim$ Gamma$(k, 1)$ *and* $z(\delta)$ *is its* δ*th quantile (i.e.* $z(\delta)$ *satisfies* $\mathbb{P}(\bar{Z} \geq z(\delta)) = \delta$*), then:*

$$\mathsf{E}\left[\bar{Z}\mathbf{1}_{\bar{Z} \geq z(\delta)}\right] \leq k\delta + O\left(\frac{1}{\sqrt{k}}\right)$$

Proof. We have:

$$\begin{aligned}
\mathsf{E}\left[\bar{Z}\mathbf{1}_{\bar{Z}\geq z(\delta)}\right] &= \int_{z(\delta)}^{\infty} z\frac{z^{k-1}\exp(-z)}{\Gamma(k)}dz \\
&= \frac{\Gamma(k+1)}{\Gamma(k)}\int_{z(\delta)}^{\infty}\frac{z^{k}\exp(-z)}{\Gamma(k+1)}dz \\
&= k\left[\frac{\Gamma(k+1,z(\delta))}{k\Gamma(k)}\right] \\
&= k\left[\frac{k\Gamma(k,z(\delta))+z(\delta)^{k}\exp(-z(\delta))}{k\Gamma(k)}\right] \\
&= k\left[\delta+\frac{z(\delta)^{k}\exp(-z(\delta))}{k\Gamma(k)}\right]
\end{aligned}$$

where $\Gamma(\cdot,\cdot)$ is the right incomplete Gamma function. The final equality uses the fact that

$$\frac{\Gamma(k,z(\delta))}{\Gamma(k)}=\delta$$

by the definition of $z(\delta)$. But $\frac{z^{k}\exp(-z)}{k\Gamma(k)}$ is maximized at $z=k$, so that

$$\frac{z(\delta)^{k}\exp(-z(\delta))}{k\Gamma(k)}\leq\frac{k^{k}\exp(-k)}{k\Gamma(k)}=O\left(\frac{1}{k^{3/2}}\right)$$

where we have used Stirlings approximation for $\Gamma(k)$. The result follows. ■

We anticipate that tighter control on the big-oh error term is possible in the above proof, but this level of crudeness suffices. Using the preceding two lemmas now immediately yields:

Lemma 16.

$$\mathsf{E}_{\mu^{*}}\left[\|\Delta_{n}\|^{2}\mathbf{1}_{\|\Delta_{n}\|^{2}\geq\alpha(\delta)}\right]\leq n^{2}(p-1)\delta+O\left(\sqrt{\frac{n}{p-1}}\right)$$