Decomposition Methods for Large Scale Stochastic and Robust Optimization Problems

by

Adrian Bernard Druke Becker

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2011

Author ....
Sloan School of Management
July 31, 2011

Certified by ....
Dimitris Bertsimas
Boeing Leaders for Global Operations Professor
Co-Director, Operations Research Center
Thesis Supervisor

Accepted by ....
Patrick Jaillet
Dugald C. Jackson Professor
Co-Director, Operations Research Center

Decomposition Methods for Large Scale Stochastic and Robust Optimization Problems

by

Adrian Bernard Druke Becker

Submitted to the Sloan School of Management
on July 31, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

We propose new decomposition methods for use on broad families of stochastic and robust optimization problems in order to yield tractable approaches for large-scale real world application. We introduce a new type of a Markov decision problem named the Generalized Restless Bandits Problem that encompasses a broad generalization of the restless bandit problem. For this class of stochastic optimization problems, we develop a nested policy heuristic which iteratively solves a series of sub-problems operating on smaller bandit systems. We also develop linear-optimization based bounds for the Generalized Restless Bandit problem and demonstrate promising computational performance of the nested policy heuristic on a large-scale real world application of search term selection for sponsored search advertising.

We further study the distributionally robust optimization problem with known mean, covariance and support. These optimization models are attractive in their real world applications as they require the model consumer to only rely on those statistics of uncertainty that are known with relative confidence rather than making arbitrary assumptions about the exact dynamics of the underlying distribution of uncertainty. Known to be $\mathcal{NP}$ – *hard*, current approaches invoke tractable but often weak relaxations for real-world applications. We develop a decomposition method for this family of problems which recursively derives sub-policies along projected dimensions of uncertainty and provides a sequence of bounds on the value of the derived policy. In the development of this method, we prove that non-convex quadratic optimization in n-dimensions over a box in two-dimensions is efficiently solvable. We also show that this same decomposition method yields a promising heuristic for the MAXCUT problem. We then provide promising computational results in the context of a real world fixed income portfolio optimization problem.

The decomposition methods developed in this thesis recursively derive sub-policies on projected dimensions of the master problem. These sub-policies are optimal on relaxations which admit "tight" projections of the master problem; that is, the projection of the feasible region for the relaxation is equivalent to the projection of that of master problem along the dimensions of the sub-policy. Additionally, these decomposition strategies provide a hierarchical solution structure that aids in solving large-scale problems.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Leaders for Global Operations Professor
Co-Director, Operations Research Center

Acknowledgments

I would like to thank the members of my thesis committee, Dimitris Bestsimas, Vivek Farias, and Retsef Levi for their support and valuable input.

I am especially grateful to my thesis advisor Dimitris Bertsimas. His guidance in pursuit of this Ph.D. helped me to identify my strengths and foster a passion for my work. His advice as a mentor in both life and academics will have lasting impact.

The Operations Research Center has been like a home away from home throughout these past five years. The atmosphere of collaboration and discussion was invaluable. I would particularly like to thank Vihn Doan and Alex Rikun whose insights and suggestions were instrumental in the development of this work. I would also like to thank my classmates Jason Acimovic, David Goldberg, Dan Iancu, Jonathan Kluberg, Shashi Mittal, Diana Pfeil, Andy Sun, and Gareth Williams whose comradery and debate enriched both my enjoyment and education.

I could not have come this far without the love and support of my parents and extended family. They taught me to keep an open mind and appreciate the views of those from all walks of life; and they demonstrated that no challenge is insurmountable.

Finally, I would like to thank my wife, Catie, and daughter, Sara. Their unwavering love throughout the good times and bad will never be forgotten. The smiles on their faces are the greatest accomplishments I can achieve.

Cambridge, July 2011 *Adrian Becker*

Dedication

This work is dedicated to my mother, Mary Druke Becker. Her love, perseverance, and selflessness are responsible for my best qualities and made all of my achievements possible. She believed in me even when I did not believe in myself and pushed me to always do better. I love you Mom, your memory will never fade.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sophisticated models for stochastic and distributionally robust optimization exist in the academic community, but their application in practice is often limited by the "curse of dimensionality". Exact formulations as mathematical optimization models for Markov decision problems and distributionally robust optimization problems with known mean, covariance and support information exist, but the size of these formulations are not polynomially bounded and quickly explode for moderately sized problems. Previous attempts at practical heuristics for solving these problems generally rely on policies obtained by solving relaxations with polynomially bounded size. The main contribution of this thesis is to introduce efficient and computationally promising decomposition methods for these problems.

The unified theme of the decomposition methods developed in this thesis is that they study a sequence of efficiently solvable relaxations of the master problem. Each relaxation is 'tight' when projected along specific dimensions of the master problem, that is, the projection of the relaxation along those dimensions is exactly the same as the projection of the master problem along those same "tight" dimensions. Such relaxations can be efficiently solved to optimality yielding what we call a sub-policy for the "tight" dimensions. A sub-policy sets our decisions along tight dimensions, allowing us to collapse the overall dimension of the master problem.

As an example, suppose the decisions in our master problem involve how much money, w_i, should be invested in each of a collection of assets $i = 1, ..., N$, each asset having uncertain returns r_i that lie in some space $r \in R$ which yields a difficult optimization problem. We then consider a subset of dimensions, say assets 1 and 2, and examine a relaxed space $\bar{R} \supset R$ over which the investment problem becomes tractable, such that $\pi_{1,2}(R) = \pi_{1,2}(\bar{R})$, where π_{12} is the projection operator onto the dimension r_1 and r_2. Solving the investment problem over $\bar{R}$, we obtain a sub-policy for how to relatively invest in assets 1 and 2. By fixing this sub-policy, we can replace assets 1 and 2 in the master problem with a portfolio corresponding to their relative investment, thus reducing the dimension of the master problem.

In summary, the decomposition methods studied in this thesis involve iterating the following three step process:

1. Obtain an efficiently solvable relaxation that admits a tight projection on selected dimensions. That is, the projection of the feasible region for the relaxation is equivalent to the projection of that of the original problem along these dimensions.

2. Solve the relaxed problem exactly to obtain an optimal sub-policy over the projected dimension.

3. Use the optimal sub-policy to collapse the dimension of the original problem.

The nature of the decomposition methods studied in this thesis is to iteratively solve a number of small tractable subproblems. The number of subproblems solved is typically linear in the overall dimension of the problem; thus these methods are well suited for large-scale application.

In this work, we will also introduce a broad class of Markov decision problems called Generalized Restless Bandits for which such decomposition methods perform well. We will study decomposition methods in the context of Generalized Restless Bandits as well as distributionally robust optimization with known mean, covariance and support information. In each of these regimes we will:

- Derive decomposition strategies which are efficiently solvable in practice.
- Derive sequences of efficiently solvable relaxations which can be used to bound the true optimum and evaluate performance of our heuristics a posteriori.
- Provide computational evidence of the effective performance of our heuristics compared to prior work
- Discuss application areas for these problems.
- Perform an in depth case study of the application of our methods to a real-world problem in the context of sponsored search advertising for Generalized Restless Bandits and fixed income portfolio valuation for distributionally robust optimization.

1.1 Structure of the Thesis

- **Chapter 2: Generalized Restless Bandits: Algorithms and Applications.** Traditional multi-armed bandits model a class of Markov decision processes where the state space is decomposable into the product of a number of independent sub-processes called arms. The there are two actions available in each time step for each arm "on" or "off" along with a constraint controlling how many "on" actions must be chosen in each timestep. Arms transition only when the "on" action is taken. Early work by Gittins [19, 18] and Whittle [54, 55] found that such multi-armed bandit problems are efficiently solvable by calculating an index for each state of each arm. The highest index states are played in each time period.

  Restless bandit models are an extension where arms also transition when an "off" action is taken. These problems are in general $\mathcal{P}$-space hard and current research focuses on finding classes of restless bandits that are indexable (an index policy is optimal) or on obtaining index based heuristics similar to the non-restless case.

In the academic community, bandits are often used to model research investment budgeting problems [3, 19, 31, 30, 54]. In this setting, arms can be thought of projects and "on"/"off" as invest/not invest. Real-life projects, however, often allow for multiple levels of investment rather than an all-or-nothing choice. This and other application areas motivats the definition of Generalized Restless Bandits (GRBs). GRBs are similar to traditional bandit models in that the state space is the product space of several independent subsystems; however, rather than just two modes of operation, GRBs allow for an arbitrary number of modes each with an associated cost.

We propose a decomposition method called nested policies to solve GRBs. In this approach we recursively project the full polytope onto the space of two arms and decide how a budget should be allocated optimally in the restricted two-armed problem. This optimal sub-policy is then used to combine those two arms into a single one, collapsing the number of arms in the overall problem.

In this chapter, we formally define the Generalized Restless Bandit problem, derive linear optimization based bounds and propose the Nested Policy decomposition method. We also derive an additional algorithm to solve GRBs called the generalized primal-dual method which is a generalization of existing methods to solve traditional restless bandit problems. We then demonstrate computational results that indicate the superiority of the Nested Policy method over primal-dual approaches on a wide range of problems.

- **Chapter 3: Sponsored Search Optimization Application.** In this chapter, we perform a case study of GRB's based on real world data from an online retailer of herbal suppliments. In this problem an advertiser considers a collection of keywords and phrases on which to bid. Each time a user performs a search containing selected search terms, an advertisement is displayed. The position of the advertisement and correspondingly the probability the user clicks on it are related to the amount bid on the search term. In a GRB model, each candidate search term is an arm, modes of operation are analogous to potential bid levels, and the state space represents the advertiser's learning about the response function of bid level-to-click through probability for the search term.

In the context of the resulting large-scale model, we also describe several steps that can be taken to simplify the derivation of a nested policy.

- **Chapter 4: Distributionally Robust Optimization Problems.** In this chapter, we study the distributionally robust optimization problem with known mean, covariance, and support. These models allow practitioners to incorporate known statistics about underlying distributions of uncertainty which being robust to dynamics that are unknown. These problems are known to be $\mathcal{NP}$ – *hard*. We derive a decomposition strategy for these problems which iteratively finds sub-policies along pairs of dimensions of uncertainty. In order to demonstrate an efficiently-solvable relaxation which is "tight" on projected dimensions, we prove that the Sherali-Adams [47] closure is tight for the semidefinite relaxation of n-dimensional quadratic optimization over a 2-dimensional box.

  We implement this decomposition method in the context of a real world fixed income portfolio optimization problem. Distributionally robust optimization with known mean, covariance, and support is particularly well suited for this problem since returns for fixed income investments are bounded between the face value and recovery rate upon default. Additionally, while the practitioner may have reliable statistics on individual asset default as well as probabilities of the joint default of pairs of assets, dynamics of the underlying uncertainty related to baskets of default for larger subsets of assets are relatively unknown. We also show that this same decomposition strategy yields a promising heuristic for the MAXCUT problem, providing computational results on exhaustive graphs for small problems, random graphs for moderately sized problems, and the DIMACS challenge [1] problems for large instances.

- **Chapter 5: Conclusions.** This chapter contains the concluding remarks of the thesis.

1.2 Contributions

- **Definition of the Generalized Restless Bandit Problem (GRB).** We define a generalization of multi-armed restless bandit problem with multiple actions available for each arm with each action consuming a different amount of shared budget over all arms.

- **Algorithms for solving GRBs.** We propose a decomposition method for solving GRBs called the nested policy approach. This method iteratively looks at pairs of arms and decides how to allocate various amounts of budget between them in each state by solving the corresponding two-armed subproblem exactly. Once such a sub-policy is derived for a pair, the pair is combined into a single arm, thus reducing the dimension of the overall problem. We also propose a generalization of the primal dual heuristic of Bertsimas and Niño-Mora [8] for traditional restless bandits. Superior performance of the nested policy approach is shown in computational studies as the action space grows larger. We further discuss linear optimization models which provide bounds on the true optimum for GRBs.

- **Real world application of GRB to the sponsored search advertising problem.** We model the advertiser's problem for sponsored search as a GRB. We then implement this model on a large scale real world data set and show promising performance of the nested policy approach.

- **Decomposition strategy for the distribtutionally robust optimization problem.** We study the distributionally robust optimization problem with known mean, covariance, and interval support. We propose a decomposition strategy for this $\mathcal{NP}-\boldsymbol{hard}$ problem that iteratively derives sub-policies for pairs of decision variables. The relaxations solved incorporate full information of the projected distribution of corresponding uncertain random variables for each iteration in contrast to existing approaches which relax distributional information for the entire problem.

- **Non-convex quadratic optimization in n-dimensions over a box in two-**

dimensions in polynomially solvable. We prove that non-convex quadratic optimization in n-dimensions over a box in two-dimensions is efficiently solvable. Our proof is a generalization of a result by Anstreicher and Burer [2] for $n = 2$ using simpler proof technology. This result ensures the tight nature of our projected relaxations in the decomposition approach for the distributionally robust optimization problem.

- **Real world application to the fixed income portfolio optimization problem.** We model the fixed income portfolio optimization problem in the context of distributionally robust optimization with known mean, covariance, and support. This model is then implemented on a real world data set showing promising computational results.

- **Decomposition approach for MAXCUT.** We demonstrate the applicability of the decomposition strategy for distributionally robust optimization to the MAXCUT problem. The resulting decomposition approach for MAXCUT shows promising computational results when compared to the well known approach of Goemans-Williamson rounding [24] on random graphs and performs well on large scale problems from the DIMACS competition [1].

Chapter 2

Generalized Restless Bandits: Algorithms and Applications

Government institutions such as the National Science Foundation, drug companies, and industrial conglomerates evaluating competing emerging technologies are faced with complex multi-year research investment decisions with highly uncertain outcomes. Restless multi-armed bandit models have been used in previous attempts to solve these problems. Such models allow us to evaluate yes or no decisions on which collection of projects to fund, but fall short of allowing us to consider various degrees of investment in each individual project. We introduce a generalization of restless multi-armed bandits to address this issue. The resulting class of Generalized Restless Bandit processes is sufficiently broad to address numerous applications in investment management, inventory management, scheduling, and statistical sampling. We develop a Nested Policy heuristic to solve the Generalized Restless Bandit problem by recursively decomposing a bandit system and exploiting our ability to efficiently solve a two-armed system exactly. We also provide linear optimization models which can be used to bound the true optimum and thus measure the degree of sub-optimality of a Nested Policy. To keep our proposed decomposition method tractable, we employ a novel state-clustering technique based on solving integer optimization models. We present promising computational results in both generalized and classical settings.

2.1 Introduction

The multi-period research investment problem is ubiquitous in industrial and institutional research. The National Science Foundation (NSF) must decide each year from amongst numerous applications which projects to support on an ongoing basis. Pharmaceutical companies must decide which drug lines to continue or initiate in various stages of maturity. Large corporations such as General Electric must decide which competing emerging technologies to pursue, such as competing approaches to solar energy. Multi-armed bandit models and restless multi-armed bandit models have been widely employed to solve such problems as seen in Asawa and Teneketzis [3], Gittins and Jones [19], Kavadias and Loch [31], Kavadias and Chao [30], and Whittle [54]. Bandit models are a special case of Markov decision processes which are attractive in that each arm or project evolves independently over time. This gives a natural decomposition of the overall state space of the bandit process as the product space of the states of each arm. In these models, in each time period, several projects are chosen to be pursued. Each pursued project incurs a reward and transitions to another state.

Restless bandits are a variant in which non-pursued projects also transition to a new state according to a different transition probability matrix. In these models, however, projects must be either fully committed to or abandoned in each time period. Attempted extensions beyond full commitment or abandonment in bandit processes have been limited to rewards linear in the degree of commitment by Niculescu-Mizil [35].

In this chapter, we address a further generalization called Generalized Restless Bandit problems (GRBs). In a GRB each project may be pursued to any fixed number of degrees in each time period under a total fixed budget rather than simply being set "on" or "off". Additionally, each project can produce non-linear payoffs in the degree of commitment and change states according to separate transition matrices for each degree of commitment. This allows us to preserve the appealing arm-independence property of bandit processes while addressing a much richer class of applications since we are not limiting our action space to simply "on" or "off" for each arm.

2.1.1 Previous Work

Whittle [55] first studied a continuous-time version classical restless bandit under a dynamic programming framework. He introduced a relaxed version of the problem where the total discounted time average number of N arms being played was equal to some budget Whittle's relaxation could be solved to optimality in polynomial time. He proposed a priority-index heuristic which reduces to the optimal Gittins' index described by Gittins [19] and Whittle [54] in the non-restless case. This index is derived by finding the least subsidy necessary to be paid in each state to be indifferent between being active and passive. In any given time period, the arms with highest indices in their current state are played. Whittle's heuristic only applies to a restricted class of restless bandit which satisfy an *indexability* property as discussed in Whittle [55], Bertsimas and Niño-Mora [7, 8], as well as Weber and Weiss [52], which ensures that if a certain subsidy induces passivity for a state, all greater subsidies induce passivity.

Since Gittins' work, research into solving restless bandit problems has focused on identifying classes of problems that are indexable and for which index policies are optimal as in Glazebrook et al. [22, 23], Ruiz-Hernandez [44], and Bertsimas and Niño-Mora [7], and developing other index-like heuristics to solve larger families of problems as in Bertsimas and Niño-Mora [8] which are not subsidy based. Such algorithmic approaches attempt to assign a priority-index to each state of each arm and then play those arms whose current state has the highest index. Additionally, Weber and Weiss [52] established conditions under which, for indexable restless bandits index policies are asymptotically optimal; that is if a one takes a system with n identical copies of an indexable restless bandit, index policies are optimal as $n \to \infty$ if the total budget is scaled in proportion. When dealing with a larger action space for each arm, with each action consuming a different amount of the total budget, the subsidy-based index policies of Whittle do not directly apply.

Since the work of this thesis, it has been brought to the author's attention that Hodge and Glazerbrook [27] have defined a notion of subsidy-based index policies for GRBs (referred to in their work Multi-Action Restless Bandits). They define a different subsidy-based index for each action to be the amount required to cause indifference between that action and an action consuming one less unit of budget. *Indexability* in this setting refers ensuring that if a certain subsidy induces no greater than a specific level of budget consumption x for a state, no greater subsidies can induce a budget consumption greater than x in that state. They have established conditions for the asymptotic optimality of such policies for indexable GRBs and are working on identifying restrictive classes of GRBs which are indexable. While having proved effective in solving classical restless bandit problems, this notion of a state-based indexing policy is much more restrictive in the general case which we introduce, where arms can be played to varying degrees rather than simply "on" or "off".

2.1.2 Applications

In addition to the research budgeting problem, the generalized restless bandit problem allows us to model many applications of classical restless bandit models within a richer framework as illustrated by the following examples:

- **Modern Slot Machines.** The archetypal example of a multi-armed bandit problem is when a player sits in front of a collection of N slot machines. In each time period, the player with M coins may pick M machines to play, putting a coin in those machines and pulling their arms. Modern slot machines, however, offer the ability to play with various amounts of money for a single pull. For example, if the player inserts a single coin into a machine he or she may win if three of the same symbol appear in a horizontal line after the pull; but if the player inserts two coins into the machine he or she may win if those three symbols appear in either a horizontal of diagonal line after the pull. The GRB problem can model a player sitting in front of a collection of N such modern slot machines with a budget of M coins in each period and the ability to put multiple coins into a single machine for a single pull, offering differing payouts and possibly different transitions.

- **Multi-target sensor networks.** In these problems, arms represent N moving targets which can be tracked using M radars which can be electronically steered by a central controller. The objective is to derive a dynamic schedule of track updates which minimize the variance in tracking error. Niño-Mora and Villar [36] and Washburn and Schneider [51] model this problem as a classical restless multi-armed bandit problem and derive a near optimal schedule that decides which targets to track by individual radar in each time period. In the GRB setting we are allowed to point multiple radars at a single target, giving us the ability to track high-value targets with increased precision.

- **Keyword selection in search-based advertising.** Search-based advertising involves a customer bidding a certain amount on a search keyword or phase. Whenever this keyword is included in an online search, the customer's advertizement is presented to the search user in a screen position determined by the bid amount. Rusmevichientong and Williamson [45] discuss the possibility of using a multi-armed bandit model in the keyword selection problem but point out that in such classical models the bid price of each keyword must remain constant. A GRB model allows us to incorporate varying bid amounts for keywords and simultaneously solve the keyword selection and bid level setting problems.

- **Multi-task worker scheduling.** Whittle [55] and Bertsimas and Niño-Mora [8] discuss the use of multi-armed bandits in the worker scheduling problem. In each time period, M employees out of a pool of N must be assigned to work. Work-assignment changes the state of each worker resulting in temporary exhaustion and recuperation. The GRB problem allows us to extend this model to the assignment of workers to multiple tasks with different levels of exhaustion.

- **Make-to-stock production facility control.** A make-to-stock production facility can be used to produce N different product classes. Each finished product experiences exogenous demand and the objective is to minimize lost-sales. Veatch and Wein [49] propose a classical restless bandit formulation of this problem where arm states represent class inventory levels and the facility is assigned production of a specific product in each period. GRB models allow us to increase our level of control over the production facility. Rather than simply turn production of a specific product "on" or "off" in each period, we can model the ability to produce any level of product mix.

- **Clinical trials.** In these problems, arms represent varying medical treatments with states representing one's knowledge of treatment efficacy. Whittle [55] and Bertsimas and Niño-Mora [8] discuss the use of multi-armed bandits to decide which collection of treatments to test in each time period with a clinical trial. In the GRB setting, rather than having one type of clinical trial available for each treatment, we can explore the possibility of various trials of differing cost.

2.1.3 Contributions

Our contributions include:

1. We introduce a generalization of restless multi-armed bandit problem called the Generalized Restless Bandit (GRB) problem that allows for more modeling power, compared to restless bandits. Rather than having two modes of operation, "on" or "off", each arm can operate in several modes of varying cost, with the total budget in each time period being fixed. Each degree of operation has its own associated reward and transition probability matrix for each state of each arm.

2. We extend work by Bertsimas and Niño-Mora [8] on classical restless bandits to derive a hierarchy of linear optimization based relaxations for the GRB problem. These relaxations not only provide increasingly tight bounds on the optimal solution to a GRB, but also are useful in constructing the Nested Policy heuristic described next. We also discuss a generalization of the primal-dual index method used by Bertsimas and Niño-Mora [8] on classical restless bandits and use this approach to benchmark our Nested Policy heuristic.

3. We propose a Nested Policy heuristic to solve the GRB problem based on our ability to optimally solve the two-armed GRB problem in an efficient manner. We show that if we consider pairs of arms in the original GRB problem, the resulting two-armed GRB problem can be solved exactly in polynomial time through linear optimization. To progress to the multi-armed case, the key insight is that the dynamics of a two-armed GRB process under a fixed allocation policy between the two arms can itself be viewed as an arm of a larger GRB process. Thus, by implementing the optimal policy for two independent arms, we can combine these two arms into one that operates on a larger state space. We use a mixed-integer optimization based clustering technique to shrink the state-space of this resulting arm, thus iteratively decreasing the number of arms in our problem without expanding the state space.

4. We present promising computational results evaluating the performance of the Nested Policy heuristic in three regimes:

   (a) the classical non-restless multi-armed bandit setting

   (b) the classical restless multi-armed bandit setting

   (c) the general setting.

2.1.4 Structure of the chapter

In Section 2.2, we define the Generalized Restless Bandit (GRB) problem and examine the polyhedral structure of its associated performance region leading us to an exact solution method for the GRB problem and discussing its complexity. Section 2.3 provides polynomially solvable linear optimization bounds on the optimal solution. Section 2.3.1 extends the primal-dual index heuristic of Bertsimas and Niño-Mora [8] to the GRB setting. Section 2.4 describes our proposed Nested Policy approach to solving GRBs. This approach allows us to iteratively decompose a large GRB problem into smaller ones that can be solved exactly. We begin discussion of the Nested Policy Heuristic by detailing an example in Section 2.4.1. Section 2.4.2 details the general steps in construction and implementation of the policy as well as the mixed-integer optimization formulations of the Arm Aggregation and State Clustering subproblems which are solved when deriving a Nested Policy. In Section 2.5, we report promising computational results for the Nested Policy heuristic, evaluating performance in both general and classical regimes.

2.2 The GRB Problem and its polyhedral structure

Consider a system of n arms, each arm i can be played up to a maximum degree D_i in each time period, each degree $d_i = 0, ..., D_i$ representing the cost of a different mode of operation. In each time period, we must play all arms to a combined degree M representing the total budget available. Each arm is a Markov chain over a finite state space S_i. In each period, the current state $\boldsymbol{s} = \{s_i\}_{i=1}^n$ of the n arms is known, where $s_i \in S_i$ and $|S_i| = k_i$ for all $i = 1, \ldots, n$. We need to decide the degree vector $\boldsymbol{d} \in \mathbb{Z}_+^n$ to which the arms will be played, $\sum_{i=1}^{n} d_i = M$. The reward we obtain from the arm i is $R_i^{d_i}(s_i)$ for all $i = 1, \ldots, n$. Each arm i evolves independently with the transition probability $P_i^{d_i}(s_i \rightarrow s_i')$ for all $s_i, s_i' \in S_i$. We would like to find the optimal *Markovian policy* $u \in \mathcal{U}$ that optimizes the total expected discounted reward

$$\mathbf{E}_u \left[\sum_{t=0}^{\infty} \beta^t \left(\sum_{i=1}^{n} R_i^{d_i(t)}(s_i(t)) \right) \right],$$

where β is the discount factor, $0 < \beta < 1$, and $\mathcal{U}$ is the class of all *admissible* policies. We assume that the probabilities of initial states are known and described by the vector $\{\alpha_{\boldsymbol{s}}\}_{\boldsymbol{s} \in \bigotimes_{i=1}^n S_i}$.

2.2.1 The Performance Region Polyhedron

In order to formulate the problem mathematically, we adapt an approach described by Bertsimas and Niño-Mora [8]. Let us consider the total expected discounted time

$x^{\boldsymbol{d}}(\boldsymbol{s}, u)$ that the degree vector $\boldsymbol{d}$, $\sum_{i=1}^{n} d_i = M$, is chosen in state $\boldsymbol{s}$ under the policy u,

$$x^{\boldsymbol{d}}(\boldsymbol{s}, u) = \mathbb{E}_u \left[\sum_{t=0}^{\infty} \beta^t I_{\boldsymbol{s}}^{\boldsymbol{d}}(t) \right],$$

where $I_{\boldsymbol{s}}^{\boldsymbol{d}}(t)$ is the indicator whether the degree vector $\boldsymbol{d}$ is chosen in state $\boldsymbol{s}$ in period t. The problem can then be formulated as follows

$$Z^* = \max_{u \in \mathcal{U}} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{\boldsymbol{s} \in S} \left(\sum_{i=1}^{n} R_i^{d_i}(s_i) \right) x^{\boldsymbol{d}}(\boldsymbol{s}, u), \tag{2.1}$$

where $\mathcal{D} = \left\{ \boldsymbol{d} \in \mathbb{Z}_+^n : 0 \le d_i \le D_i, \sum_{i=1}^{n} d_i = M \right\}$.

Consider the *performance region* X of the vector $\boldsymbol{x}(u) = \{x^{\boldsymbol{d}}(\boldsymbol{s}, u)\}_{\boldsymbol{s} \in S, \boldsymbol{d} \in \mathcal{D}}$ for all $u \in \mathcal{U}$. For ease of presentation, we suppress u and write $\boldsymbol{x} = \{x^{\boldsymbol{d}}\}(\boldsymbol{s}), \boldsymbol{s} \in S, \boldsymbol{d} \in \mathcal{D}$. Problem (2.1) is equivalent to the following problem:

$$Z^* = \max_{\boldsymbol{x} \in X} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{\boldsymbol{s} \in S} \left(\sum_{i=1}^{n} R_i^{d_i}(s_i) \right) x^{\boldsymbol{d}}(\boldsymbol{s}). \tag{2.2}$$

Bertsimas and Niño-Mora [8] prove the following theorem for the more general case of Markov decision chains, where $\mathcal{D}$ is any discrete set and can depend on $\boldsymbol{s}$:

Theorem 2.1 (Bertsimas and Niño-Mora [8]) *The following statements hold:*

(a) $X = \mathcal{P}$, where $\mathcal{P}$ is the polyhedron defined as follows

$$\mathcal{P} = \left\{ \boldsymbol{x} \in \mathbb{R}_+^{|\mathcal{D}| \times |S|} : \sum_{\boldsymbol{d} \in \mathcal{D}} x_{\boldsymbol{s}}^{\boldsymbol{d}} = \alpha_{\boldsymbol{s}} + \beta \sum_{\boldsymbol{s}' \in S} \sum_{\boldsymbol{d} \in \mathcal{D}} \left(\prod_{i=1}^{n} P_i^{d_i}(s_i' \to s_i) \right) x_{\boldsymbol{s}'}^{\boldsymbol{d}}, \quad \forall \boldsymbol{s} \in S \right\}.$$

(b) The vertices of polyhedron $\mathcal{P}$ are achievable by stationary deterministic *policies.*

The stationary deterministic policy can be determined from the fact that for each state $\boldsymbol{s} \in S$, every extreme point of $\mathcal{P}$ has at most *one* positive value $x^{\boldsymbol{d}}(\boldsymbol{s}) > 0$ among all values $x^{\boldsymbol{d}}(\boldsymbol{s})$, $\boldsymbol{d} \in \mathcal{D}$.

As the GRB problem encompasses the traditional Restless Bandit problem, GRB is *PSPACE-hard* as shown by Papadimitriou and Tsitsiklis [38]. Additionally, this problem is not in general indexable, that is we cannot assign an index $\gamma_i(s_i)$ to each state of each arm such that the optimal policy in each time period is to play the arms with the highest index. To see non-indexability, consider the case where each arm starts in an initial dummy state and transitions deterministically to another state regardless of the degree to which it is played. This is in fact equivalent to solving a different multiple-choice knapsack problem in each time period where we must pick exactly one degree from each arm with the knapsack constraint that the total degree picked is equal to the budget.

2.3 Linear Optimization Bounds

Whittle [55] first introduced a relaxed version of the classical restless bandit problem which is polynomially solvable. In this relaxation, rather than requiring the arms to be played to a total budget of exactly M in each time period, we require the discounted time average total degree to be M. Bertsimas and Niño Mora [8] use this relaxed problem to formulate the first-order relaxation polyhedron for the classical restless bandit problem. We extend this formulation to the GRB case here to a set of arms $\mathcal{A}$ and use $x_i^{d_i}(s_i)$ to represent the total expected discounted time that arm i is in played to degree d_i in state s_i:

$$\begin{aligned} Z_1(\mathcal{A}) = \max \quad & \sum_{i\in\mathcal{A}} \sum_{d_i=0}^{D_i} \sum_{s_i\in S_i} R_i^{d_i}(s_i) x_i^{d_i}(s_i) \\ \text{s.t.} \quad & \sum_{d_i=0}^{D_i} x_i^{d_i}(s_i) = \alpha_{s_i} + \beta \sum_{s_i'\in S_i} \sum_{d_i=0}^{D_i} P_i^{d_i}(s_i' \to s_i) x_i^{d_i}(s_i'),\ \forall\, i \in \mathcal{A},\ s_i \in S_i, \end{aligned} \tag{2.3}$$

$$\begin{aligned} &\sum_{i\in\mathcal{A}}\sum_{s_i\in S_i}\sum_{d_i=0}^{D_i} d_i x_i^{d_i}(s_i) = \frac{M}{1-\beta}, \\ &x_i^{d_i}(s_i) \geq 0, \forall\, i \in \mathcal{A},\ s_i \in S_i,\ d_i = 0,\ldots,D_i. \end{aligned} \tag{2.4}$$

This problem can be viewed as solving the generalization of Whittle's relaxation simultaneously over the projections of the GRB polytope onto the dynamics of each individual arm. These projections are coupled by the time-average budgeting constraint: $\sum_{i\in\mathcal{A}}\sum_{s_i\in S_i}\sum_{d_i=0}^{D_i} d_i x_i^{d_i}(s_i) = \frac{M}{1-\beta}$.

We can obtain higher order relaxations by considering partitions of the arms into groups and examining the projections of the GRB polytope onto the joint-dynamics of each group. For example, if we consider a partition of the arms into pairs $\{i,j\} \in \mathcal{Q} \subset 2^{\mathcal{A}}$ and use the variable $x_{i,j}^{d_i,d_j}(s_i,s_j)$ to denote the total expected discounted time that arm i is played to degree d_i and arm j is played to degree d_j in the state (s_i,s_j), we obtain the following second order relaxation:
$Z_2(\mathcal{Q}) =$

$$\begin{aligned} \max\ &\sum_{\{i,j\}\in\mathcal{Q}}\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{D_j,M-d_i\}}\sum_{(s_i,s_j)\in S_i\otimes S_j} (R_i^{d_i}(s_i) + R_j^{d_j}(s_j))x_{i,j}^{d_i,d_j}(s_i,s_j) \\ s.t.\ &\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{D_j,M-d_i\}} x_{i,j}^{d_i,d_j}(s_i,s_j) = \alpha_{s_i,s_j} \\ &\qquad +\beta \sum_{(s_i',s_j')\in S_{i,j}}\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{D_j,M-d_i\}} P_i^{d_i}(s_i' \to s_i)\cdot P_j^{d_j}\left(s_j' \to s_j\right) x_{i,j}^{d_i,d_j}(s_i',s_j'), \\ &\qquad\qquad \forall\, \{i,j\} \in \mathcal{Q},\ s_i \in S_i,\ s_j \in S_j, \\ &\sum_{\{i,j\}\in\mathcal{Q}}\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{D_j,M-d_i\}}\sum_{(s_j,s_j)\in S_{i,j}} (d_i + d_j)x_{i,j}^{d_i,d_j}(s_i,s_j) = \frac{M}{1-\beta}, \\ &x_{i,j}^{d_i,d_j}(s_i,s_j) \geq 0, \\ &\quad \forall\, \{i,j\} \in \mathcal{Q},\ s_i \in S_i,\ s_j \in S_j,\ d_i = 0,\ldots,D_i, d_j = 0,\ldots,\min\{D_j, M - D_i\}. \end{aligned} \tag{2.5}$$

A series of refining partitions of the arms beginning with all n arms leads to a hierarchy of linear relaxations providing a non-decreasing upper bound on the optimal solution.

For relaxations of higher order, we partition the arms into groups $g \in \mathcal{G}$ of size ℓ. We use $x_g^{\boldsymbol{d}}(\boldsymbol{s})$ to represent the total expected discounted time that group g is played to degree $\boldsymbol{d}$ in state $\boldsymbol{s}$. The ℓ-order relaxation is then:

$$\begin{aligned} Z_\ell(\mathcal{G}) = \max \quad & \sum_{g \in \mathcal{G}} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \sum_{\boldsymbol{s} \in S_g} \left(\sum_{i \in g} R_i^{d_i}(s_i) \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}) \\ s.t. \quad & \sum_{\boldsymbol{d} \in \mathcal{D}_g} x_g^{\boldsymbol{d}}(\boldsymbol{s}) = \alpha_{\boldsymbol{s}} + \beta \sum_{\boldsymbol{s}' \in S_g} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \left(\prod_{i \in g} P_i\,(s_i' \to s_i) \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}'), \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \forall\, g \in \mathcal{G},\ \boldsymbol{s} \in S_g, \\ & \sum_{g \in \mathcal{G}} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \sum_{\boldsymbol{s} \in S_g} \left(\sum_{i \in g} d_i \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}) = \frac{M}{1-\beta}, \\ & x_g^{\boldsymbol{d}}(\boldsymbol{s}) \geq 0,\ \forall\, g \in \mathcal{G},\ \boldsymbol{s} \in S_g,\ \boldsymbol{d} \in \mathcal{D}_g, \end{aligned} \tag{2.6}$$

where the set $\mathcal{D}_g = \left\{ \boldsymbol{d} \in \mathbb{Z}_+^\ell \cap \bigotimes_{i \in g} \{0 \leq d_i \leq D_i\} : \sum_{i \in g} d_i \leq M \right\}$ and $S_g = \bigotimes_{i \in g} S_i$.

For a fixed ℓ, ℓ-order relaxations can be solved in polynomial time as a linear optimization problem providing an efficiently computable bound. For the remainder of this work, we use the second order relaxation, $\ell = 2$.

2.3.1 Generalized Primal-Dual Method

Bertsimas and Niño Mora [8] propose a primal-dual heuristic for the restless bandit problem based on the linear relaxation Z_1. In the classical restless bandit setting, the first-order relaxation is given by:

$Z_1(\mathcal{A}) =$

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{A}} \sum_{s_i \in S_i} \left(R_i^0(s_i) x_i^0(s_i) + R_i^0(s_i) x_i^0(s_i) \right) \\ s.t. \quad & x_i^0(s_i) + x_i^1(s_i) = \alpha_{s_i} + \beta \textstyle\sum_{s_i' \in S_i} \sum_{d_i=0}^{1} P_i^{d_i}(s_i' \to s_i) x_i^{d_i}(s_i'),\ \forall\, \boldsymbol{i} \in \mathcal{A},\ s_i \in S_i, \\ & \sum_{i \in \mathcal{A}} \sum_{s_i \in S_i} x_i^1(s_i) = \frac{M}{1-\beta}, \\ & x_i^0(s_i), x_i^1(s_i) \geq 0,\ \forall\, \boldsymbol{i} \in \mathcal{A},\ s_i \in S_i,\ d_i = 0, \ldots, D_i. \end{aligned} \tag{2.7}$$

The primal-dual method hinges on finding the optimal reduced costs $\gamma_i^0(s_i)$ and $\gamma_i^1(s_i)$ associated with optimal values of $x_i^0(s_i)$ and $x_i^1(s_i)$ which have the following natural interpretation:

- $\gamma_i^0(s_i)$: the rate of decrease in the objective function value of the LP relaxation per unit increase in $x_i^0(s_i)$.

- $\gamma_i^1(s_i)$: the rate of decrease in the objective function value of the LP relaxation per unit increase in $x_i^1(s_i)$.

Bertsimas Niño Mora [8] primal-dual approach reduces to assigning and index $\delta_i(s_i) = \gamma_i^1(s_i) - \gamma_i^0(s_i)$ to each state of each arm and, in each time period, playing the arms with the M lowest indices.

The natural extension of this primal-dual approach to the generalized restless bandit setting is as follows:

- Solve the first-order linear programming bound $Z_1(\mathcal{A})$ to obtain optimal reduced costs $\gamma_i^{d_i}(s_i)$ associated with $x_i^{d_i}(s_i)$.

- In each time period, when the system is in state $\bar{s}$, solve the multiple-choice knapsack problem:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{A}} \gamma_i^{d_i}(\bar{s}_i) \cdot y_i^{d_i} \\ s.t. \quad & \sum_{i \in \mathcal{A}} \sum_{d_i=0}^{D_i} d_i \cdot y_i^{d_i} = M \\ & \sum_{d_i=0}^{D_i} = 1, \forall i \in \mathcal{A} \\ & y_i^{d_i} \in \{0,1\}, \forall i \in \mathcal{A},\, d_i = 0, \ldots, D_i \end{aligned} \tag{2.8}$$

- Play arm i to degree d_i where $y_i^{d_i} = 1$ in the optimal solution.

Note that in practice, the involved multiple-choice knapsack problems can be solved efficiently using a Lagrangian relaxation approach, relaxing the constraint

$$\sum_{i \in \mathcal{A}} \sum_{d_i=0}^{D_i} d_i \cdot y_i^{d_i} = M \tag{2.9}$$

or by approximate dynamic programming methods such as those by Bertsimas and Demir [4] or when of manageable size, full dynamic programming such as found in Pisinger [41].

We will use this generalized primal-dual approach to benchmark our Nested Policy heuristic in Section 2.5.

2.4 The Nested Policy Approach

In this section, we outline the derivation and implementation of the Nested Policy Heuristic. This algorithm is briefly described as follows:

1. Partition the set of available arms into smaller subsets of size ℓ, ℓ is a parameter of the algorithm.
2. Solve the GRB problem exactly over each of these subsets.
3. Replace each subset with a single arm having k states representative of the dynamics of the subset under the policy derived in Step 2.
4. The above steps reduce the number of arms we are dealing with by a factor of ℓ each time they are performed. We repeat them until we are left with only one arm.

We can control the level of approximation of the Nested Policy Algorithm by adjusting the size, ℓ, of the subsets found in Step 1 above and the maximum size, k, of the state space for the representative arm created in Step 3 above.

To aid the reader's understanding of the Nested Policy approach, we begin with a small example of its implementation and then present a conceptual description of the heuristic.

2.4.1 Example

Suppose we have a GRB problem with a budget of $M = 2$ and eight arms labeled $i = 1, ..., 8$, each with 2 possible states, $|S_i| = 2$, and each arm able to be played up to a maximum degree $D_i = 2$. The following steps outline the derivation of a Nested Policy with $\ell = 2$ and $k = 2$:

- We decompose the problem by considering pairs of arms, and for each pair deciding how a given budget of some number D should be divided between the two paired arms, while ignoring the dynamics of all arms outside the pair.

- For any given pair we need to know how budgets of $D = 0, 1, 2$ should be divided between the paired arms since each of these budgets correspond to playing the paired arms to a total degree of D and those arms outside the pair to a total degree of $M - D$.

- We solve each subproblem created in this manner for a pair of arms $\{i, j\}$ and budget of D by solving the corresponding two-armed GRB exactly using the performance-region approach which involves solving the linear optimization problem (2.2) discussed in Section 2.2.1.

- This approach limits us in that the total budget constraint of M is the only link we consider between the dynamics of arms in one pair and the dynamics of arms outside that pair in the policy we derive. This limitation corresponds to the second-order relaxation of the GRB, $Z_2(\mathcal{Q})$, for the particular pairing $\mathcal{Q}$ of arms selected. Thus we would like to choose the pairing $\mathcal{Q}^*(\{1, \ldots, 8\})$ which maximizes $Z_2(\mathcal{Q})$. We call this the Arm Aggregation problem which can be modeled as a mixed-integer optimization problem as shown in equation (2.10) in Section 2.4.2.

Construction of the Policy:

1. Let $\mathcal{A}_1 = \{1, ..., 8\}$ be the set of active arms. Suppose we solve the Arm Aggregation problem (equation (2.10) in Section 2.4.2) to obtain the pairing: $\mathcal{Q}^*(\mathcal{A}_1) = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$.

2. Consider the pair $\{1,2\}$. Jointly these arms can be in one of four states as shown in Table 2.1. Given that we allocate some portion D of the total buget M to arms 1 and 2 combined, we would like to know how D should be divided between arms 1 and 2 in each of these states. We answer this question for each possible value of $D = 0,1,2$ by solving an instance of the two-armed GRB, equation (2.2) in Section 2.2.1, exactly over arms 1 and 2 with a total budget of D.

	$s_2 = 1$	$s_2 = 2$
$s_1 = 1$	$\mathbf{s}_{\{1,2\}} = (1,1)$	$\mathbf{s}_{\{1,2\}} = (1,2)$
$s_1 = 2$	$\mathbf{s}_{\{1,2\}} = (2,1)$	$\mathbf{s}_{\{1,2\}} = (2,2)$

Table 2.1: Joint states for arms 1 and 2.

3. Suppose the optimal policies for state $\mathbf{s}_{\{1,2\}} = (1,1)$ and $D = 0,1,2$ are as shown in Table 2.2. Let us fix this portion of the policy by always playing arm 1 to degree 1 and arm 2 to degree 0 when they are in state $(1,1)$ and we have chosen to play arms 1 and 2 to a combined degree $D = 1$. We similarly fix our policy for all four joint states and values of D.

4. The key insight is that once this portion of the policy is fixed, we can view the behavior of arm 1 and arm 2 together as that of a single arm $A_{\{1,2\}}$ which can be played to a maximum degree $D_{A_{\{1,2\}}} = 2$ and whose transition probabilities and rewards are governed by the portion of the policy described in Table 2.2.

	Allocation to Arm 1	Allocation to Arm 2
$D = 0$	0	0
$D = 1$	1	0
$D = 2$	1	1

Table 2.2: Optimal policies for state $\mathbf{s}_{\{1,2\}} = (1, 1)$.

For instance:

$$\begin{aligned} P^1_{A_{\{1,2\}}}((1,1) \rightarrow (1,2)) &= P^1_1(1 \rightarrow 1) \cdot P^0_2(1 \rightarrow 2) \\ R^1_{A_{\{1,2\}}}((1,1)) &= R^1_1(1) + R^0_2(1) \end{aligned}$$

The state space of this new arm is of size 4, whereas our original arms had state spaces of size 2. If we continue combining this new arm with others directly, the state space for each new arm would grow exponentially.

5. To address the issue of exponential state space growth, we approximate the behavior of the arm $A_{\{1,2\}}$ by another arm $\tilde{A}_{\{1,2\}}$ on a smaller state space of size $k = 2$. This is accomplished by clustering the states $(1,1), (1,2), (2,1), (2,2)$ into two clusters. This approximation will limit us in that when we derive further portions of our policy we cannot differentiate between states in the same cluster. Selecting the best state clustering in light of this limitation can be modeled as a mixed-integer optimization problem as shown in equation (2.14) in Section 2.4.2. We call this the State Clustering problem. For this example, suppose the clusters derived are as in Table 2.3.

$\mathbf{s}_{\{1,2\}} = (1, 1)$ or $(1, 2)$	→	$\tilde{s}_{\{1,2\}} = 1$
$\mathbf{s}_{\{1,2\}} = (2, 1)$ or $(2, 2)$	→	$\tilde{s}_{\{1,2\}} = 2$

Table 2.3: Clustered states for arms 1 and 2.

6. We repeat Steps 2-5 above for the other subsets of $\mathcal{Q}^*(\mathcal{A}_1)$: $\{3,4\}$, $\{5,6\}$, and $\{7,8\}$ to obtain four new arms in total: $\mathcal{A}_2 = \{\tilde{A}_{\{1,2\}}, \tilde{A}_{\{3,4\}}, \tilde{A}_{\{5,6\}}, \tilde{A}_{\{7,8\}}\}$ each on a state space of size 2.

7. We again solve the Arm Aggregation problem, now on this new set of arms, and obtain the pairing:
$\mathcal{Q}^*(\mathcal{A}_2) = \Big\{\{\tilde{A}_{\{1,2\}}, \tilde{A}_{\{3,4\}}\}, \{\tilde{A}_{\{5,6\}}, \tilde{A}_{\{7,8\}}\}\Big\}$.

8. We repeat steps 2-5 for the subsets $\{\tilde{A}_{\{1,2\}}, \tilde{A}_{\{3,4\}}\}$ and $\{\tilde{A}_{\{5,6\}}, \tilde{A}_{\{7,8\}}\}$ to obtain two new arms $\tilde{A}_{\{1,2,3,4\}}$ and $\tilde{A}_{\{5,6,7,8\}}$ both on a state space of size 2.

9. Now we are left with only two active arms $\mathcal{A}_3 = \{\tilde{A}_{\{1,2,3,4\}}, \tilde{A}_{\{5,6,7,8\}}\}$ and thus only one possible pairing. Repeating Steps 2 and 3 on this pairing completes derivation on the Nested Policy.

Implementation the Policy:

- To implement the Nested Policy when the system of original arms is in a particular state $\mathbf{s}$, we examine how each pairwise joint state was clustered in each iteration of Step 5 to determine the corresponding states of our approximate arms $\tilde{A}_{\{\cdot\}}$. To illustrate this suppose the original arms are in state $\mathbf{s} = (1,1,1,1,1,1,1,1)$ and that the relevant clusters derived in iterations of Step 5 are as in Table 2.4. These clusterings give us the states of all approximating

$\mathbf{s}_{\{1,2\}} = (1,1)$	$\rightarrow$	$\tilde{s}_{\{1,2\}} = 1$
$\mathbf{s}_{\{3,4\}} = (1,1)$	$\rightarrow$	$\tilde{s}_{\{3,4\}} = 1$
$\mathbf{s}_{\{5,6\}} = (1,1)$	$\rightarrow$	$\tilde{s}_{\{5,6\}} = 2$
$\mathbf{s}_{\{7,8\}} = (1,1)$	$\rightarrow$	$\tilde{s}_{\{7,8\}} = 2$
$\mathbf{s}_{\{\tilde{A}_{\{1,2\}},\tilde{A}_{\{3,4\}}\}} = (1,1)$	$\rightarrow$	$\tilde{s}_{\{1,2,3,4\}} = 1$
$\mathbf{s}_{\{\tilde{A}_{\{5,6\}},\tilde{A}_{\{7,8\}}\}} = (2,2)$	$\rightarrow$	$\tilde{s}_{\{5,6,7,8\}} = 2$

Table 2.4: Relevant clusters for example Nested Policy implementation.

arms corresponding to the state $(1,1,1,1,1,1,1,1)$ of the original system. We then examine the solutions we derived from solving two-armed GRBs in Step 2 at each level. We first allocate M between $\{1,2,3,4\}$ and $\{5,6,7,8\}$; then we refine this allocation to $\{1,2\},\{3,4\},\{5,6\}$, and $\{7,8\}$; and finally refine down to $\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\}$, and $\{8\}$. Suppose Table 2.5 shows the relevant portions of optimal derived optimal policies. By breaking down the total budget of $M = 2$ in this nested fashion, we find that we should play the arms 1 and 4 to degree 1 and the rest to degree 0.

First Arm State, Second Arm State, Budget	Allocate to First Arm	Allocate to Second Arm
$\tilde{s}_{\{1,2,3,4\}} = 1, \tilde{s}_{\{5,6,7,8\}} = 2, D = M = 2$	2	0
$\tilde{s}_{\{1,2\}} = 1, \tilde{s}_{\{3,4\}} = 1, D = 2$	1	1
$\tilde{s}_{\{5,6\}} = 2, \tilde{s}_{\{7,8\}} = 2, D = 0$	0	0
$s_1 = 1, s_2 = 1, D = 1$	1	0
$s_3 = 1, s_4 = 1, D = 1$	0	1
$s_5 = 1, s_6 = 1, D = 0$	0	0
$s_7 = 1, s_8 = 1, D = 0$	0	0

Table 2.5: Relevant portions of two-armed GRB solutions for example Nested Policy implementation.

2.4.2 The Nested Policy Heuristic

Here we give a detailed description of our approach for any choice of ℓ and k.

For the GRB problem, we describe a policy over a set of arms g on a state space S_g with a total budget D by the mapping $\pi_g^D : S_g \rightarrow \mathbb{Z}_+^{|g|}$ which allocates D over the members of g. We use the notation $\pi_g^D(\mathbf{s}_g)|_i$ to refer to the i^{th} entry in this allocation, that is, the degree that the policy allocates to arm i when in state $\mathbf{s}_g$.

Algorithm 1 (Nested Policy Construction). *A Nested Policy with subset size ℓ and a number of representative states k for the GRB problem with n arms and a per-period budget of M is constructed as follows:*

1. *Let $j := 1$ and let $\mathcal{A}_j$ be the set of arms for the original GRB.*

2. ***Arm Aggregation:***

   - *Input: A set of arms $\mathcal{A}_j$ and a subset size ℓ.*
   - *Output: A partitioning $\mathcal{G}^*(\mathcal{A}_j)$ of the arms in $\mathcal{A}_j$ into subsets of size ℓ.*
   - *Objective: Our policy will consider joint dynamics of arms within a subset at the granularity of their product state-space. The joint dynamics of arms in different subsets are only coupled by the time-average fraction of the total budget allocated to each subset. We wish to pick the partitioning which minimizes the impact of this limitation.*
   - *Method: We solve $\mathcal{G}^*(\mathcal{A}_j) := \arg\max_{\mathcal{G}} Z_\ell(\mathcal{G})$ which can be modeled as a mixed-integer linear optimization problem. If $|\mathcal{A}|$ is not divisible by ℓ, we add a sufficient number of dummy arms to $\mathcal{A}$ with each with one state and no rewards. We present the formulation for $\ell = 2$ here; formulation for general ℓ can be found in Appendix A.*

     Define the binary variable $w(i,j)$ for all $\{i,j\} \subset \mathcal{A}$ such that $w(i,j) = 1$ if $\{i,j\} \in \mathcal{G}^*$. The Arm Aggregation problem for $\ell = 2$ is then:

$$\max_{\boldsymbol{x},\boldsymbol{w}} \sum_{\{i,j\}\subset\mathcal{A}} \sum_{d_i=0}^{D_i} \sum_{d_j=0}^{\min\{M-d_i,D_j\}} \sum_{(s_i,s_j)\in S_i\otimes S_j} \left(R_i^{d_i}(s_i) + R_j^{d_j}(s_j)\right) x_{i,j}^{d_i,d_j}(s_i,s_j) \tag{2.10}$$

$$s.t.\ x_{i,j}^{d_i,d_j}(s_i,s_j) \le Uw(i,j), \tag{2.11}$$

$$\forall\,\{i,j\} \subset \mathcal{A},\ s_i \in S_i,\ s_j \in S_j,\ d_i = 0,\dots,D_i,\ d_j = 0,\dots,\min\{M-d_i,D_j\},$$

$$\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{M-d_i,D_j\}} x_{i,j}^{d_i,d_j}(s_i,s_j) = \alpha_{s_i,s_j} w(i,j) \tag{2.12}$$

$$+\beta \sum_{(s_i',s_j')\in S_i\otimes S_j}\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{M-d_i,D_j\}} P_i^{d_i}\left(s_i'\to s_i\right)\cdot P_j^{d_j}\left(s_j'\to s_j\right) x_{i,j}^{d_i,d_j}(s_i',s_j'),$$

$$\forall\{i,j\}\subset\mathcal{A},\ s_i\in S_i,\ s_j\in S_j,$$

$$\sum_{\{i,j\}\subset\mathcal{A}}\sum_{d_i=0}^{D_i}\sum_{d_j=0}^{\min\{M-d_i,D_j\}}\sum_{(s_j,s_j)\in S_i\otimes S_j}(d_i+d_j)x_{i,j}^{d_i,d_j}(s_i,s_j) = \frac{M}{1-\beta},$$

$$\sum_{\{i,j\}\ni k} w(i,j) = 1,\ \forall k\in\mathcal{A}, \tag{2.13}$$

$$x_{i,j}^{d_i,d_j}(s_i,s_j)\geq 0,$$

$$\forall\{i,j\}\subset\mathcal{A},\ s_i\in S_i,\ s_j\in S_j,\ d_i=0,\ldots,D_i, d_j=0,\ldots,\min\{M-d_i,D_j\},$$

$$w(i,j)\in\{0,1\},\ \forall\{i,j\}\subset\mathcal{A}.$$

Constraints (2.11) *ensure that if the pair* $\{i,j\}$ *is not chosen* $x_{i,j}^{d_i,d_j}(s_i,s_j)$ *must be set to zero for all* (s_i,s_j)*;* $U = \frac{M}{1-\beta}$ *here is a sufficiently large constant. In constraints* (2.12), *we then multiply* α_{s_i,s_j} *by* $w(i,j)$ *to ensure that both sides of the constraint vanish if the pair* $\{i,j\}$ *is not chosen. The constraints* (2.13) *ensure that each arm appears in exactly one selected pair and thus all pairs chosen constitute a partition of* $\mathcal{A}$.

3. For each subset of arms $g\in\mathcal{G}^*(\mathcal{A}_j)$*:*

(a) ***Arm Substitution:***

- *Input: A set of arms* g
- *Output: A new arm* A_g *on a state space* S_g *representative of the dynamics of* g *under a fixed optimal sub-policy, and vectors* y_g^D *for* $D = 0,...,\min\{M,\sum_{i\in g} D_i\}$ *with* $y_g^D(s_g)$ *representing the expected discounted fraction of time that the set* g *is in state* s_g *if the arms in* g *are played to a total degree* D *under this fixed optimal sub-policy.*

- *Method: For each* $D = 0, ..., \min\{M, \sum_{i \in g} D_i\}$, *solve the GRB problem over* g *with a budget of* D *exactly using the performance region approach as in Equation (2.2) to obtain a policy* $\pi_g^D(\cdot)$. *We use the notation* $x_{s_g}^{d,*}(D)$ *to denote the optimal values of the variables* $x^d(s)$ *in this implementation of Equation (2.2).* y_g^D *is the given by:*

$$y_g^D(s_g) = \sum_{d:\sum_{i \in g} d_i = D} x_{s_g}^{d,*}(D)$$

Create a new arm A_g *on the state space* $S_g = \bigotimes_{i \in g} S_i$ *with maximum degree* $D_{A_g} = \min\{M, \sum_{i \in g} D_i\}$ *as follows:*

  - $P_g^D(\mathbf{s}_g \to \mathbf{s}'_g) = \prod_{i \in g} P_i^{D_i^*}(\rho_i(\mathbf{s}_g) \to \rho_i(\mathbf{s}'_g))$
  - $R_g^D(\mathbf{s}_g) = \sum_{i \in g} R_i^{D_i^*}(\rho_i(\mathbf{s}_g))$

  Where $\rho_i(\mathbf{s}_g)$ *is the projection of* $\mathbf{s}_g$ *onto the state space of arm* i, S_i; *and* $D_i^* = \pi_g^D(\mathbf{s}_g)|_i$ *is the optimal allocation to arm* i *under the derived optimal sub-policy.*

4. ***State Clustering:***

- *Input: A set of arms* $\mathcal{A}_{\mathcal{G}^*}$, *each* $A_g \in \mathcal{A}_{\mathcal{G}^*}$ *having* $|S_g|$ *states, vectors* y_g^D, *and a number* k.

- *Output: A new set of arms* $\tilde{\mathcal{A}}_{\mathcal{G}^*}$, *each* $\tilde{A}_g \in \tilde{\mathcal{A}}_{\mathcal{G}^*}$ *with no more than* k *states which approximates the dynamics of* A_g *and a mapping* $\phi(\cdot)$ *which maps the states* S_g *of* A_g *to corresponding states* $\tilde{S}_g$ *of* $\tilde{A}_g$.

- *Method: For each* A_g, *if* $|S_g| \leq k$, *we leave the state space* S_g *unchanged and let* $\tilde{S}_g = S_g$. *Let* $C \subseteq \mathcal{A}_{\mathcal{G}^*}$ *be the set of* A_g *such that* $|S_g| > k$, *we will shrink the state-space of these arms. Note that in future steps our policy will be limited in that it will not be able to differentiate between states in the same cluster. We wish to pick the clustering which minimizes the impact of* *this limitation. This clustering problem can be modeled as a mixed integer linear optimization problem as follows.*

For each $i \in C$, each $\boldsymbol{s}_i \in S_i$ and each $\tilde{s}_i \in \tilde{S}_i = \{1, \dots, k\}$ ***define the binary*** *variable $\varphi_i(\boldsymbol{s}_i, \tilde{s}_i)$ such that $\varphi_i(\boldsymbol{s}_i, \tilde{s}_i) = 1$ if $\phi(\boldsymbol{s}_i) = \tilde{s}_i$.* ***Additionally, define*** *the binary variables $\psi_i^d(\tilde{s}_i)$ which will indicate whether degree d is allocated to arm i in clustered state $\tilde{s}_i$.* ***We also define auxiliary variables*** *$z_i^d(\boldsymbol{s}_i, \tilde{s}_i)$ which are linearizations of equation* (2.24) *below. The State Clustering problem is then:*

$$\max_{\varphi, \boldsymbol{x}, \boldsymbol{\psi}} \sum_{i \in \mathcal{A}_{\mathcal{G}^*}} \sum_{d=0}^{M} \sum_{\boldsymbol{s}_i \in S_i} R_i^d(\boldsymbol{s}_i) x_i^d(\boldsymbol{s}_i) \tag{2.14}$$

$$s.t. \sum_{d=0}^{D_i} x_i^d(\boldsymbol{s}_i) = \boldsymbol{\alpha}_{\boldsymbol{s}_i} + \beta \sum_{\boldsymbol{s}_i' \in S_i} \sum_{d=0}^{M} P_i^d(\boldsymbol{s}_i' \rightarrow \boldsymbol{s}_i) x_i^d(\boldsymbol{s}_i'), \, \forall\, i \in \mathcal{A}_{\mathcal{G}^*}, \, \boldsymbol{s}_i \in S_i,$$

$$\sum_{i \in \mathcal{A}_{\mathcal{G}^*}} \sum_{\boldsymbol{s}_i \in S_i} \sum_{d=0}^{D_i} d x_i^d(\boldsymbol{s}_i) = \frac{M}{1 - \beta},$$

$$\sum_{d=0}^{D_i} \psi_i^d(\tilde{s}_i) \leq 1, \, \forall\, i \in C, \, \tilde{s}_i \in \tilde{S}_i, \tag{2.15}$$

$$z_i^d(\boldsymbol{s}_i, \tilde{s}_i) \leq U \varphi(\boldsymbol{s}_i, \tilde{s}_i), \, \forall\, i \in C, \, \boldsymbol{s}_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \dots, M, \tag{2.16}$$

$$x_i^d(\boldsymbol{s}_i) - z_i^d(\boldsymbol{s}_i, \tilde{s}_i) \leq U(1 - \varphi(\boldsymbol{s}_i, \tilde{s}_i)), \tag{2.17}$$

$$\forall\, i \in C, \, \boldsymbol{s}_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \dots, D_i,$$

$$\sum_{\boldsymbol{s}_i \in S_i} z_i^d(\boldsymbol{s}_i, \tilde{s}_i) \leq U \psi_i^d(\tilde{s}_i), \, \forall\, i \in C, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \dots, D_i, \tag{2.18}$$

$$0 \leq z_i^d(\boldsymbol{s}_i, \tilde{s}_i) \leq x_i^d(\boldsymbol{s}_i), \, \forall\, i \in C, \, \boldsymbol{s}_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \dots, M, \tag{2.19}$$

$$\sum_{\tilde{s}_i \in \tilde{S}_i} \varphi(\boldsymbol{s}_i, \tilde{s}_i) = 1, \, \forall\, i \in C, \, \boldsymbol{s}_i \in S_i, \tag{2.20}$$

$$x_i^d(\boldsymbol{s}_i) \geq 0, \, \forall\, i \in \mathcal{A}_{\mathcal{G}^*}, \, \boldsymbol{s}_i \in S_i, \, d = 0, \dots, D_i, \tag{2.21}$$

$$\psi_i^d(\tilde{s}_i) \in \{0,1\},\ \forall i \in C,\ \tilde{s}_i \in \tilde{S}_i,\ d = 0,\dots,D_i \tag{2.22}$$

$$\varphi(s_i,\tilde{s}_i) \in \{0,1\},\ \forall i \in C,\ s_i \in S_i, \tilde{s}_i \in \tilde{S}_i. \tag{2.23}$$

Constraints (2.15) *ensure that at most one degree d is chosen uniformly to be played for all states in a cluster. The constraints* (2.16),(2.17),(2.18),(2.19) *together linearize the non-linear implications:*

$$\psi_i^d(\tilde{s}_i) = 0 \Rightarrow \sum_{s_i \in S_i} x_i^d(s_i)\varphi(s_i,\tilde{s}_i) = 0,\ \forall i,\ \tilde{s}_i \in \tilde{S}_i,\ d = 0,\dots,D_i,$$

through the sufficiently large constant $U = \frac{M}{1-\beta}$ *and the auxiliary variables*

$$z_i^d(s_i,\tilde{s}_i) = x_i^d(s_i)\varphi(s_i,\tilde{s}_i) \tag{2.24}$$

.

Once the optimum clustering is obtained from this mixed-integer optimization problem, for each arm $i \in C$*, we reduce the underlying Markov chains* $P^d_{A_g}$ *on* S_g *to the corresponding* optimal *Markov chains on the clustered state space* $\tilde{S}_g$ *with respect to Kullback-Leibler (K-L) divergence metric using the method of Deng et al. [15]. The corresponding transition probabilities are:*

$$P_i^d(\tilde{s} \to \tilde{s}') = \sum_{s \in \phi^{-1}(\tilde{s})} \left(\frac{y_i^d(s)}{\sum\limits_{s \in \phi^{-1}(\tilde{s})} y_i^d(s)} \right) \sum_{s' \in \phi^{-1}(\tilde{s}')} P_i^d(s) \to (s'),\ \forall\, \tilde{s}, \tilde{s}' \in \tilde{S}_i.$$

Similarly, the rewards of the clustered state $\tilde{s}$ *can be approximated as follows:*

$$R_i^d(\tilde{s}) = \left(\frac{1}{\sum\limits_{s \in \phi^{-1}(\tilde{s})} y_i^d(s)} \right) \sum_{s \in \phi^{-1}(\tilde{s})} y_i^d(s) R_i^d(s),\ \forall\, \tilde{s} \in \tilde{S}_i^q.$$

5. Set $\mathcal{A}_{j+1} := \bigcup_{g \in \mathcal{G}^*(\mathcal{A}_j)} \{\tilde{A}_g\}$.

6. If $|\mathcal{A}_{j+1}| = 1$ *STOP, else let* $j := j + 1$ *and goto Step 2.*

Notice that $|\mathcal{A}_{j+1}| \leq \left\lceil \frac{|\mathcal{A}_j|}{\ell} \right\rceil$, so Algorithm 1 terminates after $\lceil \log_\ell(n) \rceil$ iterations. Additionally, with ℓ fixed, each GRB subproblem we solve in Step 3(a) of Algorithm 1 is a linear optimization problem whose dimension is polynomial in the input size of the original GRB problem.

For GRBs where arms have moderately sized state-spaces the formulation of the State Clustering problem 2.14 can be solved efficiently by modern integer optimization solvers. However, if there are many arms in the GRB and each arm has thousands of states, solving the State Clustering problem as a mixed-integer optimization problem can prove prohibitively expensive. In such instances, we have found two alternative to be empirically successful.

The first approach is that the State Clustering problem can be solved independently for each arm in series rather than simultaneously for all arms that must be clustered. This can be accomplished by solving (2.14) on the set of arms $(\mathcal{A} \backslash g) \cup A_g$ and clustering the singleton A_g for each $A_g \in C$ in Step 4 of Algorithm (1). In this way, we solve $|C|$ mixed-integer optimization problems but the number of continuous and binary variables is greatly reduced.

Alternatively, we can consider the State Clustering problem as a traditional clustering problem with respect to the arm states S_i. For each state space S_i, we would like to generate $\bar{k}$-partitions, which can be defined by the aggregation function $\phi_i : S_i \rightarrow \tilde{S}_i$, where $\tilde{S}_i$ is the index set of the partition. For each state $s_i \in S_i$, we have a reward vector $R_i^d(s_i) \in \mathbb{R}^{M+1}$. Empirically, we have found that clustering states based on this reward vector using a traditional method such as *spectral clustering* discussed by von Luxburg [50] provides a feasible solution for our mixed-integer optimization formulation of the State Clustering problem which is not too far from optimal. Thus traditional clustering methods can be used in lieu of a mixed-integer optimization approach for large problems or to provide a good initial solution for an integer optimization solver to speed up solution time for moderately sized problems.

Once the Nested Policy is constructed using Algorithm 1, it is implemented through Algorithm 2.

Algorithm 2 (Nested Policy Implementation). *Let J be the total number of iterations run in Nested Policy Construction. Using the objects derived in Algorithm 1, the Nested Policy is implemented through a call to the recursive function* ***nestedPolicy***$(\cdot)$*:*

1. *Let $\mathbf{s}_{\mathcal{A}_1} \in S$ be the state of the original n bandits.*
2. *For each arm $i \in \mathcal{A}_1$, play i to degree* ***nestedPolicy***$(1, i, \mathbf{s}_{\mathcal{A}_1})$

- *function* ***nestedpolicy***$(j, i, \mathbf{s}_{\mathcal{A}_j})$
    1. *If $j = J$, RETURN $\pi^M_{\mathcal{A}_j}(\mathbf{s}_{\mathcal{A}_j})|_i$ (this quantity represents the allocation of the total budget M to arm i in the clustered state $\mathbf{s}_{\mathcal{A}_j}$);*
    2. *otherwise:*
        (a) *For each $g \in \mathcal{G}^*(\mathcal{A}_j)$, let $\mathbf{s}_g$ be the components of $\mathbf{s}_{\mathcal{A}_j}$ corresponding to members of g.*

        (b) *Let $\tilde{\mathbf{s}}_{\mathcal{A}_{j+1}}$ be a $|\mathcal{G}^*(\mathcal{A}_j)|$-dimensional vector with components $\tilde{s}_g = \phi(\mathbf{s}_g)$.*

        (c) *Let $d :=$* ***nestedPolicy***$(j+1, A_{\hat{g}}, \tilde{\mathbf{s}}_{\mathcal{A}_{j+1}})$ *where $\hat{g} \ni i$. This quantity d represents the amount budgeted to arm $A_{\hat{g}}$ when the system is in the state $\tilde{\mathbf{s}}_{\mathcal{A}_{j+1}}$*

        (d) *RETURN $\pi^d_{\hat{g}}(\mathbf{s}_{\hat{g}})|_i$ where $\hat{g} \ni i$ (this quantity represents the allocation of a budget of d to arm i in the clustered state $\mathbf{s}_{\hat{g}}$.*

2.5 Computational Results

In this section, we evaluate the Nested Policy heuristic's ability to solve the GRB problem in its full generality, and also gauge its performance on classical special cases of the GRB problem where well established algorithms may be used as benchmarks. To this end we evaluate the Nested Policy heuristic through simulation in three settings:

- The regular bandit setting, $D_i = 1$, $P^{\mathbf{0}}_{\boldsymbol{s}\to\boldsymbol{s}'} = \boldsymbol{I}$ where $\boldsymbol{I}$ is the $k_i \times k_i$ identity matrix. Here we compare to the optimal Gittins' index found in Gittins [19] and Whittle [54].

- The restless bandit setting, $D_i = 1$. Here we compare to the primal-dual heuristic of Bertsimas and Niño Mora [8].

- The generalized bandit setting where each arm can be played to an arbitrary degree D_i. In this regime we perform a more detailed analysis of large problems with parameter scaling and various problem structures. Here we compare to a myopic multiple-choice knapsack heuristic as well as the generalized primal-dual approach of Section 2.3.1.

Instances for simulation in the regular and restless settings were generated on bandits with 5 arms and 3 states. When performing State Clustering with the Nested Policy heuristic in these settings, expanded state spaces were clustered into 3 states. In the general setting, we explore larger bandit problems with 10 arms and 7 states with State Clustering into 7 states for the Nested Policy heuristic. We study performance as the discount rate is and the maximum degree of each arm vary in different instances as well as examine performance under structures of reward monotonicity and decreasing marginal returns. In all settings, arm Aggregation was performed with $\ell = 2$.

The probability transition matrices were generated as matrices of uniform $[0, 1]$, each row normalized to sum to 1. For the regular an restless settings, a bugdet of $M = 1$ is used and the discount factor used was $\beta = 0.9$ to avoid trivial study of myopic problems. For these settings, we generated 10 problem instances and simulated 600 trials to obtain a distribution of scores under each policy for each instance. Each trial ran for t time periods such that $\beta^t > 10^{-10}$. For each setting we present the average performance over all 10 instances, the average standard deviation across instances, and the average distance from the second-order optimality bound derived during the first Arm Aggregation step for both the Nested Policy and the associated benchmark.

In the general setting, we study problems with 10 arms, 7 states, and a total budget of $M = 8$. We examine performance as the discount rate is varied ($\beta \in \{.1, .5, .9\}$) and the maximum degree of each arm in increased $D_i \in \{3, 6\}$. We also study resilience to different reward structures:

- Rewards for each (arm, state, degree) triplet generated independently.
- Monotonicity: For each (arm, state) pair, rewards are monotonically non-decreasing in degree, $R_i^a(s_i) \geq R_i^b(s_i) \forall a \geq b$.
- Diminishing Returns: Rewards are monotonically non-decreasing in degree under decreasing marginal returns.

When called for, rewards were generated as follows to create an environment of decreasing marginal returns.:

1. $R_i^0(s_i) = 0$
2. For each state s_i, generate D_i uniform $[0, 1]$ numbers sorted in decreasing order $\{u_k(s_i)\}_{k=1}^{D_i}$.
3. $R_i^{d_i}(s_i) = \sum_{k=1}^{d_i} u_k(s_i)$

2.5.1 Regular Bandits

The baseline for comparison used is the Gittins' index policy which is the optimal policy for this problem.

Figure 2-1 shows the cumulative distribution of the discounted valuation obtained for one instance demonstrating that Nested Policy performance is close to optimal in distribution. As seen in Table 2.6, the Nested Policy performs within 2.5% of optimal on average over 10 instances and has only slightly higher variance than the optimal policy.

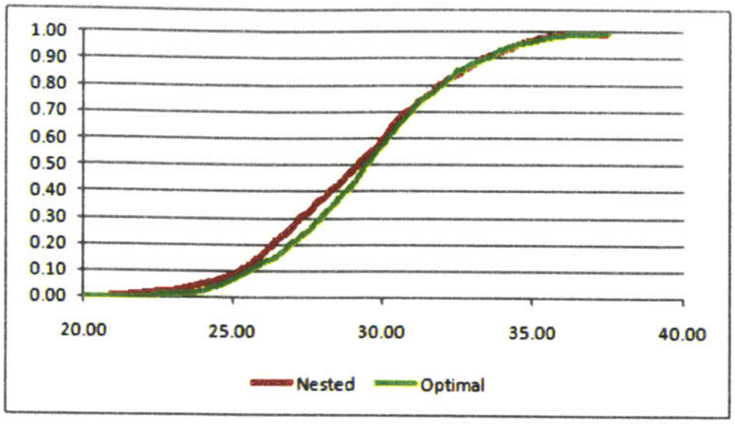


Figure 2-1: Regular Bandit Setting: Cumulative distribution of value obtained in simulation.

	Average Performance	Average Standard Deviation	Distance From Optimal Solution
Nested Policy	29.34	2.22	2.4%
Optimal Policy Benchmark	30.07	2.09	0.0%

Table 2.6: Regular Bandit Setting: Summary statistics from simulation (10 instances).

2.5.2 Restless Bandits

The baseline for comparison used is the primal/dual heuristic due to Bertsimas and Niño Mora [8]. This heuristic assigns an index to each state of each arm based on reduced costs from the associated first order relaxation. In each time period, the bandits whose current states have the highest index are played.

The Nested Policy and primal/dual heuristic have very close performance overall with the Nested Policy performing less than 2% better on average as seen in Table 2.7. Figure 2-2 shows that the value distributions for these policies nearly overlap for a particular instance. Figure 2-3 shows the value distribution for the nine other instances where it can be seen that average performance is affected by primal/dual

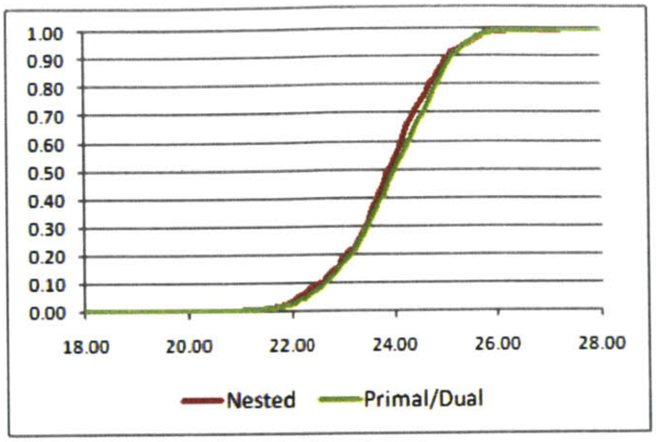


Figure 2-2: Restless Setting: Cumulative distribution of value obtained in simulation.

	Average Performance	Average Standard Deviation	Distance From Optimality Bound
Nested Policy	23.35	0.73	0.6%
Primal-Dual Benchmark	22.95	0.77	2.3%

Percentage of instances where Nested exceeds benchmark	30%

Table 2.7: Restless Setting: Summary statistics from simulation (10 instances).

under-performing significantly on 2 instances; on the majority of instances the performance of the two methods nearly overlaps or primal/dual slightly outperforms the Nested Policy.

2.5.3 Generalized Bandits

Here we compare the Nested Policy heuristic to two benchmarks:

- The generalized primal-dual approach of Section 2.3.1.
- A myopic multiple-choice knapsack heuristic.

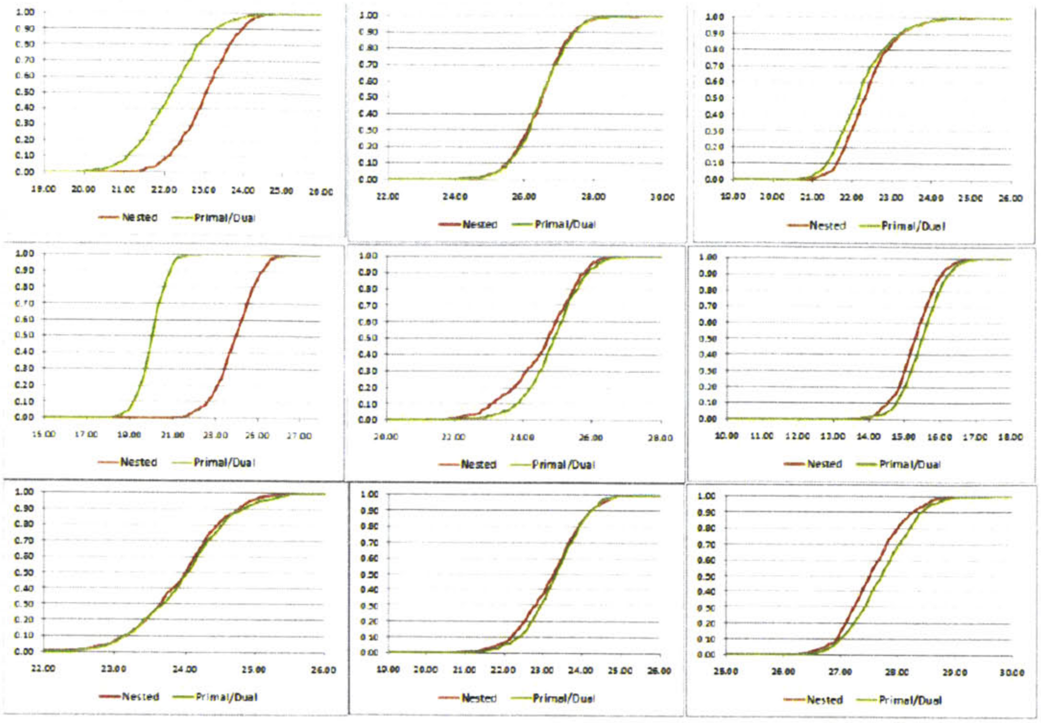


Figure 2-3: Restless Setting: cumulative distribution of value obtained in simulation for additional instances.

For the second benchmark, the myopic multiple-choice knapsack problem is solved in each time period to play that set of arms which maximizes rewards for the current time period. In each time period, this algorithm must pick exactly one degree number to play for each arm with a knapsack constraint such that the total combined degree picked is equal to M. The reward assigned to each degree d_i is equal to $R_{s_i}^{d_i}$ where s_i is the current state of arm i.

Table 2.8 shows performance of the three methods in 18 instances with various parameters and problem structures as a percentage of second-order optimality bounds. The problem structures listed in Table 2.8 are as follows:

- Independent: Rewards for each (arm, state, degree) triplet generated independently.

- Monotonic: For each (arm, state) pair, rewards are monotonically non-

#	D_i	β	Problem Type	Myopic Knapsack	Primal-Dual	Nested
1	3	0.1	Diminishing	66%	98%	97%
2	3	0.5	Diminishing	54%	87%	94%
3	3	0.9	Diminishing	41%	95%	94%
4	3	0.1	Monotonic	98%	100%	99%
5	3	0.5	Monotonic	83%	93%	90%
6	3	0.9	Monotonic	84%	90%	93%
7	3	0.1	Independent	97%	98%	98%
8	3	0.5	Independent	96%	95%	97%
9	3	0.9	Independent	84%	90%	94%
10	6	0.1	Diminishing	53%	59%	96%
11	6	0.5	Diminishing	48%	83%	91%
12	6	0.9	Diminishing	51%	75%	89%
13	6	0.1	Monotonic	64%	63%	99%
14	6	0.5	Monotonic	54%	59%	95%
15	6	0.9	Monotonic	74%	83%	99%
16	6	0.1	Independent	56%	57%	92%
17	6	0.5	Independent	47%	58%	96%
18	6	0.9	Independent	54%	76%	98%

Table 2.8: General Setting: Performance as a percentage of optimality bound.

decreasing in degree, $R_i^a(s_i) \geq R_i^b(s_i) \forall a \geq b$.

- Diminishing: Rewards are monotonically non-decreasing in degree under decreasing marginal returns.

As expected, the myopic knapsack policy is the worst performing method and tends to perform its best relative to the other methods when future rewards are heavily discounted (β low). The Nested Policy heuristic shows resilience to changing problem parameters and types, always achieving at least 89% of the optimality bound. As a carryover from the restless setting, the generalized primal-dual method performs comparable to the Nested Policy heuristic when the degree of each arm is low, but its performance suffers greatly as this degree is increased. The Nested Policy heuristic achieves a 95% overall average of the optimal bound over all instances of both degree 3 and 6 whereas the overall performance of the generalized primal-dual methods drops from 94% to 66% moving from 3 to 6. This is consistent with expectations as the generalized primal dual method is an index-based heuristic that relies heavily on a Whittle relaxation of the coupling budget to hold in expectation over time rather than with probability 1 in each time period. The work of Whittle [55] and Weiss and Weber [52] posits that index policies based on this type of relaxation perform well asymptotically as a number of indexable identical restless bandit arms (and total budget) is scaled appropriately due to Law of Large numbers type arguments. When we scale the number of different actions, taking increasing amounts, of budget available for each arm, we increase in general the variance of a particular policy on the generalized restless bandit, theoretically causing slower convergence to good performance of index or index-based heuristics.

2.6 Concluding Remarks

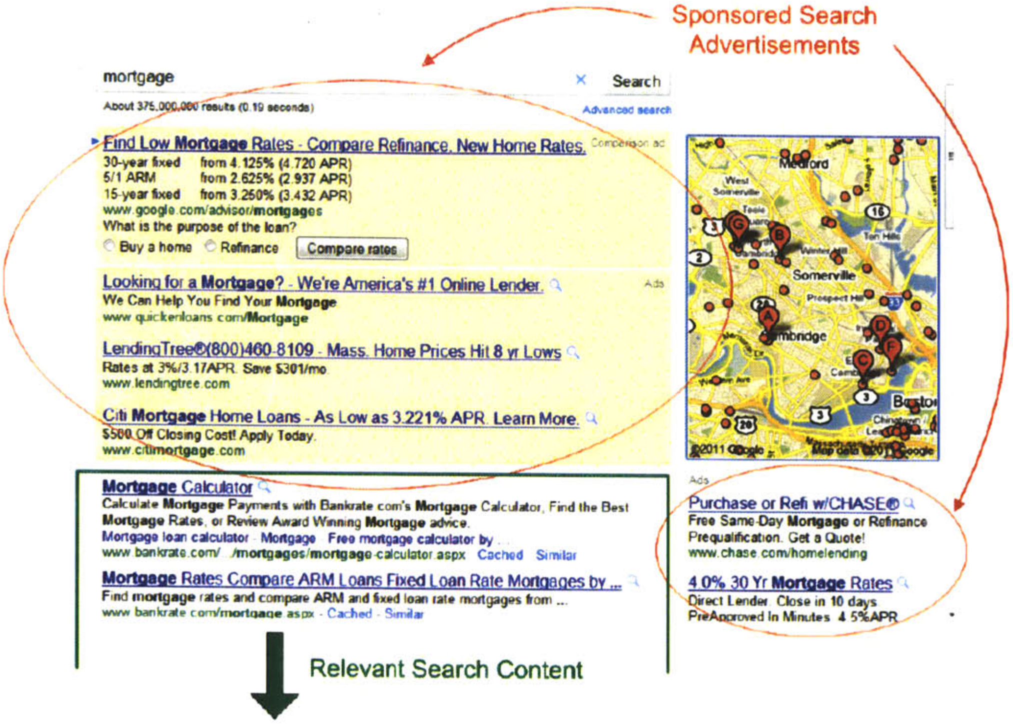
We have introduced a generalization of the restless bandit problem (GRBs) motivated by a broad range of applications. To solve such GRBs we have proposed a Nested Policy approach that generates a feasible policy exploiting our ability to solve smaller subproblems optimally. We have also provided methods to bound the objective for

GRBs and thus provide an a posteriori guarantee on the suboptimality of the Nested Policy heuristic. Our computational results show that the Nested Policy heuristic performs on par with other well established heuristics on restricted GRBs and is not too far from optimal in both restricted and general settings. Furthermore, this work demonstrates the viability of decomposition methods and integer-optimization based clustering methods in solving stochastic optimization problems.

Chapter 3

Sponsored Search Optimization Application

Internet search engine companies generate revenue through the sale of sponsored search advertising. When a particular search keyword or phrase is queried by an end user, in addition to being shown relevant internet search content the user is exposed to several advertisements related to the search as shown in Figure 3-1. A bidding process is used by potential advertisers to determine which ads are displayed to a user with each search query. In this process, each competing advertiser submits a bid level for each candidate keyword or phase along with a total daily, weekly or monthly budget. Each time an end-user search query is submitted, this information is used by the search engine company to rank ads, which are then displayed in rank-order to the user on the search results page. Advertisers do not pay for these impressions, but rather when and if the user actually clicks on an ad and is directed to the advertisers' website. Competing advertisers are not aware of each other's bids but usually have access to some historical data from the search engine company showing the search popularity of each term. The ranking process used by the search engine company is in general a black-box to advertisers. The amount actually paid for a user click-through is typically equal to the bid amount of the next-highest-ranked bid plus some nominal increment and does not exceed the advertiser's own bid level.



mortgage

About 375,000,000 results (0.19 seconds)

Find Low Mortgage Rates - Compare Refinance, New Home Rates. Comparison ad
30-year fixed from 4.125% (4.720 APR)
5/1 ARM from 2.625% (2.937 APR)
15-year fixed from 3.250% (3.432 APR)
www.google.com/advisor/mortgages
What is the purpose of the loan?
○ Buy a home ○ Refinance

Looking for a Mortgage? - We're America's #1 Online Lender. Ads
We Can Help You Find Your Mortgage
www.quickenloans.com/Mortgage

LendingTree®(800)460-8109 - Mass. Home Prices Hit 8 yr Lows
Rates at 3%/3.17APR. Save $301/mo.
www.lendingtree.com

Citi Mortgage Home Loans - As Low as 3.221% APR. Learn More.
$500 Off Closing Cost! Apply Today.
www.citimortgage.com

Mortgage Calculator
Calculate Mortgage Payments with Bankrate.com's Mortgage Calculator, Find the Best Mortgage Rates, or Review Award Winning Mortgage advice.
Mortgage loan calculator - Mortgage - Free mortgage calculator by ...
www.bankrate.com/.../mortgages/mortgage-calculator.aspx Cached Similar

Mortgage Rates Compare ARM Loans Fixed Loan Rate Mortgages by ...
Find mortgage rates and compare ARM and fixed loan rate mortgages from ...
www.bankrate.com/mortgage.aspx - Cached - Similar

Ads
Purchase or Refi w/CHASE®
Free Same-Day Mortgage or Refinance Prequalification. Get a Quote!
www.chase.com/homelending

4.0% 30 Yr Mortgage Rates
Direct Lender. Close in 10 days
PreApproved In Minutes. 4.5%APR

Figure 3-1: Sponsored search advertising.

In this chapter, we will focus on the problem an individual advertiser faces of maximizing search-user exposure to their website given a fixed daily budget. This problem can be conveniently modeled as a modification of the Generalized Restless Bandits presented in Chapter 2 where keywords and phrases are represented by bandit arms, different bid levels are represented by different degrees of play for each arm, and uncertainty in the bid response and the black-box nature of rankings are captured in the underlying Markov chains. We then implement and test this model using real-world data from an online retailer.

3.1 The advertiser's problem

The problem faced by an advertiser is to maximize total exposure to its website from search user traffic. The advertiser is faced with hundreds or thousands of keywords, phrases, and groups of words or phrases that it can bid on. Each bid will compete with those of other advertisers for exposure to search users and prices the cost of user click-through. Higher bids can generally lead to more impressions on the customer as well as higher ad rankings which can increase the likelihood a user will click on the ad; but with higher bids each customer click consumes a larger fraction of the underlying advertising budget. Advertiser's typically set a weekly budget and are free to change their bid levels for each word or phrase each week, receiving feedback in the form of the prior week's performance

The primitives of this problem are the total weekly budget M and the bid level associated with keyword in each week t, $b_i(t)$. Feasible bid levels are drawn from a discrete set $\mathcal{B}$, typically in small increments, such as 5 cents. Over the period of a week, ads are displayed to customers as a result of search queries and result in a total number of customer impressions $Imp_i(b_i(t))$; customers have an aggregate propensity to click on an ad which will depend on the ad's placement on the search page which is a black-box function of the advertiser's bid. We call the resulting click-through rate (number of clicks divided by number of impressions) $CTR_i(b_i(t))$. Each click will result in an average cost to the advertiser $C_i(b_i(t))$ which depends on the unknowns of ad ranking and competing bids for the keyword in addition to the advertiser's bid. The total weekly cost incurred by the advertiser for a particular keyword or phrase is then:

$$Cost_i(b_i(t)) = C_i(b_i(t)) \cdot Imp_i(b_i(t)) \cdot CTR_i(b_i(t)).$$

The total weekly exposure gained from a particular keyword or phrase is given by:

$$Clicks_i(b_i(t)) = Imp_i(b_i(t)) \cdot CTR_i(b_i(t)).$$

The advertiser's objective is to maximize the total exposure to its website while keeping expenses below the advertising budget. Typically exposure today is worth more than exposure tomorrow, so future exposure is discounted at some rate β^t. Advertisers have access to their own historical bidding behavior and performance; while the true responses of $C_i(b_i(t)), Imp_i(b_i(t))$, and $CTR_i(b_i(t))$ are unknown results

Search terms	$\mapsto$	Arms (i)
Weekly budget	$\mapsto$	Budget (M)
Bid level	$\mapsto$	Degree (d_i)*
Estimate of response	$\mapsto$	State (s_i)
Total click-through	$\mapsto$	Reward $R_i^{d_i}(s_i)$
Uncertainty	$\mapsto$	Transitions $P_i^{d_i}(s_i' \to s_i)$

*Through budget consumption that bid level leads to.

Table 3.1: Mapping sponsored search to a GRB.

of search engine ranking and user behavior, advertisers can hold estimates of these response functions based on historical cost-per-click, number of impressions, and click-through-rate observed at different bid levels. Advertisers generally have sparse data available with which to estimate keyword performance as a function of bid level. With historical data available, only a handful of bid levels may have been tested on each word and a small number of impressions may lead to imprecise estimates of click-though rate. To address this problem, advertisers may cluster historical data for groups of keywords can lead to reliable estimates of desired metrics.

3.2 Modeling as a modified GRB

The mapping of the advertiser's sponsored search problem is outlined in Table 3.1. Given a collection of N candidate keywords or phrases, each such search term can be viewed as an arm of a slightly modified Generalized Restless Bandit problem. Since data for individual search terms can be sparse, terms that have exhibited similar responses to various bid levels can be clustered together, giving a richer picture of how a particular term's response function may change over time. The state space of each arm $i = 1, ..., N$ then consists of these possible search term clusters $s \in S$, representing advertiser's estimate of the current response function for the term. As time passes, media exposure, user behavior and advertiser pricing are in constant flux, causing terms to change clusters, from s to s' according to a bid-dependent transition probability matrix $P^{bid}(s \rightarrow s')$ which can be calibrated from publicly available historical data in conjunction with the advertiser's personal historical data. The advertiser's own personal historical data can be used to estimate its own cost and success with search terms in a particular clusters s, $Cost_s(bid)$ and $Clicks_s(bid)$ for various bid levels. The advertiser's weekly budget M is consumed by the cost of clicks. The modification we make to the GRB problem of Chapter 2 is that the entire budget does not have to be consumed, but rather in each time period the budget cannot be exceeded. Note that the fundamental actions available (keyword bids) consume differing amounts of budget in different states. For example, a bid of 5 cents on a popular, high priced word will likely lead to a very low ranking, generating no clicks and zero cost; however, the same 5 cent bid on a less popular word may generate 10 clicks in a week resulting in a 50 cent charge. Thus the feasible degrees of play that need to be considered in order to be modeled as a GRB are the potential amount of budget consumed:

$$\mathcal{D} = \{Cost_s(bid) : s \in \mathcal{S}, bid \in \mathcal{B}\}$$

where $\mathcal{B}$ is the set of feasible bid levels. To map a policy in terms of this degree set to real-world bid levels for a keyword in a particular state, we then simply find the bid level for that state that consumes closest to that degree amount without exceeding it. Since the total budget consumption constraint is relaxed, an optimizer is then allowed to select any bid level that does not exceed a parcel of budget consumption. Finally, we can specify the transition probability matrix and reward for each state in terms of this degree set as follows:

$$\begin{aligned} bid(d,s) &= argmax_{bid \in \mathcal{B}}\{Cost_s(bid) : Cost_s(bid) \leq d\},\ d \in \mathcal{D}, s \in \mathcal{S}, \\ P^d(s \rightarrow s') &= P^{bid(d,s)}(s \rightarrow s'), \\ R^d(s) &= Clicks_s(bid(d,s)). \end{aligned}$$

3.2.1 Simplification of arm aggregation and arm substitution

Note that in this GRB model for sponsored search, the bandit contains N identical copies of a single arm. Recall the Arm Aggregation step in the Nested Policy Algorithm 1 of Chapter 2. In this step, we examine all possible partitioning of the arms into pairs and solve an integer optimization problem to find the partitioning that yields the best performing Whittle-type relaxation. However, in the case where the underlying arms are identical, all such partitions yield the same relaxation. Thus, the Arm Aggregation step is trivially solved by selecting an arbitrary partitioning into pairs.

Additionally, since each atomic arm is identical, as we progress through the Nested Policy Algorithm, each aggregated arm we encounter is simply a representation of some aggregated number, m of atomic arms. Thus, we only need to solve the Arm Substitution problem of the Nested Policy Algorithm 1 of Chapter 2 once for each $m = 2, ..., N$ throughout the entire progressing of the Nested Policy Algorithm.

3.2.2 Simplification of state clustering

Consider the state clustering problem in the Nested Policy Algorithm 1 of Chapter 2 which solves a binary optimization model to approximate an aggregate arm on a large product state space by a representative arm on a smaller state space applied to the sponsored search problem:

$$\max_{\varphi,x,\psi} \sum_{i\in\mathcal{A}_{\mathcal{G}^*}} \sum_{d=0}^{M} \sum_{s_i\in S_i} R_i^d(s_i)x_i^d(s_i) \tag{3.1}$$

$$s.t. \sum_{d=0}^{D_i} x_i^d(s_i) = \alpha_{s_i} + \beta \sum_{s_i'\in S_i} \sum_{d=0}^{D_i} P_i^d(s_i' \to s_i)x_i^d(s_i'), \forall i \in \mathcal{A}_{\mathcal{G}^*}, s_i \in S_i, \tag{3.2}$$

$$\sum_{i\in\mathcal{A}_{\mathcal{G}^*}} \sum_{s_i\in S_i} \sum_{d=0}^{D_i} dx_i^d(s_i) \leq \frac{M}{1-\beta}, \tag{3.3}$$

$$\sum_{d=0}^{D_i} \psi_i^d(\tilde{s}_i) \leq 1, \forall i \in C, \, \tilde{s}_i \in \tilde{S}_i, \tag{3.4}$$

$$z_i^d(s_i, \tilde{s}_i) \leq U\varphi(s_i, \tilde{s}_i), \forall i \in C, \, s_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \ldots, M, \tag{3.5}$$

$$x_i^d(s_i) - z_i^d(s_i, \tilde{s}_i) \leq U(1 - \varphi(s_i, \tilde{s}_i)), \forall i \in C, \, s_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \ldots, D_i, \tag{3.6}$$

$$\sum_{s_i \in S_i} z_i^d(s_i, \tilde{s}_i) \leq U\psi_i^d(\tilde{s}_i), \forall i \in C, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \ldots, D_i, \tag{3.7}$$

$$0 \leq z_i^d(s_i, \tilde{s}_i) \leq x_i^d(s_i), \forall i \in C, \, s_i \in S_i, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \ldots, D_i, \tag{3.8}$$

$$\sum_{\tilde{s}_i \in \tilde{S}_i} \varphi(s_i, \tilde{s}_i) = 1, \forall i \in C, \, s_i \in S_i, \tag{3.9}$$

$$x_i^d(s_i) \geq 0, \forall i \in \mathcal{A}_{\mathcal{G}^*}, \, s_i \in S_i, \, d = 0, \ldots, D_i, \tag{3.10}$$

$$\psi_i^d(\tilde{s}_i) \in \{0, 1\}, \forall i \in C, \, \tilde{s}_i \in \tilde{S}_i, \, d = 0, \ldots, D_i \tag{3.11}$$

$$\varphi(s_i, \tilde{s}_i) \in \{0, 1\}, \forall i \in C, \, s_i \in S_i, \tilde{s}_i \in \tilde{S}_i. \tag{3.12}$$

Where we have implemented the modification of not exceeding the period budget rather than consuming the entire budget in (3.3).

Since the Arm Substitution step of each iteration of the the Nested Policy Algorithm, we are left with a collection of aggregate arms each representing identical copies of an aggregation of some number m of atomic arms. Thus the rewards and transition probability matrices for each arm are identical and conservation of flow constraints (3.2) for each arm differ only in the initial distribution $\alpha_i(s_i)$. Thus the constraints:

$$\sum_{d=0}^{D} x^d(s) = \bar{\alpha}_s + \beta \sum_{s' \in S} \sum_{d=0}^{M} P^d(s' \to s) x^d(s'), \, s \in A(S)$$

are valid where $A(s)$ represents the current set of aggregated arms. These constraints arise from averaging the conservation of flow constraints (3.2) over each arm i for each state in of the aggregated atomic arm state space $s \in A(S)$. Here $x^d(s)$ represents the expected discounted time-percentage of atomic bandits is in state (s). $\bar{\alpha}_s$ represents the initial distribution over all atomic bandits in state (s). Since all of the $\bar{N} = \mathcal{A}_{\mathcal{G}^*}$ arms in this step are identical and have the same $\bar{D} = D_i$, if we enforce that the same clustering scheme be used for each arm, we can then simplify the State Clustering problem by clustering over $x^d(\boldsymbol{s})$ with the following:

$$
\begin{aligned}
&\max_{\varphi,\boldsymbol{x},\boldsymbol{\psi}} \sum_{d=0}^{\bar{D}} \sum_{\boldsymbol{s}\in A(S)} \bar{N} \cdot R^d(\boldsymbol{s}) x^d(\boldsymbol{s}) && (3.13)\\
&s.t. \sum_{d=0}^{\bar{D}} x^d(\boldsymbol{s}) = \bar{\alpha}_{\boldsymbol{s}} + \beta \sum_{\boldsymbol{s}'\in A(S)} \sum_{d=0}^{\bar{D}} P^d(\boldsymbol{s}' \to \boldsymbol{s}) x^d(\boldsymbol{s}'),\ \boldsymbol{s} \in AS, && (3.14)\\
&\sum_{\boldsymbol{s}\in A(S)} \sum_{d=0}^{\bar{D}} \bar{N} \cdot d x^d(\boldsymbol{s}) \le \frac{M}{1-\beta}, && (3.15)\\
&\sum_{d=0}^{\bar{D}} \psi^d(\tilde{s}) \le 1,\ \tilde{s} \in \tilde{S}, && (3.16)\\
&z^d(\boldsymbol{s},\tilde{s}) \le U\varphi(s,\tilde{s}),\ \forall\, \boldsymbol{s} \in A(S),\ \tilde{s} \in \tilde{S},\ d = 0,\dots,\bar{D}, && (3.17)\\
&x^d(\boldsymbol{s}) - z^d(\boldsymbol{s},\tilde{s}) \le U(1 - \varphi(\boldsymbol{s},\tilde{s})),\ \boldsymbol{s} \in S,\ \tilde{s} \in \tilde{S},\ d = 0,\dots,\bar{D}, && (3.18)\\
&\sum_{\boldsymbol{s}\in A(S)} z^d(s,\tilde{s}) \le U\psi^d(\tilde{s}),\ \forall\, \tilde{s} \in \tilde{S},\ d = 0,\dots,\bar{D}, && (3.19)\\
&0 \le z^d(\boldsymbol{s},\tilde{s}) \le x^d(\boldsymbol{s}),\ \forall\, \boldsymbol{s} \in A(S),\ \tilde{s}_i \in \tilde{S}_i,\ d = 0,\dots,\bar{D}, && (3.20)\\
&\sum_{\tilde{s}\in\tilde{S}} \varphi(\boldsymbol{s},\tilde{s}) = 1,\ \forall\, \boldsymbol{s} \in A(S),\\
&x^d(\boldsymbol{s}_i) \ge 0,\ \forall\, i \in \mathcal{A}_{\mathcal{G}^*},\ \boldsymbol{s}_i \in A(S),\ d = 0,\dots,\bar{D},\\
&\psi^d(\tilde{s}) \in \{0,1\},\ \forall\, \tilde{s} \in \tilde{S},\ d = 0,\dots,\bar{D}\\
&\varphi(\boldsymbol{s},\tilde{s}) \in \{0,1\},\ \forall\ \boldsymbol{s} \in A(S), \tilde{s} \in \tilde{S}_i.
\end{aligned}
$$

where the left-hand-side of (3.15) has been appropriately weighted to capture the total budget being consumed by arms being played to a particular degree in a particular state.

3.3 Implementation on real-world data

We implemented the model described in the previous section on a data set provided to us by a real-world online retailer of herbal supplements. The data set included 6 months of daily search data on 1,931 keywords with feasible bid levels predominantly in 5 cent increments from \$0.05 – \$0.50 but extending up to \$2 for certain words. During the 6 month period, the retailer had experimented with various bid levels for different words and spent an average of \$2,231.55 out of a \$2,250 weekly budget to generate an average of 12,074 clicks for an average cost per click of 18.5 cents. Due to the sparse data available for each search term at differing bid levels, we clustered terms according to similar historical performance and behavior. Using the provided data as well as data available from Google Analytics [53] on the overall search popularity of the words, we clustered keywords into 10 states on the attributes of search popularity and the bid amount the online retailer had learned had to be placed to achieve various top search rankings. Examining these attributes at cross sections of time allowed us to see how frequently search terms moved between clusters. Since the retailer's learning about bid→rank response fundamentally depended on its bid level for a certain word, we obtained the action dependant transition matrix $P^d(s \to s')$. Varying bid levels for words in each cluster that yielded clicks consumed on average between \$0.03 and \$304.62[1]; we approximated this degree set $\mathcal{D}$ by 5 cent increments from \$0.00 – \$1.00, 25 cent increments from \$1 – \$10 and 1 dollar increments from \$10 – \$305. Overall we obtained a GRB system with 1,931 arms on 10 states with this degree set and a total period budget of \$2,250. Table 3.2 shows the results of 2 years simulated performance of this system, that is simulating the resulting bandit system for 104 time periods; in each time period, the state of the current state of the bandit system is observed and a degree is selected for each arm yielding a transition to a new state in the new period. Results are shown transformed to time-average performance for direct comparison with the historical baseline. The Nested Policy heuristic is able to achieve an 81% increase in average weekly clicks over the historical baseline, attaining

[1] As an example, a word with an average cost-per-click of \$0.01 with an average of 3 clicks per week consumes \$0.03 of weekly budget; whereas a word with an average cost-per-click of \$0.50 and an average of 610 clicks per week consumes \$305 of weekly budget

91.73% of the first order optimality bound by driving the cost per click down to 10.23 cents outperforming both the myopic knapsack algorithm and generalized primal-dual approach from Chapter 2, Section 2.3.1.

Method	Average Weekly Clicks	Cost per Click	Budget Utilization
Historical	12,074	\$0.185	89.26%
Myopic-Knapsack	13,899	\$0.161	89.52%
Primal-Dual	18,840	\$0.119	89.96%
Nested Policy	21,933	\$.1023	89.72%
Optimality Bound$\cdot(1-\beta)$	23,908	$\leq$ \$0.094	N/A

Table 3.2: Weekly performance in simulation of sponsored search GRB model.

3.4 Concluding Remarks

In this chapter, we demonstrated the feasibility of applying the Nested Policy approach to large-scale Generalized Restless Bandit models based on real world data. The superior performance of the Nested Policy compared to both the myopic knapsack and generalized primal-dual approaches was maintained in this large scale setting. Furthermore, we discussed simplifications to the steps of the Nested Policy Algorithm for large-scale problems with identical bandit arms to decrease computation time.

Chapter 4

Distributionally Robust Optimization Problems

In this chapter, we study the distributionally robust optimization problem with known mean, covariance and support information. This problem is particularly relevant for fixed income investing where certain statistics on the marginal performance of the underlying notes are known with relative certainty, but the dynamics which drive the joint distribution of large baskets of investments are unknown.

It is well known that incorporating support information over intervals along with known mean and covariance admits a reduction from matrix co-positivity [9] making the robust problem of interest $\mathcal{NP}$ – *hard*. Previous approaches [42, 34] to this problem involve using one efficiently solvable relaxation for the entire problem and applying the resulting policy. We propose an algorithm which uses a sequence of efficiently solvable relaxations to determine nested policies involving pairs of decision variables. Each of relaxations is tight in the projected dimension of each pair of interest.

An additional contribution from this chapter is a proof that non-convex quadratic optimization in n-dimensions over a box in two-dimensions is efficiently solvable.

This proof is a generalization of a result by Anstreicher and Burer [2] for $n = 2$ using simpler proof technology. This result ensures the tight nature of our projected relaxations.

4.1 Problem Definition

Distributionally robust optimization allows practitioners who face uncertainty to use statistics of an underlying distribution which can be readily estimated while being robust to the underlying dynamics and structure that is unknown. This is in contrast to stochastic optimization, where knowledge of the entire distribution must be assumed.

In the distributionally robust problem with known mean, covariance and support, we consider an optimization problem whose uncertain data is comprised of n random variables, $N_1 = \{1, ..., n\}$. We are given moment information for the joint distribution of these variables up to order 2; that is $\mathbb{E}\left[r_i^\alpha \cdot r_j^\beta\right] = \mu_{i,j}^{\alpha,\beta}$ for all $i \neq j, \alpha + \beta \leq 2$. Additionally, the support of all variables is bounded: $supp(r_i) = [a_i, b_i]$. The goal is to select a weighted exposure, w, to the underlying random variables in the context of maximizing a measure of worst-case expected utility. Given a decision maker's utility function and weighted exposure, worst-case expected utility can be viewed the expected utility of that exposure under a worst-case distribution μ selected by an adversary from among those distributions that share known characteristics.

4.2 Motivating example: fixed income portfolio optimization

Fixed income investing typically involves a purchase of, or loan originating, a debt obligation from an obligor to an investor. One fundamental property of this type of investing is that returns are naturally bounded between the face value of the obligation and total loss of investment or some ensured recovery rate thereof. Additionally several statistics are typically assumed or inferred with relatively high certainty about the return including the probability an individual obligor defaulting, the expected loss given default, and the probability of joint default for pairs of obligors. In order to value portfolios of fixed income investments what remains to be modeled are the dynamics of the distribution that governs default and loss distributions jointly for all obligors. Typical approaches in industry such as CreditMetrics [26] and Moody's KMV [13, 16] involve assuming an overarching copula model[1]. Such copulas models involve assuming there is some underlying parameterized distribution, typically Gaussian in nature. Model parameters are then chosen as to match the known marginal statistics. The problem with these models is that there is no intuition about the structure that these copulas should have and thus the models used are most often simply those with the most convenient parameterizations. Additionally, possibly one the most relevant information in fixed income portfolio valuation is the probability of simultaneous default of a large basket of obligors. Gaussian copula models by their very nature lead to light tails and recent history has shown that the use of such models severely underestimate the probability of such events [11, 29].

Robust optimization is an alternative to assuming a copula model for the joint distribution of dynamics. This approach allows us to use the statistics we know and be robust to the dynamics we don't. Rather than assuming a copula with little real world justification simply because it is easily parameterizable, robust optimization allows us to prepare for all possible distribution which would exhibit the known statistics.

4.2.1 Formulation

For the fixed income portfolio optimization problem, we are given a set of n assets $N_1 = \{1, ..., n\}$. We index the set N here so that we may alter the set of assets under consideration in further analysis. For each asset $i \in N_1$ it is common to be able to

[1] See Glasserman [21, 20] for examples of simulation involving copula models.

estimate the following with relative certainty [26, 13, 16]:

- Face value (b_i): the maximum return achievable for the asset if the obligor for the underlying asset does not default prior to maturity of the asset.
- Probability of default (p_i): the probability that the obligor for the underlying asset will default before maturity, leading to an overall return less than b_i.
- Expected loss given-default (lgd_i) : expected loss of face value given that the obligor on the underlying asset defaults before maturity.
- Variance of loss-given default (vgd_i) : variance of loss of face value given that the obligor on the underlying asset defaults before maturity.
- Minimum insured recovery amount (a_i) : minimum return achievable by the asset; this amount is often insured by outside derivative contracts, ensuring that losses from face value cannot cause total return to fall below this threshold.
- Probability of joint default (p_{ij}): the probability that the obligors for the underlying assets i and j will both default together. It is assumed that, due to the recovery process, losses given default are marginally independent for assets i and j.

If we consider each assets return as a random variable r_i we then have the mean, covariance, and support of r: $\mathbb{E}\left[r_i^\alpha \cdot r_j^\beta\right] = \mu_{i,j}^{\alpha,\beta}$ for all $i \neq j, \alpha + \beta \leq 2$ and $supp(r_i) = [a_i, b_i]$. Where the moments are given by:

$$\begin{aligned}
\mu_{i,\cdot}^{1,0} &= f_i - p_i \cdot lgd_i, && (4.1)\\
\mu_{i,\cdot}^{2,0} &= p_i \cdot \left(lgd_i^2 + vgd_i\right), && (4.2)\\
\mu_{i,j}^{1,1} &= p_{ij} \cdot \left(lgd_i \cdot lgd_j\right). && (4.3)
\end{aligned}$$

We wish to select a collection of weights of exposure to the assets that satisfy some linear constraints:

$$A_1 w_i \leq g \tag{4.4}$$

It is assumed that the constraints $\boldsymbol{w} \geq \mathbf{0}$ are included in this constraint set. The most common constraint set is to invest a in a total portfolio with unit weight $(\sum_i w_i = 1)$ while disallowing short-selling $\boldsymbol{w} \geq \mathbf{0}$. Other common constraints include ensuring a minimum or maximum exposure to a certain subset of assets, such as an industry sector.

Our goal is to maximize worst-case expected utility. The utility functions we study are piecewise quadratic, allowing us to directly capture Markowitz-type utility functions [33], and closely model more complex utility functions with piecewise approximation:

$$\min_u \left\{ q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u \right\}. \tag{4.5}$$

To ensure robustness to any distribution that matches known mean, covariance, and support information, we allow an adversary to select any probability distribution matching these characteristics by solving the robust optimization problem:

$Z^*(N_1) =$

$$\begin{array}{lll} \max\limits_{A_1 \boldsymbol{w} \leq \boldsymbol{b}} \inf\limits_{\mu(\boldsymbol{r}_{N_1})} & \int \min\limits_u \left\{ q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u \right\} d\mu(\boldsymbol{r}_{N_1}) & \\ \quad s.t. & \int r_i^\alpha r_j^\beta d\mu(\boldsymbol{r}_{N_1}) = \mu_{i,j}^{\alpha,\beta} & \forall i \neq j \in N_1, \alpha + \beta \leq 2, \\ & \int_{a_i \leq r_i \leq b_i} d\mu(\boldsymbol{r}_{N_1}) = 1 & \forall i \in N_1. \end{array} \tag{4.6}$$

Isii [28] and Smith [48] have shown that strong duality holds for the inner problem with the following polynomial optimization problem:

$$\begin{array}{ll} \max\limits_{A_1 \boldsymbol{w} \leq \boldsymbol{g}} \sup\limits_{\boldsymbol{y}} & y_0 + \sum\limits_{\alpha+\beta \leq 2} \mu_{i,j}^{\alpha,\beta} y_{i,j}^{\alpha,\beta} \\ \quad s.t. & q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 - \sum\limits_{i \neq j \in N_1, \alpha+\beta \leq 2} y_{i,j}^{\alpha,\beta} \left(r_i^\alpha r_j^\beta \right) + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u - y_0 \geq 0 \\ & \qquad \forall u, \boldsymbol{r} : a_i \leq r_i \leq b_i, i \in N_1. \end{array} \tag{4.7}$$

Given $\boldsymbol{w}, \boldsymbol{y}$, we have for each u the following separation problem:
$S_u(\boldsymbol{w}, \boldsymbol{y}) =$

$$\min_{\boldsymbol{r}:a_i \leq r_i \leq b_i, \in N_1} \quad q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 - \sum_{i \neq j \in N_1, \alpha+\beta \leq 2} y_{i,j}^{\alpha,\beta} \left(r_i^\alpha r_j^\beta \right) + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u - y_0 \geq 0 \tag{4.8}$$

Thus, each constraint demands coefficients of a quadratic that is non-negative over a hypercube in $\mathbb{R}^n$. The corresponding separation problem is $\mathcal{NP} - \textbf{\textit{hard}}$ as it admits a reduction from matrix co-positivity [9]. Previous approaches have focused on tractable relaxations which avoid full incorporations of mean, covariance and support simultaneously. The decomposition method we will describe in Section 4.4 motivates us to focus on sub-problems incorporating mean, covariance and support simultaneously; in Section 4.5.1 we show that such focused sub-problems are tractable.

4.3 Prior work

Distributionally robust optimization has been well studied when only moment information is known. Scarf [46] studied the application of optimization of the newsvendor problem under the worst case distribution having a known mean an variance. El Ghaoui et al. [17] studied robust optimization of the Value-at-Risk metric under mean and covariance information. Popescu [42] proposed a methodology based on parametric quadratic programming to handle more general utility functions when mean and covariance information is known. Bertsimas et al. [5, 6] and Natarajan et al. [34] show that when only first and second moment information or only first moment information is known along with support, the problem is tractable with concave piecewise linear objectives. Goh and Sim [25] study the variation with known mean and support and show that one can even incorporate adaptive decision rules to allow primal decisions to be anticipatory functions of adversarial uncertainty. Chen et al. [12] allow for the inclusion of information on directional derivatives. The full problem we study with known mean, covariance, and interval support was shown by Bertsimas and Popescu [9] to be $\mathcal{NP} - \textbf{\textit{hard}}$. Delage and Ye [14] propose a data-driven method where the distributional support is outer-approximated by a ball. This relaxation of the adversarial problem is then tractable. Natarajan et al. [34] exploit the fact that the problem with known mean and covariance is tractable alongside the problem of known mean and support in the context of portfolio optimization. Their approach convolves the two bounds available from these relaxations by essentially allowing an adversary to pick two distributions, one which matches the mean and covariance, and one which matches the mean and support. The primal decision-maker is then allowed to select on an asset-by-asset basis which distribution the uncertain return for that asset is drawn from.

4.4 Decomposition strategy

The decomposition strategy we propose involves solving a sequence of adversarial problem relaxations which have tight projections on two-variable dimensions. We then use the solutions to these relaxations to obtain a sub-policy with respect to the "tight" dimensions, that is we fix the proportional weighting in these two uncertain random variables. With this fixed sub-policy, we may replace the two individual random variables with a new one representing their fixed proportional weighting. In this way, we iteratively reduce the dimension of the overall problem.

We begin by picking two uncertain random variables, indexed by say $n-1$ and n. We then solve the following problem which relaxes the support constraints on variables $1, ..., n-2$:
$Z(N_1) =$

$$\begin{array}{lll} \max\limits_{A_1 w \le g} \inf\limits_{\mu(r_{N_1})} & \int \min\limits_{u} \left\{ q_u \cdot (w^T r)^2 + c_u \cdot w^T r + d_u \right\} d\mu(r_{N_1}) & \\ s.t. & \int r_i^{\alpha} r_j^{\beta} d\mu(r_{N_1}) = \mu_{i,j}^{\alpha,\beta} & \forall i \neq j \in N_1, \alpha + \beta \leq 2 \\ & \int_{a_i \leq r_i \leq b_i, i > n-2} d\mu(r_{N_1}) = 1 & \forall i \in N_1 \end{array} \tag{4.9}$$

By solving $Z(N_1)$ we obtain the desired weights w_{n-1} and w_n and fix our relative weighting policy between variables r_{n-1} and r_n.

Once this portion of our policy is fixed, we can consider a new random variable

$$r_{\overline{n-1}} = \frac{w_{n-1}r_{n-1} + w_n r_n}{w_{n-1} + w_n} \tag{4.10}$$

. We know that

$$supp(r_{\overline{n-1}}) \subseteq \left[\frac{w_{n-1}}{w_{n-1} + w_n} \cdot a_{n-1} + \frac{w_n}{w_{n-1} + w_n} \cdot a_n, \frac{w_{n-1}}{w_{n-1} + w_n} \cdot b_{n-1} + \frac{w_n}{w_{n-1} + w_n} \cdot b_n\right] \tag{4.11}$$

and we know that the moments of $r_1, r_2, ..., r_{n-2}, r_{\overline{n-1}}$ must obey the original moment constraints on $r_1, ..., r_{n-2}$ as well as:

$$\mathbb{E}\left[r_i^\alpha \cdot r_{\overline{n-1}}^\beta\right] = \frac{w_{n-1}}{w_{n-1} + w_n} \cdot \mu_{i,n-1}^{\alpha,\beta} + \frac{w_n}{w_{n-1} + w_n} \cdot \mu_{i,n}^{\alpha,\beta} = \mu_{i,\overline{n-1}}^{\alpha,\beta} \tag{4.12}$$

$$\forall i = 1, ..., n-2, \alpha + \beta \leq 2 \tag{4.13}$$

Thus, if we define the new set $N_2 = \{1, 2, ..., n-2, \overline{n-1}\}$ we can consider the distributionally robust optimization problem $Z(N_2)$ characterized by these constraints. The linear constraints on weight variables take the form:

$$A_2 w \leq g \tag{4.14}$$

where A_2 is formed by replacing columns n and $n-1$ ($A_1(n)$ and $A_1(n-1)$) with a single column:

$$A_2(\overline{n-1}) = \frac{w_{n-1}A_1(n-1) + w_n A_1(n)}{w_{n-1} + w_n} \tag{4.15}$$

We can then iterate this procedure combining random variables r_{n-2} and $r_{\overline{n-1}}$ to obtain a fixed policy between these them, yielding a new random variable $r_{\overline{n-2}}$ and likewise the set N_2, continuing until solving $Z(N_{n-1})$. In each iteration, we consider the problem:

$$
\begin{array}{lll}
Z(N_t) = & & \\
\max\limits_{A_t w \leq g} \inf\limits_{\mu(r_{N_t})} & \int \min\limits_{u} \left\{ q_u \cdot (w^T r)^2 + c_u \cdot w^T r + d_u \right\} d\mu(r_{N_t}) & \\
\qquad s.t. & \int r_i^{\alpha} r_j^{\beta} d\mu(r_{N_t}) = \mu_{i,j}^{\alpha,\beta} & \forall i \neq j \in N_t, \alpha + \beta \leq 2 \\
& \int_{a_i \leq r_i \leq b_i} d\mu(r_{N_t}) = 1 & i = n - t, \overline{n - t + 1}
\end{array}
\tag{4.16}
$$

More formally this algorithm can be stated as follows:

Algorithm 3 (Decomposition method for distributionally robust optimization). *Given a set of random variables $r_i, i \in N_1 = \{1, ..., n\}$ having known support $supp(r_i) = [a_i, b_i]$ and mean and covariance $\mathbb{E}\left[r_i^{\alpha} \cdot r_j^{\beta}\right] = \mu_{i,j}^{\alpha,\beta}$ for all $i \neq j, \alpha + \beta \leq 2$, and the problem of finding a feasible weighting $A_1 w \leq g$, decomposition is performed as follows:*

1. *FOR $t = 1$ to $n - 1$*

   (a) *Let $m := |N_t|$*

   (b) *Solve the problem $Z(N_t)$ (4.16) to obtain a weighting w^t.*

   (c) *Define a new random variable $r_{\overline{m-1}}$ with known mean, covariance an support as follows:*

$$
\begin{aligned}
supp(r_{\overline{m-1}}) &= \left[\frac{w_{m-1}^t \cdot a_{m-1} + w_m^t \cdot a_m}{w_{m-1}^t + w_m^t}, \frac{w_{m-1}^t \cdot b_{m-1} + w_m^t \cdot b_m}{w_{m-1}^t + w_m^t} \right], \\
\mathbb{E}\left[r_i^{\alpha} \cdot r_{\overline{m-1}}^{\beta} \right] &= \frac{w_{m-1}^t}{w_{m-1}^t + w_m^t} \cdot \mu_{i,m-1}^{\alpha,\beta} + \frac{w_m^t}{w_{m-1}^t + w_m^t} \cdot \mu_{i,m}^{\alpha,\beta} \\
&= \mu_{i,\overline{m-1}}^{\alpha,\beta}, \\
& \qquad \forall i = 1, ..., m - 2, \alpha + \beta \leq 2.
\end{aligned}
$$

   (d) *Define the matrix A_{t+1} by replacing columns m and $m - 1$ of A_t ($A_t(n)$ and $A_t(n - 1)$) with a single column:*

$$
A_{t+1}(\overline{n-1}) = \frac{w_{n-1} A_1(n-1) + w_n A_1(n)}{w_{n-1} + w_n} \tag{4.17}
$$

(e) $|N_{t+1}| = N_t \cup \{\overline{m-1}\} \backslash \{m-1, m\}$

2. *Let* $t^* = argmax_t Z(N_t)$

3. ***The final weight given to asset*** i *is then:*

$$w_i^* = \begin{cases} w_i^i \cdot \prod_{t<i}^{i-1} w_{\overline{|N_t|-1}}^t, & i \leq n - t^*, \\ w_i^{t^*}, & i > n - t^*. \end{cases} \tag{4.18}$$

There are also several possible extensions to this algorithm. Selecting which two assets to combine in each iteration can be done in a similar manner to the Arm Aggregation method discussed for Generalized Restless Bandits in 1 of Chapter 2 rather than simply combining the two with highest index. Additionally, if supplemental application-based information is known about the support of a joint distribution of any two uncertain random variables r_{m-1} and r_m, this information can be directly incorporated into the support of $r_{\overline{m-1}}$. In each iteration of the algorithm, we are tightening the support of two assets of the adversarial problem; however when we combine two assets with known support into a single asset with known support, we are slightly relaxing the adversarial problem since:

$$supp(r_{\overline{m-1}}) \subseteq \left[\frac{w_{m-1} \cdot a_{m-1}}{w_{m-1} + w_m} + \frac{w_m \cdot a_m}{w_{m-1} + w_m}, \frac{w_{m-1} \cdot b_{m-1}}{w_{m-1} + w_m} + \frac{w_m \cdot b_m}{w_{m-1} + w_m} \right],$$

and this inclusion need not be equality. Thus using such known real-world information on the joint support of r_{m-1} and r_m can tighten the bounds obtained.

Even without additionally joint support information, the decomposition algorithm can provide us bounds on performance for the full problem with known mean, covariance, and support. Let $Z_t(\boldsymbol{w}, \mu)$ denote the value of the policy $\boldsymbol{w}$ under distribution μ for asset set N_t and let $\mu^*(\boldsymbol{w})$ be the worst case worst case distribution for the overall problem with full support for a weighting of $\boldsymbol{w}$. That is:

$\mu^*(\boldsymbol{w}) =$

$$\begin{aligned} argmin_{\mu(\boldsymbol{r}_{N_1})} \quad & \int \min_u \left\{ q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u \right\} d\mu(\boldsymbol{r}_{N_1}) && \\ s.t. \quad & \int r_i^\alpha r_j^\beta d\mu(\boldsymbol{r}_{N_1}) = \mu_{i,j}^{\alpha,\beta} && \forall i \neq j \in N_1, \alpha + \beta \leq 2, \\ & \int_{a_i \leq r_i \leq b_i} d\mu(\boldsymbol{r}_{N_1}) = 1 && \forall i \in N_1. \end{aligned} \tag{4.19}$$

We then have the following:

Theorem 4.1 *$Z(N_t) \leq Z_t(\boldsymbol{w}^t, \mu^*(\boldsymbol{w}^t))$. That is, in each iteration of the decomposition algorithm, $Z(N_t)$ provides a lower bound on the true worst-case performance of the policy $\boldsymbol{w}^t$.*

Proof. Let μ_t be the worst-case distribution for $Z(N_t)$. Since

$$supp(r_{\overline{m-1}}) \subseteq \left[\frac{w_{m-1} \cdot a_{m-1}}{w_{m-1} + w_m} + \frac{w_m \cdot a_m}{w_{m-1} + w_m}, \frac{w_{m-1} \cdot b_{m-1}}{w_{m-1} + w_m} + \frac{w_m \cdot b_m}{w_{m-1} + w_m} \right],$$

$\mu^*(\boldsymbol{w}^t)$ is feasible for the adversarial problem in $Z(N_t)$. Thus $Z(N_t) = \bar{Z}_t(\boldsymbol{w}^t, \mu^t) \leq \bar{Z}_t(\boldsymbol{w}^t, \mu^*(\boldsymbol{w}^t))$. □

Corollary 4.2 *$Z(N_{t^*})$ provides the best achievable bound encountered in the decomposition algorithm. This bound is achievable by using policy $\boldsymbol{w}^*$.*

There is a natural interpretation of this decomposition strategy in the context of application to fixed income. In each iteration, we select two assets to discover a relative investment strategy. That is, given an amount of money to invest in assets (a) and (b), what percentage should be invested in asset (a) and what percentage in (b)? To answer this question, we solve $Z(N_t)$ which restricts the adversary as tightly as possible in the dimensions of these two assets. The solution, w^t, fixes a sub-policy by giving us the desired relative investment strategy. With proportions of investments fixed, we can then replace assets (a) and (b) in further problems by a representative portfolio.

It remains to be shown that the key problem $Z(N_t)$ is efficiently solvable, this is the focus of the following section.

4.5 Solution of the key separation problem

Recall the problem $Z(N_t)$ with two-variable support solved in each iteration of the decomposition algorithm:
$Z(N_t) =$

$$\begin{array}{lll} \max\limits_{A_t \boldsymbol{w} \leq \boldsymbol{g}} \inf\limits_{\mu(\boldsymbol{r}_{N_t})} & \int \min\limits_{u} \left\{ q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u \right\} d\mu(\boldsymbol{r}_{N_t}) & \\ \quad s.t. & \int r_i^\alpha r_j^\beta d\mu(\boldsymbol{r}_{N_t}) = \mu_{i,j}^{\alpha,\beta} & \forall i \neq j \in N_t, \alpha + \beta \leq 2 \\ & \int_{a_i \leq r_i \leq b_i} d\mu(\boldsymbol{r}_{N_t}) = 1 & i = n-t, \overline{n-t+1} \end{array} \tag{4.20}$$

In the same manner as the problem with full support, due to Isii [28] and Smith [48], we have that strong duality holds for the inner problem with the following polynomial optimization problem:

$$\begin{array}{rl} Z(N_t) = \max\limits_{A_t \boldsymbol{w} \leq \boldsymbol{b}} \sup\limits_{\boldsymbol{y}} & y_0 + \sum\limits_{\alpha+\beta \leq 2} \mu_{i,j}^{\alpha,\beta} y_{i,j}^{\alpha,\beta} \\ s.t. & q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 - \sum\limits_{i \neq j \in N_1, \alpha+\beta \leq 2} y_{i,j}^{\alpha,\beta} \left(r_i^\alpha r_j^\beta \right) + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u - y_0 \geq 0 \\ & \forall u, \boldsymbol{r} : a_i \leq r_i \leq b_i, i = n-t, \overline{n-t+1}. \end{array} \tag{4.21}$$

Here each constraint demands coefficients of a quadratic in n-dimensions that is non-negative over a box in two-dimensions. Unlike the case where the quadratic must be non-negative over a hypercube in $\mathbb{R}^n$ which admits an $\mathcal{NP} - hard$ separation problem, we will show that the corresponding separation problem here is efficiently solvable.

Given $\boldsymbol{w}, \boldsymbol{y}$, we have for each u the following separation problem:
$S_u(\boldsymbol{w}, \boldsymbol{y}) =$

$$\min_{\boldsymbol{r}: a_i \leq r_i \leq b_i, i = n-t, \overline{n-t+1}} \quad q_u \cdot (\boldsymbol{w}^T \boldsymbol{r})^2 - \sum_{i \neq j \in N_1, \alpha + \beta \leq 2} y_{i,j}^{\alpha,\beta} \left(r_i^{\alpha} r_j^{\beta} \right) + c_u \cdot \boldsymbol{w}^T \boldsymbol{r} + d_u - y_0 \geq 0. \tag{4.22}$$

Normalizing the support of r_{n-t} and $r_{\overline{n-t+1}}$ to fall between 0 and 1, we obtain the equivalent problem:

$$\begin{aligned} S_u(\boldsymbol{w}, \boldsymbol{y}) = \min \quad & \begin{pmatrix} 1 \\ \boldsymbol{r} \end{pmatrix}^T Q_u(\boldsymbol{w}, \boldsymbol{y}) \begin{pmatrix} 1 \\ \boldsymbol{r} \end{pmatrix} \\ s.t. \quad & 0 \leq r_{n-t}, r_{\overline{n-t+1}} \leq 1. \end{aligned} \tag{4.23}$$

The semi-definite relaxation [10] (SDP relaxation) to this problem is given by:

$$\begin{aligned} \bar{S}_u(\boldsymbol{w}, \boldsymbol{y}) = \min \quad & \mathit{trace}\left(R^T Q_u(\boldsymbol{w}, \boldsymbol{y}) \right) \\ s.t. \quad & R \succeq 0, \\ & r_{11} = 1, \\ & 0 \leq r_{1,n-t}, r_{1,\overline{n-t+1}} \leq 1. \end{aligned} \tag{4.24}$$

where $R \succeq 0$ demands that the matrix R be positive semi-definite. We will show that by adding a small number of linear constraints to $\bar{S}_u(\boldsymbol{w}, \boldsymbol{y})$ we obtain an exact formulation of $S_u(\boldsymbol{w}, \boldsymbol{y})$, thus yielding an efficiently solvable problem.

4.5.1 On quadratic optimization in n dimensions over a two-dimensional box

In this section, we consider the problem:

$$\begin{aligned} \min \quad & \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}^T Q \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \\ s.t. \quad & 0 \leq x_1, x_2 \leq 1. \end{aligned}$$

with $\boldsymbol{x} \in \mathbb{R}^n$. Where we have the box constraints:

$$\begin{aligned} x_1 &\leq 1, && (4.25)\\ 0 &\leq x_1, && (4.26)\\ x_2 &\leq 1, && (4.27)\\ 0 &\leq x_2. && (4.28) \end{aligned}$$

Taking the SDP relaxation of the problem we consider the matrix:

$$Y = \begin{pmatrix} 1 & x_1 & x_2 & \boldsymbol{x}_3^T \\ x_1 & x_{11} & x_{12} & X_{13}^T \\ x_2 & x_{12} & x_{22} & X_{23}^T \\ \boldsymbol{x}_3 & X_{13} & X_{23} & X_{33} \end{pmatrix},$$

and applying the reformulation-linearization technique (taking the first Sherali-Adams closure) [47], multiplying each pair of upper and lower bound constraints together, we obtain:

$$\begin{aligned} Y &\succeq 0, && (4.29)\\ y_{11} &= 1, && (4.30)\\ x_1 - x_{11} &\geq 0, && (4.31)\\ x_2 - x_{22} &\geq 0, && (4.32)\\ x_1 - x_{12} &\geq 0, && (4.33)\\ x_2 - x_{12} &\geq 0, && (4.34)\\ x_{12} &\geq 0, && (4.35)\\ x_{12} - x_1 - x_2 + 1 &\geq 0. && (4.36) \end{aligned}$$

Let $\mathcal{Y} = \{Y : (4.29) - (4.36)\}$ and consider the region:

$$\mathcal{D} = conv\left\{ \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : 0 \leq x_1, x_2, \leq 1 \right\}.$$

Theorem 4.3 $\mathcal{Y} = \mathcal{D}$.

This theorem generalizes a result of Anstreicher and Burer [2] for $n = 2$ using simpler proof technology. This result ensures that if we relax the support constraints for all but two variables, the problem can be solved exactly and efficiently.

Note that by construction $\mathcal{Y} \supseteq \mathcal{D}$. It remains to prove $\mathcal{Y} \subseteq \mathcal{D}$.

Definition 4.4 (Face) *Let S be a convex set. A convex subset F of S is called a face of S if $x \in F, y, z \in S$ where x is a strict convex combination of y and z implies that y and z are in F. An extreme point of S is a non-trivial zero dimensional face. An extreme direction of S is a one-dimensional face.*

It is well known that a closed convex set S is the convex hull of its extreme points and extreme directions[2]. We will show that all extreme points of $\mathcal{Y}$ are rank one matrices of the desired form and that $\mathcal{Y}$ has no extreme directions, thus proving $\mathcal{Y} \subseteq \mathcal{D}$.

Our strategy for proving Theorem 4.3 is summarized as follows:

1. We show that the rank of an extreme matrix of an SDP can be related to the number of linear constraints it satisfies with equality (Theorem 4.5).

2. We then show that the upper-left 3×3 sub-matrix of any extreme matrix of $\mathcal{Y}$ is rank 1 (Theorem 4.6).

3. Next we show that this implies that all extreme matrices of $\mathcal{Y}$ is rank 1 (Corollary 4.7).

[2] see Rockefellar [43] Theorem 18.5

4. Finally, we show that $\mathcal{Y}$ has no extreme directions, in other words, it is 'curved' (Theorem 4.8).

The following theorem is adapted from Pataki [39]:

Theorem 4.5 *For any face F of the feasible region of an SDP decribed by:*

$$\begin{aligned} Y &\succeq 0, && (4.37)\\ tr(A_i \cdot Y) &= b_i \; i = 1, ..., m, && (4.38)\\ tr(C_j \cdot Y) &\geq d_j \; j = 1, ..., p. && (4.39) \end{aligned}$$

if J is the index set of inequality constraints binding on F, then for $Y \in F$:

$$\frac{1}{2} rank(Y) \cdot (1 + rank(Y)) \leq m + |J| + dim(F). \tag{4.40}$$

Proof. Let $r = rank(Y)$. Since $Y \succeq 0$, by LDL decomposition, we can write:

$$Y = Q \Lambda Q^T,$$

with $Q \in \mathbb{R}^{nxr}, \Lambda \in \mathbb{S}^r, \Lambda \succ 0, diagonal$. We then have:

$$\begin{aligned} tr(Q^T A_i Q \cdot \Lambda) &= b_i, \; i = 1, ..., m,\\ tr(Q^T C_j Q \cdot \Lambda) &= d_j, \; j \in J. \end{aligned}$$

For the sake of contradiction, assume $\frac{1}{2} r(r+1) > m + |J| + dim(F)$. Note that the linear system above is defined by $m + |J|$ equality constraints and that $dim(\mathbb{S}^r) = \frac{1}{2} r(r+1)$. Thus there exist $\Lambda_1, ..., \Lambda_{dim(F)+1} \in \mathbb{S}^r$ linearly independent matrices in the null space of $Q^T A_i Q$ and $Q^T C_j Q, \; j \in J$:

$$\begin{aligned} tr(Q^T A_i Q \cdot \Lambda_k) &= 0, \; i = 1, ..., m, k = 1, ..., dim(F) + 1,\\ tr(Q^T C_j Q \cdot \Lambda_k) &= 0, \; j \in J, k = 1, ..., dim(F) + 1. \end{aligned}$$

Since $\Lambda \succ 0$ and $tr(Q^T C_j Q \cdot \Lambda) > d_j$, $j \notin J$, $\exists \epsilon > 0$:

$$\begin{aligned} \Lambda \pm \epsilon \Lambda_k &\succeq 0, \; k = 1, ..., dim(F) + 1, \\ \left(Q^T C_j Q \cdot (\Lambda \pm \epsilon \Lambda_k)\right) &\geq d_j \; j \notin J, k = 1, ..., dim(F) + 1. \end{aligned}$$

For all k, the matrices $Y_k^+ = Q(\Lambda + \epsilon \Lambda_k) Q^T$ and $Y_k^- = Q(\Lambda + \epsilon \Lambda_k) Q^T$ are in the feasible set of the SDP and $Y = \frac{1}{2}(Y_k^+ + Y_k^-)$. Thus $Y_k^+, Y_k^- \in F$ $k = 1, ..., dim(F) + 1$. However, since the matrices Λ_k are linearly independent, the matrices $\Lambda, \Lambda_1, ..., \Lambda_{dim(F)+1}$ are affinely independent and thus $Y, Y_1^+, ..., Y_{dim(F)+1}^+ \in F$ are affinely independent. Thus $dim(F) \geq dim(F) + 1$ which is a contradiction. □

We will now proceed to prove that the extreme points of $\mathcal{Y}$ are rank 1 matrices.

Theorem 4.6 *Let F be a zero-dimensional face of $\mathcal{Y}$. Then for $\bar{Y} \in F$ with*

$$\bar{Y} = \begin{pmatrix} 1 & x_1 & x_2 & x_3^T \\ x_1 & x_{11} & x_{12} & X_{13}^T \\ x_2 & x_{12} & x_{22} & X_{23}^T \\ x_3 & X_{13} & X_{23} & X_{33} \end{pmatrix},$$

the submatrix:

$$Y = \begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & x_{11} & x_{12} \\ x_2 & x_{12} & x_{22} \end{pmatrix},$$

is rank 1.

Proof. We have $rank(Y) \leq rank(\bar{Y})$, applying Theorem 2, we obtain:

$$\frac{1}{2} rank(Y) \cdot (1 + rank(Y)) \leq |J| + 1, \tag{4.41}$$

where J is the set of (4.31)-(4.36) binding on F. Note that $rank(Y) \neq 0$ since $y_{11} = 1$. We examine 3 cases.

Case 1: $|J| \leq 1$. Then

$$rank(Y) \cdot (1 + rank(Y)) \leq 4 \Rightarrow rank(Y) = 1. \tag{4.42}$$

Case 2: $|J| \geq 5$. The rhs of constraints (4.31)-(4.36) have rank 5 and the system with (4.31)-(4.36) all at equality is infeasible, so $|J| = 5$. All systems with five of (4.31)-(4.36) are systems of 5 linear equations in 5 unknowns. It is easy to check that these all correspond to rank 1 matrices.

Case 3: $2 \leq |J| \leq 4$. Then

$$rank(Y) \cdot (1 + rank(Y)) \leq 10 \Rightarrow rank(Y) \in \{1, 2\}. \tag{4.43}$$

Suppose $rank(Y) = 2$. Since (4.31)-(4.36) implies that the first column of Y dominates the second and third, which gives us that that:

$$\begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \in cone \left\{ \begin{pmatrix} x_1 \\ x_{11} \\ x_{22} \end{pmatrix}, \begin{pmatrix} x_2 \\ x_{12} \\ x_{22} \end{pmatrix} \right\},$$

which gives:

$$\begin{aligned} 1 &= \alpha x_1 + \beta x_2, && (4.44) \\ x_1 &= \alpha x_{11} + \beta x_{12}, && (4.45) \\ x_2 &= \alpha x_{12} + \beta x_{22}, && (4.46) \\ \alpha, \beta &\geq 0, && (4.47) \\ \alpha + \beta &> 0. && (4.48) \end{aligned}$$

Equation (4.44) implies:

$$1 = (\alpha x_1 + \beta x_2)^2. \tag{4.49}$$

Substituting (4.45) and (4.46) into (4.44) gives us:

$$\begin{aligned} 1 &= \alpha^2 x_{11} + 2\alpha\beta x_{12} + \beta^2 x_{22}, && (4.50)\\ 1 &= \alpha(\alpha x_{11} + \beta x_{12}) + \beta(\alpha x_{12} + \beta x_{22}), && (4.51)\\ 1 &\underbrace{\leq}_{(4.31)-(4.34)} \alpha(\alpha x_1 + \beta x_2) + \beta(\alpha x_1 + \beta x_2), && (4.52) \end{aligned}$$

which implies that $\alpha + \beta \geq 1$.

Combining (4.49) and (4.50) we obtain:

$$(\alpha x_1 + \beta x_2)^2 = \alpha^2 x_{11} + 2\alpha\beta x_{12} + \beta^2 x_{22}. \qquad (4.53)$$

Finally, rearranging terms we obtain:

$$\alpha^2(x_{11} - x_1^2) + 2\alpha\beta(x_{12} - x_1 x_2) + \beta^2(x_{22} - x^2) = 0. \qquad (4.54)$$

The semi-definite constraint on Y gives us that:

$$\begin{aligned} x_{11} &\geq x_1^2, && (4.55)\\ x_{22} &\geq x_2^2. && (4.56) \end{aligned}$$

If (4.33) ($x_1 = x_{12}$) or (4.34) ($x_2 = x_{12}$) is binding, then $x_{12} \geq x_1 \cdot x_2$ and (4.54) implies $x_{11} = x_1^2, x_{22} = x_2^2, x_{12} = x_1 x_2$. Thus Y is rank 1, arriving at contradiction.

Therefore, since $|J| \geq 2$, we will examine the subcases where any 2 of the remaining constraints (4.31),(4.32), (4.35), (4.36) are binding as follows:

- Case 3a: (4.31),(4.32) binding: $x_1 = x_{11}$, $x_2 = x_{22}$.
- Case 3b: (4.35),(4.36) binding: $x_{12} = 0$, $x_{12} - x_1 - x_2 + 1 = 0$.
- Case 3c: ((4.31) or (4.32)) and (4.35) binding, by symmetry: $x_1 = x_{11}$,$x_{12} = 0$.
- Case 3d: ((4.31) or (4.32)) and (4.36) binding, by symmetry: $x_1 = x_{11}$,$x_{12} -$

$x_1 - x_2 + 1 = 0.$

(a) Case 3a: (4.31),(4.32) binding: $x_1 = x_{11}, x_2 = x_{22}$.
Then

$$\begin{aligned} x_1(1-\alpha) &= \beta x_{12}, \\ x_2(1-\beta) &= \alpha x_{12}. \end{aligned}$$

Adding these together we obtain:

$$x_1 + x_2 - 1 = (\alpha + \beta)x_{12}. \qquad (4.57)$$

Subtracting (4.36) from (4.57) then gives $(1 - \alpha - \beta)x_{12} \geq 0$. Since $\alpha + \beta \geq 1$ we have $x_{12} = 0$ or $\alpha + \beta = 1$. But $\alpha + \beta = 1$ implies that either $x_1 = x_{12}$ or $x_2 = x_{12}$ in which case either (4.33) or (4.34) are binding which we have already shown to be a contradiction. Thus $x_{12} = 0$ and by (4.57), $x_1 + x_2 = 1$. But then Y has the form:

$$Y = \begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & x_1 & 0 \\ x_2 & 0 & x_2 \end{pmatrix} = x_1 \cdot \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \cdot \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

If either x_1 or x_2 are 0, then Y is a rank 1 matrix, otherwise since $rank(\bar{Y}) = rank(Y)$ we have $\bar{Y}$ is a strict convex combination of feasible matrices contradicting that F is a vertex.

(b) Case 3b:(4.35),(4.36) binding: $x_{12} = 0, x_{12} - x_1 - x_2 + 1 = 0.$
Thus $x_1 + x_2 = 1$ and

$$\begin{aligned} 1 &= \alpha x_1 + \beta x_2 && (4.58) \\ x_1 &= \alpha x_{11} && (4.59) \\ x_2 &= \beta x_{22} && (4.60) \end{aligned}$$

We then have:

$$\begin{aligned} 1 &= \alpha x_1 + \beta x_2 && (4.61) \\ x_1 + x_2 &= \alpha x_1 + \beta x_2 && (4.62) \\ (1-\alpha)x_1 + (1-\beta)x_2 &= 0 && (4.63) \end{aligned}$$

By combining (4.31) with (4.59) and (4.32) with (4.60) we obtain:

$$\begin{aligned} (1-\alpha)x_1 &\leq 0 && (4.64) \\ (1-\beta)x_2 &\leq 0 && (4.65) \end{aligned}$$

We must have $x_1, x_2 > 0$ or else (4.33) ($x_1 = x_{12}$) or (4.34) ($x_2 = x_{12}$) would be tight respectively which we have already shown to be contradictions. Thus $\alpha = \beta = 1$. But then (4.31) and (4.32) are tight, reducing to Case 3a.

(c) Case 3c: ((4.31) or (4.32)) and (4.36) binding, by symmetry: $x_1 = x_{11}$,$x_{12} = 0$: Then

$$\begin{aligned} 1 &= \alpha x_1 + \beta x_2 \\ x_1 &= \alpha x_1 \\ x_2 &= \beta x_{22} \end{aligned}$$

If $x_1 = 0$ then $x_{12} \geq x_1 x_2$ and if $x_1 = 1$ then (4.36) implies $x_2 = 0$. In these cases (4.54) implies $x_{11} = x_1^2, x_{22} = x_2^2, x_{12} = x_1 x_2$ implying Y is rank 1, arriving at contradiction. Thus in order for Y to be rank 2, $0 < x_1 < 1$.
But Y has the form:

$$Y = \begin{pmatrix} 1 & x_1 & x_2 \\ x_1 & x_1 & 0 \\ x_2 & 0 & x_{22} \end{pmatrix} = x_1 \cdot \begin{pmatrix} 1 & 1 & x_2 \\ 1 & 1 & 0 \\ x_2 & 0 & x_{22} \end{pmatrix} + (1 - x_1) \cdot \begin{pmatrix} 1 & 0 & x_2 \\ 0 & 0 & 0 \\ x_2 & 0 & x_{22} \end{pmatrix}$$

and since $rank(\bar{Y}) = rank(Y)$, $\bar{Y}$ is a strict convex combination of feasible matrices contradicting that F is a vertex.

(d) Case 3d: ((4.31) or (4.32)) and (4.36) binding, by symmetry: $x_1 = x_{11}$,$x_{12} - x_1 - x_2 + 1 = 0$:

Then

$$\begin{aligned} 1 &= \alpha x_1 + \beta x_2 \\ x_1 &= \alpha x_1 + \beta x_{12} \\ x_2 &= \alpha x_{12} + \beta x_{22} \end{aligned}$$

Adding (4.66) and (4.66) and subtracting , we obtain:

$$\begin{aligned} x_1 + x_2 - 1 &= \beta(x_{22} - x_2) + (\alpha + \beta)x_{12} && (4.66) \\ 0 &= \beta(x_{22} - x_2) + (\alpha + \beta - 1)x_{12} && (4.67) \end{aligned}$$

By (4.66) and (4.33) we have:

$$x_1 \geq (\alpha + \beta)x_1 \tag{4.68}$$

Since $\alpha + \beta \geq 1$, we must have that $\alpha + \beta = 1$ or $x_1 = 0$. If $x_1 = 0$ the semidefinite constraint $x_{11}x_{22} \geq x_{12}^2$ implies $x_{12} = 0$ and we reduce to Case 3c. Thus for Y to be rank 2 we must have $\alpha + \beta = 1$.

Then together with (4.32) implies $\beta = 0$ or $x_2 = x_{22}$. $x_2 = x_{22}$ reduces to Case 3a. So in order for Y to be rank 2, we must have $\beta = 0$ which gives us that $\alpha = 1$ and $x_2 = x_{12}$. But this means that (4.34) is binding which we have already shown to be a contradiction.

□

Corollary 4.7 *The extreme points of $\mathcal{Y}$ are rank 1 matrices.*

Proof. For an extreme point of $\mathcal{Y}$:

$$\bar{Y} = \begin{pmatrix} 1 & x_1 & x_2 & \boldsymbol{x}_3^T \\ x_1 & x_{11} & x_{12} & X_{13}^T \\ x_2 & x_{12} & x_{22} & X_{23}^T \\ \boldsymbol{x}_3 & X_{13} & X_{23} & X_{33} \end{pmatrix},$$

we have that the upper 3x3 sub-matrix $Y = \boldsymbol{y}\boldsymbol{y}^T$ is rank 1. Since $\bar{Y}$ is positive semi-definite, it is a convex combination of $r = rank(\bar{Y})$ rank 1 matrices:

$$\bar{Y} = \sum_{i=1}^{r} \lambda_i \boldsymbol{w}_i \boldsymbol{w}_i^T, \; \sum_i \lambda_i = 1, \; \lambda_i > 0.$$

We write $\boldsymbol{w}_i = \begin{pmatrix} \boldsymbol{y}_i \\ \bar{\boldsymbol{w}}_i \end{pmatrix}$ with $\boldsymbol{y}_i \in \mathbb{R}^3$.
Then we have

$$\boldsymbol{y}\boldsymbol{y}^T = Y = \sum_{i=1}^{r} \lambda_i \boldsymbol{y}_i \boldsymbol{y}_i^T$$

Since rank 1 matrices are the extreme rays of the semi-definite cone, we must have that:

$$\boldsymbol{y}_i = \alpha_i \boldsymbol{y} \quad \alpha_i \neq 0, \forall i \tag{4.69}$$

$$\sum_i \lambda \alpha_i^2 = 1 \tag{4.70}$$

By (4.69) the matrix $\bar{Y}_i = \frac{1}{\alpha_i^2}\bar{Y} \in \mathcal{Y}$; and by (4.70), $\bar{Y}$ is a strict convex combination of $\bar{Y}_i, i = 1, ..., r$. Thus all $\bar{Y}_i$ must lie on the same face as $\bar{Y}$. If $r > 1$, this contradicts $\bar{Y}$ lying on a zero-dimensional face. □

Since $\mathcal{Y}$ is a closed convex set, it is the convex hull of it's extreme points and extreme directions. We have shown that the extreme points of $\mathcal{Y}$ are rank 1 matrices, it remains to study it's extreme directions.

Theorem 4.8 *$\mathcal{Y}$ has no extreme directions.*

Proof. Suppose R is an extreme direction of $\mathcal{Y}$. If $\bar{Y}$ an extreme point of $\mathcal{Y}$ then we have:

$$\{\bar{Y} + \lambda R, \lambda > 0\}$$

is an extreme ray of $\mathcal{Y}$. Let us write:

$$\begin{aligned} R &= \begin{pmatrix} A & C^T \\ C & D \end{pmatrix}, \\ \bar{Y} &= \begin{pmatrix} Y & B^T \\ B & X \end{pmatrix}, \end{aligned}$$

where $A, Y \in \mathbb{S}^3, Y = \boldsymbol{y}\boldsymbol{y}^T$. The linear constraints on $\mathcal{Y}$ bound the entries of $Y + \lambda A$ between 0 and 1 component-wise. Thus $A = \mathbf{0}$.

Suppose $C \neq \mathbf{0}$, then pick indices $\boldsymbol{i}, \boldsymbol{j} : C_{ij} \neq 0$. Since $\bar{Y} + \lambda R \succeq 0 \ \forall \lambda > 0$, the submatrix:

$$\begin{pmatrix} \bar{Y}_{jj} & \lambda C_{ij} \\ \lambda C_{ij} & \lambda D_{ii} \end{pmatrix} \succeq 0,$$

which implies $\bar{Y}_{jj} \cdot \lambda D_{ii} - \lambda^2 C_{ij}^2 \geq 0, \ \forall \lambda \geq 0$ which is a contradiction.

Since R has the form:

$$R = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & D \end{pmatrix},$$

we must have that $D = \boldsymbol{d}\boldsymbol{d}^T$ is rank 1. But then:

$$\bar{Y} + \lambda R = \begin{pmatrix} \boldsymbol{y}\boldsymbol{y}^T & \mathbf{0} \\ \mathbf{0} & \lambda \boldsymbol{d}\boldsymbol{d}^T \end{pmatrix}$$

$$= \frac{1}{2}\left(\begin{pmatrix} y \\ \sqrt{\lambda}\boldsymbol{d} \end{pmatrix}\begin{pmatrix} y \\ \sqrt{\lambda}\boldsymbol{d} \end{pmatrix}^T + \begin{pmatrix} y \\ -\sqrt{\lambda}\boldsymbol{d} \end{pmatrix}\begin{pmatrix} y \\ -\sqrt{\lambda}\boldsymbol{d} \end{pmatrix}^T\right).$$

So $\bar{Y} + \lambda R$ is a convex combination of two points in $\mathcal{Y}$ implying both these points lie on the extreme ray and thus $\boldsymbol{d} = \boldsymbol{0}$.

Since R must be the zero matrix, $\mathcal{Y}$ has no extreme directions. □

Since all extreme points of $\mathcal{Y}$ are rank 1 matrices and it has no extreme directions, we conclude that $\mathcal{Y} \subseteq \mathcal{D}$ and that the Sherali-Adams closure is exact. Hence $\mathcal{Y} = \mathcal{D}$ proving Theorem 4.3.

4.6 Computational Results

We tested Algorithm 3 against the method of Natarajan et al. [34] on a set of data provided by an institutional fixed income asset manager consisting of 30 potential obligors. We used the same objective function used in the computational testing of Natarajan et al. [34] which approximates the normalized exponential utility function:

$$u(x) = \frac{1 - \exp(-200x)}{200}$$

by ten linear segments shown in Table 4.1.

Returns were converted to a daily rate to match the scaling of this function. The results of the lower bound on performance obtained for each policy in each iteration of the decomposition method as well as the method of Natarajan et al. [34] are shown in Table 4.2. The best bound is found in the 23rd iteration, when the support of assets $\overline{8}$ and 7 are tight and thus the corresponding solution is selected. In each iteration of the algorithm, we are tightening the support of two assets of the adversarial problem; however when we combine two assets with known support into a single asset with

u	c_u	d_u
1	1.3521	0.0002
2	1.1070	0
3	0.8848	0
4	0.6891	0.0002
5	0.5367	0.0006
6	0.4179	0.0011
7	0.3178	0.0016
8	0.2355	0.0021
9	0.1626	0.0027
10	0.1037	0.0033

Table 4.1: Piecewise-linear approximation to exponential utility.

known support, we are slightly relaxing the adversarial problem since:

$$supp(r_{\overline{n-1}}) \subseteq \left[\frac{w_{n-1}}{w_{n-1}+w_n}\cdot a_{n-1}+\frac{w_n}{w_{n-1}+w_n}\cdot a_n, \frac{w_{n-1}}{w_{n-1}+w_n}\cdot b_{n-1}+\frac{w_n}{w_{n-1}+w_n}\cdot b_n\right],$$

and this inclusion need not be equality. Thus the sequence of bounds obtained is not monotone. We see that the bound obtained in each iteration for tightly projecting onto all but eight asset/portfolio pair is better than that provided by the convolution of the *mean and covariance* and *mean and support* bounds from Natarajan et al. [34].

4.7 Application to the MAXCUT Problem

We note that with a slight modification, a similar decomposition approach is applicable to the MAXCUT problem. In MAXCUT, we are given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where each edge in the graph is given a particular weight $w_e, e \in \mathcal{E}$. The objective is to partition the vertices $\mathcal{N}$ into two sets, $\mathcal{N}^+$ and $\mathcal{N}^-$, where the cut-weight, that is the total weight of edges between the two sets, $\sum_{i\in\mathcal{N}^+, j\in\mathcal{N}^-} w_{(i,j)}$ is maximized. This problem is well known to be $\mathcal{NP}$ – *hard* [37].

Method/Z_{N_t}	Bound
Natarajan et al. [34]	0.8106
$Z(N_1)$	0.8331
$Z(N_2)$	0.835
$Z(N_3)$	0.8038
$Z(N_4)$	0.8188
$Z(N_5)$	0.8311
$Z(N_6)$	0.828
$Z(N_7)$	0.826
$Z(N_8)$	0.8269
$Z(N_9)$	0.8314
$Z(N_{10})$	0.8179
$Z(N_{11})$	0.8156
$Z(N_{12})$	0.8212
$Z(N_{13})$	0.8229
$Z(N_{14})$	0.8048
$Z(N_{15})$	0.8154
$Z(N_{16})$	0.8185
$Z(N_{17})$	0.8066
$Z(N_{18})$	0.808
$Z(N_{19})$	0.8033
$Z(N_{20})$	0.8098
$Z(N_{21})$	0.8018
$Z(N_{22})$	0.7986
$Z(N_{23})$*	0.8377
$Z(N_{24})$	0.8348
$Z(N_{25})$	0.8364
$Z(N_{26})$	0.8325
$Z(N_{27})$	0.818
$Z(N_{28})$	0.8134
$Z(N_{29})$	0.8361

Table 4.2: Achievable bounds obtained on fixed income data set, 10^{-4} scaling.

It is easy to see that MAXCUT can be modeled as a discrete optimization problem by introducing decision variables:

$$x_i = \begin{cases} +1, & i \in \mathcal{N}^+, \\ -1, & i \in \mathcal{N}^-. \end{cases}$$

Note that:

$$w_{(i,j)}(1 - x_i x_j)/2 = \begin{cases} w_{(i,j)}, & (i,j) \in \delta(\mathcal{N}^+, \mathcal{N}^-), \\ 0, & \textit{otherwise}. \end{cases}$$

Here $\delta(\mathcal{N}^+, \mathcal{N}^-)$ is the set of edges that connect $\mathcal{N}^+$ and $\mathcal{N}^-$. Letting W be the matrix of edge weights MAXCUT is then equivalent to the following discrete optimization problem:

$$\begin{aligned} Z_{MC} = \min \quad & \boldsymbol{x}^T W \boldsymbol{x} \\ s.t. \quad & x_i \in \{-1, 1\}, \quad \forall i \in \mathcal{N}. \end{aligned} \tag{4.71}$$

For any $\boldsymbol{x}$ feasible for problem (4.71), we have that $x_i^2 = 1$, thus we can arbitrarily alter the diagonal elements of W to obtain an equivalent optimization problem. Let $\lambda_{min} = |\min eign(W)|$ and define $\overline{W} = W + D$, where D is a diagonal matrix such that:

$$D_{ii} = 1 + \max\{\lambda_{min}, \sum_{j \neq i} W_{i,j}\}.$$

We then have that $\overline{W}$ is positive definite and strictly diagonally dominant. Also the optimization problem:

$$\begin{aligned} \bar{Z}_{MC} = \min \quad & \boldsymbol{x}^T \overline{W} \boldsymbol{x} \\ s.t. \quad & x_i \in \{-1, 1\} \quad \forall i \in \mathcal{N}, \end{aligned}$$

is equivalent to problem (4.7).

We can apply our decomposition method to MAXCUT by iteratively selecting a pair (or other small subset) of vertices setting the subpolicy of assignment to $\mathcal{N}^+$ and $\mathcal{N}^-$ for that pair. In the context of MAXCUT, this amounts to simply dropping the constraints $x_k \in \{-1,1\}$ for $k \notin \{i,j\}$. More formally we can write this method as follows:

Algorithm 4 (Decomposition approach for MAXCUT). *Let $\mathcal{N}_t^+, \mathcal{N}_t^-$ be the set of vertices whose assignment to $\mathcal{N}^+$ and $\mathcal{N}^-$ respectively has been fixed by iteration t; and let $\mathcal{N}_t = \mathcal{N} \setminus (\mathcal{N}_t^+ \cup \mathcal{N}_t^-)$ be the set of vertices yet unassigned. The decomposition approach for using a subset size K is given as follows:*

1. *Let $t := 0, \mathcal{N}_0 := \mathcal{N}, \mathcal{N}_t^+ := \emptyset, \mathcal{N}_t^- := \emptyset$*
2. *DO UNTIL $\mathcal{N}_t^+ = \emptyset$*
    (a) *For each subset $\mathcal{S}$ of $\mathcal{N}_t$ with $|\mathcal{S}| = K$:*
        i. *Solve the optimization problem:*

$$\begin{array}{lll} \min & x^T \overline{W} x & \\ s.t. & x_i \in \{-1,1\} & \forall i \in \mathcal{S} \\ & x_i = 1 & \forall i \in \mathcal{N}_t^+ \\ & x_i = -1 & \forall i \in \mathcal{N}_t^- \end{array} \tag{4.72}$$

        to obtain an optimal solution $x^{t*}(\mathcal{S})$.
    (b) *Let $x^{t*} = \min_{\mathcal{S} \subset \mathcal{N}_t : |\mathcal{S}| = K} \left(x^{t*}\right)^T \overline{W} x^{t*}$ be the minimum value obtained and $\mathcal{S}^*$ be the subset that attains this minimum.*
    (c) *For each $i \in \mathcal{S}^*$*

$$\begin{array}{lll} \textit{Let} & \mathcal{N}_{t+1}^+ := \mathcal{N}_t^+ \cup \{i\} & \textit{if } x_i^{t*} = 1, \\ & \mathcal{N}_{t+1}^- := \mathcal{N}_t^- \cup \{i\} & \textit{if } x_i^{t*} = -1 \end{array}$$

    (d) *Let $\mathcal{N}_{t+1} = \mathcal{N}_t \setminus \mathcal{S}^*$*
    (e) *Let $t := t + 1$*

Since $\bar{W}$ is positive definite, we note that each optimization problem (4.72) can be solved by solving 2^K linear systems arising from each possible ±1 assignments of the variables with indices in $\mathcal{S}$. These linear systems have the form:

$$2\overline{W}(\mathcal{T})\boldsymbol{x}(\mathcal{T}) = \boldsymbol{b}(\mathcal{T})$$

where $\overline{W}(\mathcal{T})$ is the sub-matrix of $\overline{W}$ corresponding to the indices in

$$\mathcal{T} = \mathcal{N} \backslash \left(\mathcal{N}_t^+ \cup \mathcal{N}_t^- \cup \mathcal{S}\right)$$

and the 2^K versions of $\boldsymbol{b}$ arise from enumerating all possible $\{-1, 1\}$ assignments of x_j for $j \in \mathcal{S}$ given by:

$$b_i = \sum_{j \in \mathcal{N}_t^-} \overline{W}_{i,j} - \sum_{j \in \mathcal{N}_t^+} \overline{W}_{i,j} - \sum_{j \in \mathcal{S}} x_j \cdot \overline{W}_{i,j}$$

Since $\overline{W}(\mathcal{S})$ is symmetric and diagonally dominant, each of these linear systems can be solved very efficiently in $\mathcal{O}\left(m \log^2(n)\right)$ time due to the method of Koutis, Miller and Peng[32] where m is the number of non-zero entries in $\bar{W}$ and $n = |\mathcal{N}|$. Step 2 is repeated $\lceil \frac{n}{K} \rceil$ times and Step 2(a) is repeated $\mathcal{O}\left(\binom{n}{K}\right)$ times for an overall algorithmic complexity of $\mathcal{O}\left(mn^K \log^2(n)\right)$. Additionally, the overall running time of this heuristic can be scaled by sampling from all subsets of size K for Step 2(a) rather than exhausting them. We examine the effects of altering K and sampling in the computational results that follow.

4.8 Computational Results for MAXCUT

We applied Algorithm 4 on three families of tests:

1. Exhaustively enumerating all possible 0/1 edge-weight graphs on 3-7 vertices and comparing the method with $K = 2, 3, 4, 5$ to the optimal solution.

2. Random graphs generated on 8-30 vertices with $K = 2, 3, 4$. The method is compared to the Goemans-Williamson approach [24].

3. Four large-scale Ising-spin model challenge problems from the 2000 DIMACS challenge [1], where the method is compared to the best results reported from the technical report of the competition [40]. In this section, we test $K = 2$ with random subset sampling for Step 2(a) of the algorithm.

4.8.1 Exhaustive enumeration of graphs on 3-7 vertices

For this family of tests, we generated every possible 0/1 edge-weight graph on 3-7 vertices. The optimal solution was found using exhaustive search. The decomposition method was run on each graph with $K = 2, 3, 4, 5$. As shown in Table 4.3, this method obtains the optimal solution in all instances of possible 0/1 graphs on 3-6 vertices and almost all instances of graphs on 7 vertices. In those graphs where the optimal solution is not obtained, the cut obtained by the decomposition approach contained one less edge than that of the optimal solution; increasing K above 2 had little effect on performance. Figure 4-1 shows one such instance. Note that Goemans-Williamson rounding [24] obtained the optimal cut for this particular instance.

Number of Vertices	Number of Instances	K=2	K=3	K=4	K=5
3	8	100%	100%	N/A	N/A
4	64	100%	100%	100%	N/A
5	1,024	100%	100%	100%	100%
6	32,768	100%	100%	100%	100%
7	2,089,602	99.64%	99.69%	99.64%	99.98%

Table 4.3: Percentage of all possible instances for which optimal cut is obtained for 0/1 graphs on 3-7 vertices.

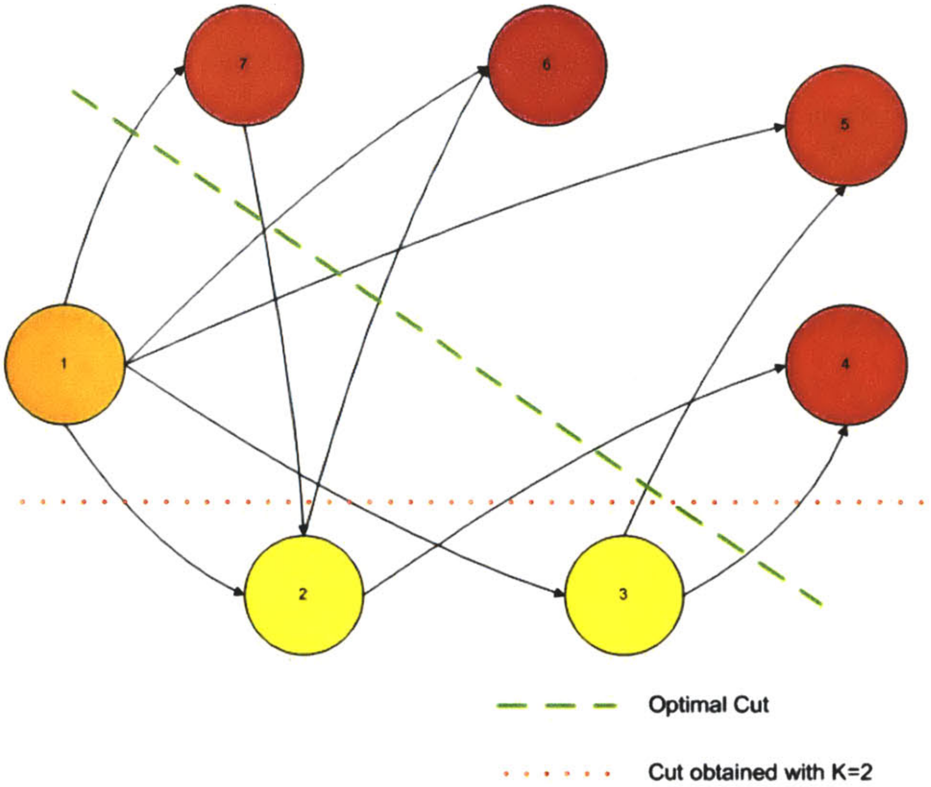


Figure 4-1: Instance where decomposition approach does not obtain optimal cut.

4.8.2 Random graphs on 8-30 vertices

For this family of tests, random 0/1 edge-weight graphs on 8-30 vertices were generated. 100 instances were created for each size. Graphs were generated by including each edge independently with 50% probability. To benchmark the decomposition method, we used the Goemans-Williamson [24] approach which provides a semi-definite optimization-based bound on the optimal solution and method of randomized rounding to obtain a feasible solution with a guaranteed 87.8% attainment of that bound in expectation. For each instance, we performed 1000 trials of randomized rounding and selected the best solution. Figure 4-2 shows the average attainment of the semi-definite optimization bound by graph size. The decomposition method attains between 98-99% of the bound on average for all problem sizes, outperforming Goemans-Williamson rounding which obtained between 95-97%. Increasing the subset size from $K = 2$ to $K = 3$ or $K = 4$ has little effect on performance.

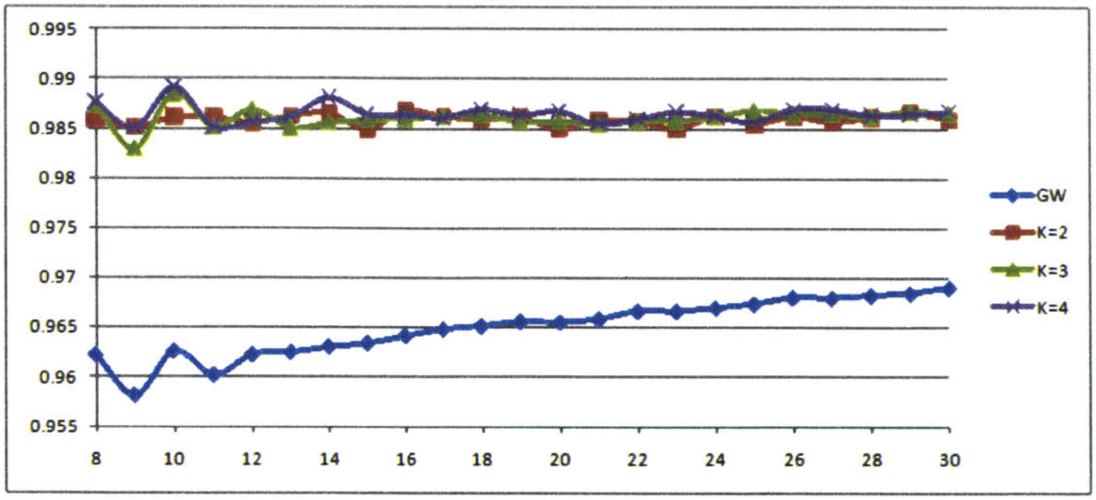


Figure 4-2: Random Graphs: Average Attainment of Optimality Bound by Graph Size.

Figure 4-3 shows the minimum attainment of each method over all instances for each problem size. Again we see a noticeable margin in performance of the decomposition method over Goemans-Williamson rounding.

4.8.3 DIMACS challenge problems

For this family of tests, we used the DIMACS torus set challenge problems [1] which are a set of four large-scale MAXCUT problems from the Ising model of spin glasses. Problems "torusg3-8" and "toruspm3-8-50" are graphs on 512 vertices whereas problems "torusg3-15" and "toruspm3-15-50" are graphs on 3,375 vertices. Due to the large-scale nature of these problems, it is prohibitively expensive to solve (4.72) for every subset in Step 2(a) of the decomposition method. As an alternative to solving for each subset for $K = 2$, we compared randomly sampling one subset with randomly

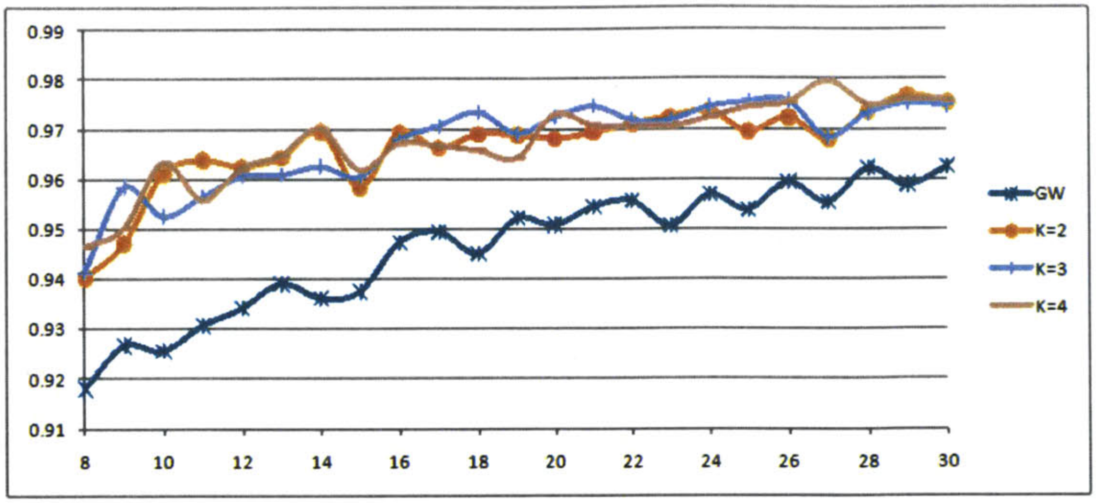


Figure 4-3: Random Graphs: Minimum Attainment of Optimality Bound by Graph Size.

sampling 30% of all subsets for the graphs on 512 vertices and 0.003% of all subsets for the graphs on 3,375 vertices. Table 4.4 shows the results of the decomposition method as compared to the best submitted for the challenge competition [40]. Even with very sparse sampling, the decomposition method was able to outperform the best reported cut from the challenge in two of the four instances, one for each of the problem sizes.

Problem	Size	Decomposition	Decomposition	best from
Problem	Size	(1 sample)	(30%/0.003%)	Competition [40]
torusg3-8	512	390.73	399.36	391.11
toruspm3-8-50	512	436	446	458
torusg3-15	3375	2545.21	2698.29	2602.03
toruspm3-15-50	3375	2788	2872	3016

Table 4.4: Value of solution obtained to DIMACS challenge problems.

4.9 Concluding Remarks

In this chapter, we demonstrated the ability to implement a decomposition method in the context of distributionally robust optimization with known mean covariance and support. In each iteration of this method, a sub-policy for the weighting of two random variables is derived by forcing the adversarial problem to be tight on the projection of those two assets. We show that this tightness can be achieved efficiently by proving that the Sherali-Adams closure of a quadratic optimization problem in n-dimensions over a box in two-dimensions is exact. Each iteration also provides us with a bound on the ultimate performance of the resulting heuristic by constraining the support of the adversarial problem in differing dimensions.

We also outline the application of our decomposition strategy for distributionally robust optimization with known mean, covariance and support to the fixed income portfolio optimization problem. In contrast to previous approaches that solve a single weak relaxation of the problem, this approach allows us to build a policy from the ground up by constructing sub-policies on relaxations that are tight in the dimensions of interest. In each step of policy construction we obtain an achievable bound and are allowed to select a policy that attains the best bound encountered.

In addition demonstrated promising performance of our decomposition method for the MAXCUT problem, whose feasible region is the boundary of that of the separation problem for distributionally robust optimization. For large problems we also see that random sampling of projected subproblems to save computation time performs well, outperforming reported results on 2 of the 4 DIMACS [1] challenge problems.

Chapter 5

Conclusions

The decomposition methods presented in this thesis provide a tractable family of heuristics for large-scale stochastic and robust optimization problems. Using relaxations that are tight on the projections of subsets of dimensions of the problem to obtain sub-policies over these dimensions, we obtain efficient heuristics that perform well in practice on large-scale instances.

- In the context of stochastic optimization, this admits not only a new heuristic for classical restless bandit problems, but also allows us to define and solve a much broader class of Markov decision problems, Generalized Restless Bandits which have widespread real world application (Chapter 2). Our decomposition strategy displays promising computational results in the form of the Nested Policy Heuristic for these problems.
- The feasibility of implementation on large-scale real world systems is also demonstrated by application to the sponsored search advertising problem (Chapter 3). Additionally, simplifications to derivation of the Nested Policy Heuristic are discovered for large-scale problems with identical arms.
- In the context of robust optimization, the separation problem for distributionally robust optimization with known mean, covariance, and two variable support

is equivalent to indefinite quadratic optimization in n dimensions over a box in 2 dimensions. The Sherali-Adams closure [47] of the semi-definite relaxation to this problem is proved to be exact, giving us an efficiently solvable approach (4). This allows us to use a decomposition strategy for distributionally robust optimization with known mean, covariance, and n-variable support. The application of this decomposition strategy to distributionally robust optimization with known mean, covariance, and support is shown in the context of fixed income portfolio optimization. This approach shows promising computational results over existing methods to solve the distributionally robust problem which often rely on weak relaxations. It is also shown that a similar Projection-Based Decomposition approach yields an efficient heuristic for the MAXCUT problem with good performance relative to the Goemans-Williamson [24] rounding on moderately sized graphs and challenge submissions for the DIMAC challenge problems [1].

Appendix A

Arm Aggregation for arbitrary subset size

Define the binary variables $w(g)$ for each $g \subset \mathcal{A} : |g| = \ell$ such that $w(g) = 1$ if $g \in \mathcal{G}^*$. The mixed-integer optimization formulation of the Arm Aggregation problem is then:

$$
\begin{aligned}
\max \; & \sum_{g \subset \mathcal{A}:|g|=\ell} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \sum_{\boldsymbol{s} \in S_g} \left(\sum_{i \in g} R_i^{d_i}(s_i) \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}) \\
s.t. \; & \sum_{\boldsymbol{d} \in \mathcal{D}_g} x_g^{\boldsymbol{d}}(\boldsymbol{s}) = \alpha_{\boldsymbol{s}} w(g) + \beta \sum_{\boldsymbol{s}' \in S_g} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \left(\prod_{i \in g} P_i \left(s_i' \rightarrow s_i \right) \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}'), \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\, g \subset \mathcal{A} : |g| = \ell,\ \boldsymbol{s} \in S_g, \\
& \sum_{g \subset \mathcal{A}:|g|=\ell} \sum_{\boldsymbol{d} \in \mathcal{D}_g} \sum_{\boldsymbol{s} \in S_g} \left(\sum_{i \in g} d_i \right) x_g^{\boldsymbol{d}}(\boldsymbol{s}) = \frac{M}{1-\beta}, \\
& x_g^{\boldsymbol{d}}(\boldsymbol{s}) \leq U w(g),\ \forall\, g \subset \mathcal{A} : |g| = \ell,\ \boldsymbol{s} \in S_g,\ \boldsymbol{d} \in \mathcal{D}_g \\
& \sum_{g \subset \mathcal{A}:|g|=\ell, g \ni k} w(g) = 1,\ \forall k \in \mathcal{A} \\
& x_g^{\boldsymbol{d}}(\boldsymbol{s}) \geq 0,\ \forall\, g \subset \mathcal{A} : |g| = \ell,\ \boldsymbol{s} \in S_g,\ \boldsymbol{d} \in \mathcal{D}_g \\
& w(g) \in \{0, 1\},\ \forall g \subset \mathcal{A} : |g| = \ell
\end{aligned}
$$

Bibliography

[1] DIMACS 7th Challange website. http://dimacs.rutgers.edu/challenges/seventh/, 2000.

[2] K. Anstreicher and S. Burer. Computable representations for convex hulls of low-dimensional quadratic forms, Working paper, Dept. of Management Sciences, University of Iowa, 2007.

[3] M. Asawa and D. Teneketzis. Multi-armed bandits with switching penalties. *IEEE Transactions on Automatic Control*, 41(3):328–348, 1996.

[4] D. Bertsimas and R. Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48(4):550–565, 2002.

[5] D. Bertsimas, X. V. Doan, and K. Natarajan. Bounds on some contingent claims with non-convex payoff based on multiple assets. *Technical Report*, 2007.

[6] D. Bertsimas, X. V. Doan, K. Natarajan, and C. Teo. Models for minimax stochastic linear optimization problems with risk aversion. *Mathematics of Operations Research*, 35(3):580–602, 2010.

[7] D. Bertsimas and J. Niño-Mora. Conservation laws, extended polymatroids and multi-armed bandit problems; a polyhedral approach to indexable systems. *Mathematics of Operations Research*, 21(2):257–306, 1996.

[8] D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000.

[9] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15:780–804, 2001.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge Univ. Press, Cambridge, U.K., 2004.

[11] X. Burtschell, J. Gregory, and L.-P. Laurent. A comparative analysis of cdo pricing models, Working paper, 2005.

[12] X. Chen, M. Sim, and P. Sun. A robust optimization perspective on stochastic programming. *Operations Research*, 55(6):1058–1071, 2007.

[13] P. Crosbie and J. Bohn. Modeling default risk. 2003.

[14] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3), 2010.

[15] K. Deng, Y. Sun, P. G. Mehta, and S. P. Meyn. An information-theoretic framework to aggregate a Markov chain. In *Proceedings of American Control Conference*, pages 731–736, St. Louis, June 2009.

[16] D. Dwyer, A. Kocagil, and R. Stein. Moodys kmv riskcalc v3.1 model: Next-generation technology for predicting private firm credit risk. 2004.

[17] L. El Ghaoui, M. Oks, and F. Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556, 2003.

[18] J. C. Gittins. *Multi-armed Bandit Allocation Indices.* John Wiley, New York, 1989.

[19] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In *Progress In Statistics: European Meeting of Statisticians*, Budapest, 1972.

[20] P. Glasserman, W. Kang, and P. Shahabuddin. Large deviations in multifactor portfolio credit risk. *Mathematical Finance*, 17(3):345–379, 2007.

[21] P. Glasserman and J. Li. Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656, 2005.

[22] K. Glazebrook, J. Niño-Mora, and P. Ansell. Index policies for a class of discounted restless bandits. *Advances in Applied Probability*, 34(4):754–774, 2002.

[23] K. Glazebrook, D. Ruiz-Hernandez, and C. Kirkbride. Some indexable families of restless bandit problems. *Advances in Applied Probability*, 38:643–672, 2006.

[24] M. X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[25] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4):902–917, 2010.

[26] RiskMetrics Group. Creditmetrics technical document. 1997.

[27] D. Hodge and K. Glazerbrook. Asymptotic optimality of multi-action restless bandits. In *Young European Queueing Theorists IV, EURANDOM*, Eindhoven, November 2010.

[28] K. Isii. On the sharpness of chebychev-type inequalities. *Ann. Inst. Stat. Math.*, 14:185–197, 1963.

[29] D. Jewan, R. Guo, and G. Witten. Copula marginal expected tail loss efficient frontiers for cdos of bespoke portfolios. In *Proceedings of the World Congress on Engineering 2008 Vol II*, London, July 2008.

[30] S. Kavadias and R. O. Chao. *Resource Allocation and New Product Development Portfolio Management. Chapter in Handbook of New Product Development Research; C. H. Loch and S. Kavadias (ed.).* Elsevier/Butterworth, Oxford, 2007.

[31] S. Kavadias and C. H. Loch. Dynamic prioritization of projects at a scarce resource. *Production and Operations Management*, 12(4):433–444, 2003.

[32] I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving sdd linear systems*, 2010.

[33] H.M. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[34] K. Natarajan, M. Sim, and J. Uichanco. Tractable robust expected utility and risk models for portfolio optimization. *Mathematical Finance*, 20(4):695–731, 2010.

[35] A. Niculescu-Mizil. Multi-armed bandits with betting. In *Conference on Learning Theory Workshop: On-line Learning with Limited Feedback*, Montreal, June 2009.

[36] J. Niño-Mora and S. Villar. Multitarget tracking via restless bandit marginal productivity indices and kalman filter in discrete time. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, December 2009.

[37] C. H. Papadimitriou. *Computational Complexity.* Addison Wesley, Boston, 1993.

[38] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queueing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.

[39] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optiaml eigenvalues. *Mathematics of Operations Research*, 1998.

[40] G. Pataki and S. Smieta. The dimacs library of semidefinite-quadratic-linear programs.

[41] D. Pisinger. A minimal algorithm for the multiple choice knapsack problem. Technical Report 94/25, University of Copenhagen, Copenhagen, May 1994.

[42] I. Popescu. Robust mean-covariance solutions for stochastic optimization. *Operations Research*, 55(1):98–112, 2007.

[43] R. T. Rockefellar. *Convex Analyis.* Princeton University Press, Princeton, New Jersey, 1970.

[44] D. Ruiz-Hernandez. *Indexable Restless Bandits: Index Policies for Some Families of Stochastic Scheduling and Dynamic Allocation Problems.* VDM Verlag, 2008.

[45] P. Rusmevichientong and D. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 260–269, New York, NY, USA, 2006. ACM.

[46] H. Scarf. *A min-max solution of an inventory problem*, pages 201–209. Stanford University Press, Stanford, CA, 1958.

[47] H.D. Sherali and W.P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems.* Kluwer, Dordrecht, 1998.

[48] J. Smith. Generalized chebychev inequalities: Theories and applications in decision analysis. *Operations Research*, 43(5):807–825, 1995.

[49] M. Veatch and L. M. Wein. Schedulinga make-to-stock queue: index policies and hedging points. *Operations Research*, 44:634–647, 1996.

[50] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[51] R. Washburn and M. K. Schneider. Optimal policies for a class of restless multiarmed bandit scheduling problems with applications to sensor management. *Journal of Advances in Information Fusion*, 3(1), 2008.

[52] R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.

[53] Google Analytics website. www.google.com/analytics/, 2011.

[54] P. Whittle. Multi-armed bandits and the Gittins' index. *Journal of Royal Statistical Society*, 42(2):143–149, 1980.

[55] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25A:287–298, 1988.