A Lagrangian Decomposition Approach to Weakly Coupled Dynamic Optimization Problems and its Applications

by

Jeffrey Thomas Hawkins

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

June 2003

© Massachusetts Institute of Technology 2003. All rights reserved.

Author.........................................................Jeff Hawkins

Sloan School of Management
May 16, 2003

Certified by.......................................................Dimitris Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Certified by.......................................................Georgia Perakis
Sloan Career Development Associate Professor of Operations Research
Thesis Supervisor

Accepted by......................................................James B. Orlin
Co-Director, Operations Research Center

ARCHIVES
A Lagrangian Decomposition Approach to Weakly Coupled Dynamic Optimization Problems and its Applications

by

Jeffrey Thomas Hawkins

Submitted to the Sloan School of Management on May 16, 2003, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Operations Research

Abstract

We present a Lagrangian based approach to decoupling weakly coupled dynamic optimization problems for both finite and infinite horizon problems. The main contributions of this dissertation are: (i) We develop methods for obtaining bounds on the optimal cost based on solving low dimensional dynamic programs; (ii) We utilize the resulting low dimensional dynamic programs and combine them using integer programming methods to find feasible policies for the overall problem; (iii) To illustrate the power of our methods we apply them to a large collection of dynamic optimization problems: multiarmed bandits, restless bandits, queueing networks, serial supply chains, linear control problems and on-line auctions, all with promising results. In particular, the resulting policies appear to be near optimal. (iv) We provide an in-depth analysis of several aspects of on-line auctions, both from a buyer's and a seller's perspective. Specifically, for buyers we construct a model of on-line auctions using publicly available data and develop an algorithm for optimally bidding in multiple simultaneous auctions. For sellers we construct a model of on-line auctions using publicly available data and demonstrate how a seller can increase the final selling price using dynamic programming.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Professor of Operations Research

Thesis Supervisor: Georgia Perakis
Title: Sloan Career Development Associate Professor of Operations Research
Acknowledgments

I would first like to thank my two advisors, Dimitris Bertsimas and Georgia Perakis. In their own ways they have both taught me to inquire and to persevere. I am thankful for the guidance, support and friendship they have provided me and for helping me grow both as a student and as a person. I am proud to be among their students. I would like to thank Jeremie Gallien for his invaluable input as a thesis committee member.

I would also like to thank the students of the Operations Research Center for making the past four years both intellectually and socially stimulating. I hope I will always be surrounded by people as intelligent and as motivated. I would also like to thank my undergraduate advisor, Martin Puterman, for introducing me to Operations Research and for being a mentor.

Finally, I would like to thank my close friends and family for their love and support.
Contents

1 Introduction
   1.0.1 An overview of the theory .............................................. 20
   1.0.2 An overview of the applications ........................................ 20
   1.0.3 Contributions .................................................................. 22
   1.0.4 Structure of dissertation .................................................. 23

2 A Lagrangian decomposition approach to weakly coupled dynamic problems ................................................. 25
   2.1 Introduction ..................................................................... 25
     2.1.1 Formulation .................................................................. 26
     2.1.2 Literature review .......................................................... 28
     2.1.3 Structure of the chapter .................................................. 29
   2.2 Infinite horizon weakly coupled discounted optimization problems ......................................................... 30
     2.2.1 Computation of optimal Lagrangian bound for infinite horizon problems ............................................. 33
   2.3 Finite horizon weakly coupled discounted optimization problems ............................................................ 36
     2.3.1 Computation of Lagrangian bound for finite horizon problems ............................................................. 39
   2.4 Lagrange-based policies for infinite horizon problems ................................................................................. 40
     2.4.1 Integer optimization ........................................................ 41
     2.4.2 Piecewise approximations ................................................. 42
2.4.3 Choice of Lagrange multipliers .................................. 42
2.4.4 Minimal-lambda feasible policy .................................. 45
2.5 Conclusions ............................................................. 46

3 Applications ............................................................. 47

3.1 Multiarmed bandits ..................................................... 47
  3.1.1 Introduction ....................................................... 48
  3.1.2 Formulation ....................................................... 48
  3.1.3 Optimality of the Lagrange-based policy ..................... 49
  3.1.4 Implementing Algorithm 1 using linear programming ........ 55
  3.1.5 The multiarmed bandit problem with $K > 1$ ................. 57
3.2 Restless bandits ....................................................... 57
  3.2.1 Introduction ....................................................... 58
  3.2.2 Structure .......................................................... 58
  3.2.3 Formulation ....................................................... 59
  3.2.4 A Lagrangian policy for restless bandits ..................... 60
  3.2.5 An integer optimization Lagrange-based policy and the primal-
       dual heuristic ..................................................... 62
  3.2.6 An $M$-order coupling approach to restless bandits .......... 67
  3.2.7 Linear programming results and pairwise couplings .......... 71
  3.2.8 Empirical results ................................................. 76
3.3 Multiclass queueing networks ...................................... 78
  3.3.1 Structure .......................................................... 78
  3.3.2 Formulation ....................................................... 79
  3.3.3 Buffered multiclass queueing networks as weakly
       coupled dynamic optimization problems ......................... 82
  3.3.4 Lagrangian decoupled formulation and LP methods ........... 85
  3.3.5 Numerical results ................................................. 86
3.4 Supply chain inventory problems .................................. 89
  3.4.1 Standard formulation ........................................... 90
3.4.2 Decomposed formulation ........................................ 93
3.4.3 Lagrangian decoupling ........................................ 100
3.5 Conclusions ..................................................... 101

4 Extensions to Dynamic Linear Systems under Constraints 103
  4.1 Formulation ................................................... 103
  4.2 Lagrangian decomposition ..................................... 104
    4.2.1 Analytical comparison of bounds ....................... 109
    4.2.2 Partial decomposition .................................. 111
  4.3 A Lagrange-based policy ...................................... 112
    4.3.1 Greedy heuristic ........................................ 114
    4.3.2 Deterministic approximation heuristic ................ 114
  4.4 Numerical results ........................................... 115
    4.4.1 Comparison of bounds .................................. 115
    4.4.2 Comparison of heuristics ............................... 115
    4.4.3 Summary ................................................ 116
  4.5 Conclusions ................................................ 117

5 Optimal bidding in on-line auctions 119
  5.1 Introduction ............................................... 119
    5.1.1 Overview of on-line auctions .......................... 120
    5.1.2 Literature review ...................................... 120
    5.1.3 Philosophy and contributions .......................... 122
    5.1.4 Structure of the chapter ............................... 124
  5.2 Single item auction ......................................... 124
    5.2.1 The model .............................................. 124
    5.2.2 State .................................................. 125
    5.2.3 Control ............................................... 125
    5.2.4 Randomness ............................................ 125
    5.2.5 Dynamics .............................................. 126
    5.2.6 Objective .............................................. 128
5.2.7 Bellman equation ........................................... 129
5.2.8 Estimation of parameters ................................. 130
5.2.9 Empirical results .......................................... 133
5.2.10 Bidding against multiple competitors ............... 136
5.3 Multiple auctions ............................................. 138
  5.3.1 Approximate dynamic programming method 1 ....... 140
  5.3.2 Approximate dynamic programming method 2 ....... 141
  5.3.3 Integer programming approximation ................... 141
  5.3.4 Pairwise integer programming approximation method 1  143
  5.3.5 Pairwise integer programming approximation method 2  145
  5.3.6 Empirical results ........................................ 145
  5.3.7 Bidding against a sophisticated competitor in multiple auctions 149
5.4 Multiple overlapping auctions ............................. 151
5.5 Summary and conclusions ................................. 152

6 Optimal selling in online auctions .......................... 153
  6.0.1 Literature review ........................................ 155
  6.0.2 Philosophy and contributions .......................... 156
  6.0.3 Structure of the chapter ................................ 157
6.1 Selling in sequential auctions ............................. 157
  6.1.1 The model ............................................... 157
  6.1.2 Randomness ............................................ 159
  6.1.3 Dynamics ............................................... 160
  6.1.4 Objective .............................................. 162
  6.1.5 Bellman equation ...................................... 162
  6.1.6 Estimation of parameters .............................. 163
  6.1.7 Empirical results ..................................... 164
6.2 Selling in a fixed time horizon ............................ 167
  6.2.1 The model ............................................... 167
  6.2.2 Estimation of parameters .............................. 169
6.2.3 Empirical results ........................................... 170
6.3 Multiple identical items in a fixed time interval ................. 173
  6.3.1 The model .................................................. 173
  6.3.2 Estimation of parameters .............................. 176
  6.3.3 Empirical results ....................................... 176
6.4 Summary and conclusions .................................... 177

7 Conclusions .................................................... 179

A Counterexample of monotonicity of $D(\lambda)$ ..................... 181

B Calculation of $J^D(x)$ ....................................... 183

C Proof of Theorem 13 ........................................ 185
List of Figures

3-1 Queueing network ........................................ 80
3-2 The decoupled queueing network from Example 1 ............ 87
3-3 $J(x_0)$ versus $L(x_0)$ for Example 1 .......................... 87
3-4 Queueing network from Example 2 ............................ 88

5-1 Sample empirical distribution of $q_t$ .......................... 131
5-2 Sample distribution of timing of arriving bids .................. 133

A-1 Transition probabilities ......................................... 181
List of Tables

3.1 Results of restless bandit heuristics ........................................ 77
3.2 Simulation results of policies for different queueing networks .... 89

4.1 Numerical results of bounds for constrained quadratic-cost linear systems ......................................................... 116
4.2 Numerical results of heuristics for constrained quadratic-cost linear systems ......................................................... 117

5.1 Approximation of the optimal bidding policy for Palm Pilot IIIIs with $p = 0.27$ ................................................................. 134
5.2 Approximation of the optimal bidding policy for Palm Pilot IIIIs with $p = 0.8$ ................................................................. 134
5.3 Performance of DP Policy for Palm Pilot IIIIs for a range of $p$-values ................................................................. 135
5.4 Performance of $DP$ Policy for Stamps for a range of $p$-values ..... 135
5.5 Performance of bidding strategies for Palm Pilot IIIIs .................... 136
5.6 Performance of bidding strategies for stamp collections .................... 136
5.7 Performance of bidding against Policy ‘Bid $A$ at T-1’ for Palm Pilot IIIIs, the agent’s budget is 150 ......................................................... 137
5.8 Performance of bidding against $DP$ Policy for Palm Pilot IIIIs, the agent’s budget is 150 ......................................................... 137
5.9 Performance of bidding against Policy ‘Bid $A$ at T-1’ ......................... 138
5.10 Comparison of $DP$, $ADP1$, $ADP2$ and $IPA$ for $N = 2$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III ........................................ 146
5.11 Comparison of $ADP1$, $ADP2$, $IPA$, and $PIPA1$ for $N = 3$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III ........................................ 147
5.12 Comparison of IPA and PIPA2 for $N = 6$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III .............................................. 147
5.13 Comparison of DP, ADP1, ADP2 and IPA for $N = 2$ auctions, $A = 10$, $C = 50$ and data from stamp collections .............................................. 147
5.14 Comparison of ADP1, ADP2, IPA, and PIPA1 for $N = 3$ auctions, $A = 10$, $C = 50$ and data from stamp collections .............................................. 147
5.15 Comparison of IPA and PIPA2 for $N = 6$ auctions, $A = 10$, $C = 50$ and data from stamp collections .............................................. 147
5.16 Comparison of IPA and PIPA1 for $N = 3$ auctions, $A_1 = A_2 = A_3 = A/2$, $A = 30$, $C = 10$, and Palm Pilots III data .............................................. 148
5.17 Comparison of IPA and PIPA1 for $N = 3$ auctions, $A_1 = A_2 = A_3 = A/2$, $A = 20$, $C = 50$, for stamp collections data .............................................. 149
5.18 Performance of bidding against Policy 'Bid budget at T-1 in lowest listed price auction' .............................................. 149
5.19 Bidding against Policy 'Bid budget of 140 at T-1 in lowest listed price auction' .............................................. 150

6.1 Final sale price using reserve and initial price .............................................. 165
6.2 Final sale price using reserve, initial price and BIN .............................................. 165
6.3 Final sale price using reserve and initial price .............................................. 165
6.4 Final sale price using reserve, initial price and BIN .............................................. 165
6.5 The optimal parameters and selling price for selling an item over a 14 day period .............................................. 170
6.6 The optimal parameters and selling price for selling an item over a 14 day period .............................................. 171
6.7 The optimal parameters and selling price for selling an item over a 14 day period .............................................. 171
6.8 The optimal parameters and selling price for selling an item over a 14 day period .............................................. 172
6.9 The optimal parameters and selling price for selling 1, 2, 4 and 8 items over a 21 day period .................................................. 178

A.1 $D(1; \lambda)$ for range of $\lambda$ .................................................. 182
Chapter 1

Introduction

Dynamic programming is the principal method to address dynamic sequential decision problems under uncertainty. While generally applicable, dynamic programming suffers from the curse of dimensionality. Coined by Bellman, the phrase refers to the nature of dynamic optimizations problems to explode exponentially in the dimensions of the state space and randomness. In response, researchers have paid considerable attention to approximation methods. Approximation methods typically focus on reducing the number of calculations, typically by sampling states or decisions. And while approximation methods have wide applicability, they are still hindered by the dimensionality of the problem.

In this dissertation we consider a broad class of problems that involve many independent subprocesses that are only weakly linked per time period. We call these problems weakly coupled dynamic optimization problems. This is a common structure of dynamic optimization problems and arises in various applications, including queueing networks, supply chain inventory problems, and multiarmed bandits to name a few. This dissertation develops a decomposition approach for solving weakly coupled dynamic optimization problems. In addition, we investigate the effectiveness of the approach in a number of settings.
1.0.1 An overview of the theory

Our approach is to use Lagrange multipliers to decompose weakly coupled dynamic optimization problems into a series of smaller dynamic optimization problems. Furthermore, we propose Lagrange-based policies based on the decoupled problems. The general idea of decoupling dynamic optimization problems has been realized by a few researches for specific problems. Castaño [15], van Ryzin and Ozcan [45], and Whittle [53], all suggest the concept of decoupling dynamic problems with Lagrange multipliers. Castaño [15] and Whittle [53] also suggest a Lagrange-based policy. However, Castaño [15], van Ryzin and Ozcan [45], and Whittle [53], consider only specific applications of their method and do not investigate general implications of the approach as we do. Our research is distinguished from previous research for two primary reasons: (i) We establish the general applicability of Lagrange methods to a broad class of problems, i.e., weakly coupled dynamic optimization problems; (ii) We introduce an integer optimization algorithm for producing Lagrange-based policies that are near-optimal. In particular, we study the implications of our approach in solving a number of applications including multiclass queueing networks, supply chain inventory problems, linear systems, and bidding in on-line auctions.

1.0.2 An overview of the applications

There are many interesting applications and classes of problems to which our methods apply. Indeed, the celebrated multiarmed bandit problem [51] is one such problem. Below we outline the applications that we will investigate.

The multiarmed bandit problem. Consider an infinite horizon discounted dynamic optimization problem. In every time period the controller must set one of $N$ projects active and set the remaining projects passive. For the active project, a state-dependent reward is received and its state goes through transition with known probabilities. All passive projects add zero reward and they remain static. The problem for the controller is to select which projects to set active based on the state of the system so as to maximize total expected discounted reward. The multiarmed
bandit problem has been used to model many applications, including job scheduling, sequential random sampling, clinical trials, investment in new products and random search, (see Varaiya et al. [46]).

The restless bandit problem. This is an extension of the multiarmed bandit problem in which both active and passive rewards are received. In addition, the $N$ projects, independent of one another, go through Markovian transitions for both active and passive actions. The problem is in each period to choose $K < N$ projects to set active. The modeling power of restless bandits is of course greater than that of multiarmed bandits, though at a considerable computational cost; The problem has been shown to be PSPACE-hard, (Papadimitriou and Tsitsiklis [37]). Applications include: medical treatment in which the $N$ patients’ states change if left untreated, and possibly improve if they were among the $K$ treated; the tracking of $N$ enemy submarines by $K$ aircraft; job scheduling whereby workers output decreases if not allowed to rest (see Whittle [53]).

Multiclass queueing networks. Consider a set of stations that must service the arrival of different classes of customers in discrete time. Each class of customers has its own service times and arrival rates. Given the routes customers follow through the network, the problem we consider involves determining which class of customers to work on at each station. The objective is to minimize a linear combination of the sojourn times of each customer class. This problem has been used to model packet-switching communication networks, scheduling of multi-processors and multiprogrammed computer systems, (see Avram et al. [2]).

Serial supply chain inventory problem. Consider a serial supply chain consisting of a finite number of stations. At one end of the chain there is a production station where goods are produced, and at the other end there is an outlet station where demand is received. For the case where holding costs and backorder costs are linear, there are no setup costs, and allowing for random lead times between adjacent stations, Muharremoglu and Tsitsiklis [34] show that such problems can be decomposed into single product-customer pairs. Then in the presence of inventory capacities, we show the resulting problem is a weakly coupled dynamic optimization problem.
Bidding in multiple simultaneous on-line auctions. On-line auctions have become established as a convenient, efficient, and effective method of buying and selling merchandise. One interesting feature of on-line auctions is that they allow bidders to participate in many auctions at once. Indeed, at any given time it is common to find multiple auctions for the same item proceeding simultaneously. We consider a bidder with a fixed budget bidding in multiple simultaneous on-line auctions.

1.0.3 Contributions

The main contributions of this dissertation are:

(a) We present a Lagrangian based approach to decoupling weakly coupled dynamic optimization problems for both finite and infinite horizon problems. We show how to use the decoupled problems to obtain bounds on the true value of the problem, and present two methods for computing the bounds.

(b) We present three Lagrange-based policies for weakly coupled dynamic optimization problems.

(c) We study the implications of our policies in multiarmed bandits, restless bandits, supply chain inventory problems and capacitated queueing networks.

(d) For restless bandits we establish duality connections between our Lagrangian-decoupled bounds and policies and the bounds and policies in the literature. In addition, for restless bandits we develop higher-order coupling methods for finding policies and bounds.

(e) We extend our results to non-decomposable problems and investigate our Lagrangian techniques in constrained quadratic-cost linear systems.

(f) We propose two models of on-line auctions, one from a bidder's perspective and the other from a seller's perspective, that allow for dynamic optimization by the user. These models are tractable, directly applicable by the user, and the parameters of the models can be constructed from publicly available data.
For the bidders, we provide several algorithms, based on approximate dynamic programming and integer programming algorithms, to find optimal bidding strategies in multiple simultaneous on-line auctions. The strongest of these methods is based on our Lagrangian-based decoupling heuristic. We provide computational evidence that these methods produce high quality solutions fast and reliably.

For the sellers problem in on-line auctions, we use our model of the dynamics of the auction in the context of dynamic optimization. We investigate the effectiveness of the reserve price, Buy it Now price and auction length. Our algorithm also provides a seller the means for determining the optimal number of items to put up for sale in sequential multi-unit auctions.

1.0.4 Structure of dissertation

The dissertation is structured as follows. In Chapter 2, we develop the formal theory of our decoupling approach and show our method's relation to the literature. In Chapter 3, we investigate several applications of our algorithm including multiarmed bandits, restless bandits, multiclass queueing networks and supply chain inventory problems. In Chapter 4 we extend our Lagrangian results to constrained quadratic-cost linear systems. In Chapter 5, we do an in-depth study of optimal bidding policies in on-line auctions. We develop an empirical model of on-line auctions from the bidders perspective and we show how dynamic optimization is used to solve the single auction case. We then investigate the problem of bidding in multiple simultaneous auctions. Chapter 6 is an extension of Chapter 5 devoted to optimal selling strategies in on-line auctions. Here we develop a tractable and empirically based model of the auction outcome as a function of the parameters set by the seller in an eBay style auction. We develop and algorithm for determining how to optimally sell an inventory of goods in a given time horizon. We provide our conclusions in Chapter 7.
Chapter 2

A Lagrangian decomposition approach to weakly coupled dynamic problems

2.1 Introduction

Consider a finite number of processes, each evolving independently, and each of which can be solved optimally when considered in isolation. Now suppose that there are state independent linear constraints on the controls across all processes in every time period. We define the resulting problem as a weakly-coupled dynamic optimization problem. The class of weakly-coupled dynamic optimization problems is broad in both its applicability and modeling power.

The central theme of this dissertation is the use of Lagrange multipliers to decompose weakly coupled problems into a number of smaller problems that can be solved independently of one another. Specifically, consider $N$ independent projects each yielding rewards dependent only on their individual controls and state. Furthermore, state transitions corresponding to each project are dependent only on each project’s own state and controls. At each period there are linear constraints connecting the controls of the individual projects; these constraints link the projects through
a common budget of constraints. In order to solve this problem we take a Lagrangian approach by relaxing the constraints with state independent multipliers. The result is that the \( N \)-dimensional optimization problem decouples into \( N \) one-dimensional subproblems. The key is in choosing the values of multipliers and in transforming the optimal controls of the decoupled problems into a feasible solution in the current period. In this chapter we establish the following:

1. For both finite and infinite horizon weakly coupled dynamic optimization problems, Lagrangian relaxation of the controls yields independent, convex and piecewise linear dynamic subproblems. Moreover, the Lagrangian function provides a bound on the true cost to go.

2. We provide two methods determining the optimal value of the Lagrangian function; one based on linear programming, the other based on a stochastic subgradient method.

3. We present three Lagrange-based algorithms for determining feasible policies.

### 2.1.1 Formulation

We begin by defining the general weakly coupled dynamic optimization problem in terms of the state, controls, randomness, dynamics, reward, constraints and objective.

**Definition 1** Let \( \mathcal{N} = \{1, \ldots, N\} \). A dynamic optimization problem is weakly coupled if it can be formulated as follows.

**State** The state at time \( t \) is \( \mathbf{x}_t = (x^1_t, \ldots, x^N_t) \), where \( x^i_t \in X^i_i, \ |X^i_i| \) is finite.

**Controls** The controls at time \( t \) are \( \mathbf{u}_t = (u^1_t, \ldots, u^N_t) \), where \( u^i_t \in U^i_t, \ |U^i_i| \) is finite.

**Randomness** The randomness realized per period is \( \mathbf{q}_t = (q^1_t, \ldots, q^N_t) \) with known distribution, and where \( q^i_t \in Q^i, \ |Q^i| \) is finite. Furthermore,

\[
P(q_t = q | x_t, u_t) = \prod_{i=1}^N P(q^i_t = q^i | x^i_t, u^i_t).
\]
Dynamics The dynamics can be written as

\[ x_{t+1} = f(x_t, u_t, q_t) = (f_1(x_t^1, u_t^1, q_t^1), \ldots, f_N(x_t^N, u_t^N, q_t^N)). \]

For infinite horizon problems we will often denote by \( y = f(x, u, q) \), the random state after one transition.

Remark It follows that the distribution of the next state satisfies

\[ P(x_{t+1} = y|x_t, u_t) = \prod_{i=1}^{N} P(x_{t+1}^i = y^i|x_t^i, u_t^i). \]

Reward The reward received per period is \( R(x_t, u_t) = \sum_{i=1}^{N} R^i(x_t^i, u_t^i) \).

Constraints There are constraints linking the controls in each period. That is,

\[ Au_t \leq b. \]

Note that \( A \) is a matrix, \( b \) is a vector, and both are independent of state \( x_t \).

The constraints can be expressed as

\[ \sum_{i=1}^{N} A_i u_t^i \leq b. \]

In the finite horizon case we allow constraints \( A_i u_t \leq b_t \), i.e., time varying.

We will often refer to this set of constraints as budget constraints. To avoid redundancy with constraint space \( U^i \), we require that every row of \( A \) have nonzero components in at least two of \( A_i, A_j \), for \( i, j \in N \).

Objective The problem is either finite horizon or infinite horizon with discount factor \( \beta < 1 \). For the finite horizon problem with \( T \) periods the objective is

\[ \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{T} \sum_{i=1}^{N} R^i(x_t^i, \pi_t^i) \right], \]
for some feasible policy $\pi$. For infinite horizon problems the objective is

$$\max_{\pi} \mathbb{E}\left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \beta^t R_t^i(x_t^i, \pi_t^i) \right].$$

Having formally defined weakly coupled dynamic optimization problems, we observe that the structure is fairly specific. However, there are numerous applications that can be modeled in this way including multiarmed bandits, queueing networks, serial supply chain problems and optimal bidding in online auctions.

It is important to realize that we are really dealing with $N$ independent dynamic optimization problems that are linked by a set of budget constraints. The key assumption we will make throughout this thesis is that were it not for the budget constraint then each of the $N$ problems could be solved in isolation in a tractable way. However, because of the budget constraints we cannot consider these problems independently. As a result, the exact solution requires the solution of a high dimensional problem that is in general intractable even if $N = 2$. For this reason we will attempt to decouple the $N$ problems in such a way that accurate estimates of the cost to go are achieved, and near optimal decisions of what action to take are made.

### 2.1.2 Literature review

The key structure of the problem we investigate involves controls that are weakly coupled. In particular application contexts, Castaño [15], Whittle [53], and van Ryzin and Ozcan[45] have all, independently, investigated the role of Lagrangian relaxation to aid in the solution. Castaño [15] provides a decoupling approach to the dynamic scheduling of multi-mode sensor resources for the problem of classification of multiple unknown objects. In this finite horizon problem, the author relaxes expected constraint compliance and shows the problems decompose into smaller problems. In a similar manner, Whittle [53] relaxes expected constraint compliance for the time-averaged restless bandit problem, and notes that a similar method can be used for discounted bandits. Moreover, he makes the connection for the time av-
anged case that selecting Gittins index as a multiplier yields an optimal heuristic for the multiarmed bandit, a result that we establish for the discounted bandit. Van Ryzin and Ozcan [45] use a Lagrangian based approach to decouple an airline revenue management problem, wherein fleets are assigned to routes in response to demand. Meuleau et al. [31] solve weakly coupled Markov decision problems by ignoring global constraints, and solving the subproblems independently.

Two of the significant results that distinguish this research from previous research are: (i) The realization of the general applicability of Lagrangian decoupling to a class of problems, i.e., weakly coupled dynamic optimization problems; (ii) An integer optimization algorithm for determining a feasible policy based on the decoupled problems.

We have loosely defined the key structure of the problems we investigate as being weakly coupled. Bertsimas and Niño-Mora [11] define a decomposable class of problems. Decomposable problems are problems that can be decoupled by Lagrangian relaxation of the linking constraints. Bertsimas and Niño-Mora [11] also define a class of problems as being indexable if solving the decoupled problems independently can still lead to an optimal algorithm.

### 2.1.3 Structure of the chapter

The chapter is structured as follows. In Section 2.2, we present a Lagrangian method for decoupling infinite horizon weakly coupled dynamic optimization problems. We provide structural results of the Lagrangian function and provide two methods for calculating the optimal Lagrangian value. In Section 2.3, we present a Lagrangian method for decoupling finite horizon weakly coupled dynamic optimization problems. Here too we provide structural results of the Lagrangian function and provide two methods for calculating the optimal Lagrangian value. Finally, in Section 2.4, we present three Lagrange-based policies for infinite horizon problems and present candidates for the value of the Lagrange multipliers.
2.2 Infinite horizon weakly coupled discounted optimization problems

We begin by restating the objective of infinite horizon weakly coupled dynamic optimization problems.

\[
\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \beta^t R^i(x^*_t, \pi^*_t) \right].
\]

The standard approach for solving problems such as this involves solving the Bellman equation for every state \( \mathbf{x} \). As shown by Blackwell [13], stationary policies are optimal. We will often suppress the notion of time by referring to the current state as \( \mathbf{x} \) and the next state as \( \mathbf{y} \). The Bellman equation is

\[
J(x) = \max_{u^1, \ldots, u^N} \left\{ \sum_{i=1}^{N} R^i(x^i, u^i) + \beta \mathbb{E}[J(y)|x, u] A u \leq b \right\}. \tag{2.1}
\]

Note that to determine the cost to go for starting state \( \mathbf{x} \), one must solve Eq. (2.1) for all \( \mathbf{x} \in X \). Typically this is achieved using either policy iteration or value iteration, see Bertsekas [7].

Again, as stated earlier, such problems are of high dimensionality and are typically intractable. Were it not for the coupling constraints the \( N \) problems could be optimized independently. Thus our approach to making the problems easier involves decoupling the controls with Lagrange multipliers \( \lambda \geq 0 \). At this point it will be necessary to make explicit reference to time. The current state at time \( t = 0 \) is \( x_0 \). Our approximation of the cost to go is found by solving

\[
\tilde{J}(x_0; \lambda) = \max_{u^0_0, \ldots, u^N_0} \left\{ \sum_{i=1}^{N} R^i(x^0_i, u^0_i) + \beta \mathbb{E}[J(x_1)|x_0, u_0] + \lambda'(Au_0 - b) \right\}. \tag{2.2}
\]

Then \( \tilde{J}(x_0; \lambda) \) is an estimate of the true cost to go \( J(x_0) \). It involves solving Eq. (2.2) for state \( x_0 \), in addition to Eq. (2.1) for all \( \mathbf{x} \in X \).

Note that \( \min_{\lambda \geq 0} \tilde{J}(x_0; \lambda) \geq J(x_0) \). Thus, the minimizing \( \lambda \) is dependent on state
\( x_0 \). With this in mind, we consider attaching a vector of Lagrange multipliers \( \lambda_x \) to the constraints of Eq. (2.1) for every vector \( x \) of the state space.

**Definition 2** For infinite horizon problems, define

\[
L(x; \lambda_x) = \max \sum_{i=1}^{N} \left( R^i(x^i, u^i) - \lambda_x' A_i u^i \right) + \beta E[L(y; \lambda_y)|x, u] + \lambda_x' b
\]

s.t. \( u^i \in U^i, \forall i \in \mathcal{N} \).

In the above definition, transition probabilities and rewards are the same as in the exact formulation of \( J(x) \). The corresponding dual problem is to \( \min_{\lambda_x \geq 0} L(x; \lambda_x) \).

Though the Lagrangian problem is now unconstrained, finding the optimal dual multipliers for every state is no easier than the original problem. However, if we set \( \lambda_x = \lambda \) for all \( x \), the decomposes as the next theorem illustrates.

**Theorem 1** Let \( \lambda_x = \lambda \) for all \( x \). Then

\[
L(x; \lambda) = \sum_{i=1}^{N} L^i(x^i; \lambda) + \lambda' b - \frac{1}{1 - \beta}; \tag{2.3}
\]

where

\[
L^i(x^i; \lambda) = \max_{u^i \in U^i} \left\{ R^i(x^i, u^i) - \lambda' A_i u^i + \beta E[L^i(y^i; \lambda)|x^i, u^i] \right\}. \tag{2.4}
\]

**Proof.** We define

\[
V_0(x; \lambda) = \sum_{i=1}^{N} \max_{u^i \in U^i} \left\{ R^i(x^i, u^i) - \lambda' A_i u^i \right\} + \lambda' b
\]

\[
= \sum_{i=1}^{N} V_0^i(x^i; \lambda) + \lambda' b.
\]

The subscript of \( V_0(\cdot) \) represents that we are only relaxing constraints of the imme-
\[ V_1(x; \lambda) = \max_{u \in U} \sum_{i=1}^{N} (R_i(x^i, u^i) - \lambda' A_i u^i) + \lambda' b + \beta E[V_0(y; \lambda)\mid x, u] \]
\[ = \max_{u \in U} \sum_{i=1}^{N} (R_i(x^i, u^i) - \lambda' A_i u^i + \beta E[V_0^i(y; \lambda)\mid x, u]) + \lambda' b(1 + \beta) \]
\[ = \sum_{i=1}^{N} \max_{u^i \in U^i} \left\{ R_i(x^i, u^i) - \lambda' A_i u^i + \beta E[V_0^i(y^i; \lambda)\mid x^i, u^i] \right\} + \lambda' b(1 + \beta) \]
\[ = \sum_{i=1}^{N} V_1^i(x^i; \lambda) + \lambda' b(1 + \beta). \]

Continuing in this manner we set

\[ L(x; \lambda) = \lim_{n \to \infty} V_n(x; \lambda) \]
\[ = \lim_{n \to \infty} \sum_{i=1}^{N} V_n^i(x^i; \lambda) + \lambda' b(1 + \beta + \beta^2 + \cdots) \]
\[ = \sum_{i=1}^{N} L_i(x^i; \lambda) + \lambda' b \frac{1}{1 - \beta}, \]

where \( L_i(\cdot) \) satisfy Eq. (2.4). ■

The key result of Theorem 1 is that \( L(x; \lambda) \) is the sum of \( N \) maximization problems of considerably smaller size that can be solved in isolation of one another. What links the \( N \) problems together are \( \lambda \). Let

\[ L(x; \lambda) = \sum_{i=1}^{N} L_i(x^i; \lambda) + \lambda' b \frac{1}{1 - \beta}, \]

and

\[ L(x) = \min_{\lambda \geq 0} L(x; \lambda). \]

**Theorem 2** We have that

(a) \( L(x) \geq J(x) \),
(b) \( L(x; \lambda) \) is convex and piecewise linear in \( \lambda \).

Proof. The inequality \( L(x) \geq J(x) \) and the convexity of \( L(x; \lambda) \) follow from standard Lagrangian theory (see Bertsekas [8]). To see the second half of Part (b), we write \( L(x) \) as the solution to a linear program, which can be achieved according the standard theory for infinite horizon discounted dynamic optimization problems. Let \( R_{x,i}^{u,i} = R(x^i, u^i) \), \( \alpha_{x,i} = 1(x^i = x_0^i) \), and \( p_{x^i,y^i}^{u,i} = P(y^i|x^i, u^i) \). The decision variables are \( \lambda \) and the cost to go \( L_{x,i}^i \) for every state \( x^i \). The linear program follows.

\[
L(x) = \min_{L_{x,i}^i, \lambda} \sum_{i \in \mathcal{N}} \sum_{x^i \in \mathcal{X}^i} \alpha_{x,i} L_{x,i}^1 + \frac{\lambda \beta}{1 - \beta} \\
\text{s.t.} \quad L_{x,i}^i \geq R_{x,i}^{u,i} - \lambda A_i u_i + \beta \sum_{y^i \in \mathcal{X}^i} p_{x^i,y^i}^{u,i} L_{y,i}^i, \quad (2.5) \\
\forall u^i \in U^i, \forall x^i \in \mathcal{X}^i, \forall i \in \mathcal{N}, \\
\lambda \geq 0.
\]

This is a linear program with a finite number of extreme points. For a particular \( \lambda \geq 0 \), \( L(x; \lambda) \) can be calculated using the same linear program with the additional constraint \( \lambda = \lambda \). It follows from standard theory (see Bertsimas and Tsitsiklis [9]) that \( L(x; \lambda) \) is a piecewise linear function of \( \lambda \). \( \blacksquare \)

In Section 2.4, and in Chapter 3, we will exploit these properties to obtain policies infinite horizon weakly coupled problems.

### 2.2.1 Computation of optimal Lagrangian bound for infinite horizon problems

In this section we present two methods for calculating \( L(x) \) for infinite horizon problems.

#### Linear programming

As shown in the proof of Theorem 2, \( L(x) \) can be solved using linear programming. Typically, policy iteration is preferred over linear programming for solving infinite
horizon dynamic programs because it takes advantage of the special structure of the problem. In our case however, there are the additional continuous decision variables $\lambda$. Thus, the approach of iteratively choosing $\lambda$ and solving the decoupled problems requires a significant amount of computation if not done carefully (see the next subsection on a stochastic subgradient approach). Linear programming allows one to consider the global problem of finding $\lambda$ and the values of the decoupled problems all in one formulation. This makes the linear program an attractive method. Were it not for $\lambda$, Problem (2.5) could be solved as $N$ independent problems. As it is, there are $O(N|X^i| + |\lambda|)$ decision variables and $O(N|X^i||U^i|)$ constraints.

**Stochastic subgradient method**

Another approach to calculating $L(x)$ is using a subgradient method. This is particularly useful when $|U^i|$ and $|X^i|$ are large, making linear programming intractable. Recall that our goal is to solve

$$L(x) = \min_{\lambda} L(x; \lambda)$$

$$\text{s.t. } \lambda \geq 0.$$  \hspace{1cm} (2.6)

Because $L(x; \lambda)$ is piecewise linear, our approach involve a subgradient algorithm (see Bertsekas [8]). The problem is further complicated by the fact that calculating $\nabla_{\lambda} L(x; \lambda)$ is not easy. Therefore, we will use a simulation based method for estimating the gradient (see Bertsekas [7]). The result is a stochastic subgradient method.

The subgradient method involves iteratively solving the equation

$$\lambda^{k+1} = [\lambda^k + \gamma^k g^k]^+,$$

where $g^k$ is the subgradient $\nabla L(x; \lambda^k)$, and $\gamma^k$ is a diminishing positive step size at
the $k$th iteration of the algorithm. For example, we can set

$$
\gamma^k = \frac{\alpha^k (\hat{L}(x; \lambda^k) - L(x; \lambda^k))}{\|g^k\|^2},
$$

where $0 < \alpha^k < 2$ and

$$
\hat{L}(x; \lambda^k) = \min_{0 \leq j \leq k} L(x; \lambda^j).
$$

The aim of the subgradient method is to approximate the solution of Problem (2.6) with the sequence $\lim_{k \to \infty} L(x; \lambda^k)$. The algorithm terminates when $\lambda^k = 0$ or $g^k$ is sufficiently small.

For our problem, $\nabla L(x; \lambda^k)$ can be difficult to determine exactly. The following simulation based approach can be used to estimate the gradient. In solving $L'(x^i; \lambda)$ for all $i$, we will have obtained a policy $\pi^i(\lambda)$. Consider simulating the process based on the policy $\pi^i(\lambda)$ for a sample path consisting of $S$ time stages. At every point in time we keep track of the values $A_i, u_i^c$, obtained from the policy; summing the discounted value of these will serve as a sample-path dependent estimate of the true gradient. We will index simulations with $m = 1, \ldots, M$. For simulation $m$ of length $S$, let the estimate of the gradient be denoted by

$$
\nabla^S_m L(x; \lambda^k) = \sum_{i=1}^{N} \nabla^S_m L'(x^i; \lambda^k) + b \frac{1}{1 - \beta}.
$$

Note that the simulations for $i = 1, \ldots, N$ can be done in parallel. The stochastic estimate of the subgradient $g^k \approx \nabla L(x; \lambda^k)$ is

$$
g^k = \frac{1}{M} \sum_{m=1}^{M} \nabla^S_m L(x; \lambda^k).
$$
2.3 Finite horizon weakly coupled discounted optimization problems

In this section we consider the finite horizon version of weakly coupled dynamic optimization problems. The results in this section mirror the results of Section 2.2. Namely, we show that decoupling problems results in a method of quickly achieving bounds on the cost to go. In this section we allow for time dependent constraints and write \( U_t \) and \( A_t u_t \leq b_t \) where

\[
A_t u_t = \sum_{i=1}^{N} A_t^i u_t^i.
\]

In addition, rewards are time varying and are written as \( R_t^i(x_t^i, u_t^i) \). We begin by stating the objective of finite horizon weakly coupled dynamic optimization problems.

\[
\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{T} \sum_{i=1}^{N} R_t^i(x_t^i, \pi_t^i) \right].
\]

The value of the above problem is denoted by \( J_0(x_0) \), and it can be obtained by recursively solving the Bellman equation starting at time \( T \). In particular,

\[
J_T(x_T) = \max_{u_T \in U_T, \pi_T \in \pi_N} \left\{ \sum_{i=1}^{N} R_T^i(x_T^i, u_T^i) \mid A_T u_T \leq b_T \right\}
\]

(2.7)

\[
J_t(x_t) = \max_{u_t \in U_t, \pi_t \in \pi_N} \left\{ \sum_{i=1}^{N} R_t^i(x_t^i, u_t^i) + \mathbb{E}[J(x_{t+1}) \mid x_t, u_t] \mid A_t u_t \leq b_t \right\}.
\]

(2.8)

We proceed by attaching Lagrange multipliers to Eqs. (2.7) and (2.8).

**Definition 3** Let \( \lambda_t^{x_t} \) be a vector of Lagrange multipliers associated with state \( x_t \) at time \( t \). Let \( \lambda_t^{x_t} = (\lambda_t^{x_t})_{x_t \in X_t} \). For finite horizon problems, the Lagrangian function
is defined by

\[
L_t(x^i_t; \lambda^x_t, \lambda^{x_{t+1}}_t, \ldots, \lambda^x_T) = \max \sum_{i=1}^N (R_t^i(x^i_t, u^i_t) - \lambda_t^{x^i_t} A_t^i u^i_t) + \beta E[L_{t+1}(x^i_{t+1}; \lambda_t^{x_{t+1}}, \ldots, \lambda_t^x)]|x_t, u_t] + \lambda_t^{x^i_t} b_t
\]

s.t. \( u^i_t \in U^i_t, \forall i \in \mathcal{N} \),

where \( L_{T+1}(\cdot) = 0 \).

Similar to the infinite horizon case, we will enforce the vector of Lagrange multipliers to be constant across all states, though they are permitted to change over time.

**Theorem 3** Set \( \lambda^x_s = \lambda_s \) for all \( x_s \in X_s, s = t, \ldots, T \) and define \( \lambda_{t:T} = (\lambda_t^x, \ldots, \lambda_T^x) \). Then

\[
L_t(x^i_t; \lambda_{t:T}) = \sum_{i=1}^N L_t^i(x^i_t; \lambda_{t+1:T}) + \sum_{s=t}^T \lambda^x_s b_s,
\]

where

\[
L_t^i(x^i_t; \lambda_{t:T}) = \max_{u^i_t \in U^i_t} R_t^i(x^i_t, u^i_t) - \lambda_t^x A_t^i u^i_t + E[L_{t+1}(x^i_{t+1}; \lambda_{t+1:T})|x_t, u_t],
\]

and \( L_{T+1}(\cdot) = 0 \).

**Proof.** We prove the case for \( t = T \) and proceed by induction.

\[
L_T(x_T; \lambda_T) = \max_{u_T \in U_T} \sum_{i=1}^N (R_T^i(x^i_T, u^i_T) - \lambda_T^x A_T^i u^i_T) + \lambda_T^x b_T,
\]

\[
= \sum_{i=1}^N \left( \max_{u_T^i \in U_T^i} R_T^i(x^i_T, u^i_T) - \lambda_T^x A_T^i u^i_T \right) + \lambda_T^x b_T
\]

\[
= \sum_{i=1}^N L_T^i(x^i_T, \lambda_T) + \lambda_T^x b_T.
\]
Suppose the result is true for time \( t + 1 \). Then at time \( t \) we have

\[
L_t(x_t; \lambda_t) = \max_{u_{i,t} \in U_t, \gamma_{i,t} \in A_t} \sum_{i=1}^{N} \left( R_i^t(x_t^i, u_t^i) - \lambda_t^i A_i u_t^i \right) + \mathbb{E}[L_{t+1}(x_{t+1}; \lambda_{t+1,T})|x_t, u_t]
+ \lambda_t^i b_t
\]

\[
= \max_{u_{i,t} \in U_t, \gamma_{i,t} \in A_t} \sum_{i=1}^{N} \left( R_i^t(x_t^i, u_t^i) - \lambda_t^i A_i u_t^i \right) + \mathbb{E}[L_{t+1}^i(x_{t+1}^i; \lambda_{t+1,T})|x_t^i, u_t^i]
+ \lambda_t^i b_t
\]

\[
= L_t^i(x_t^i; \lambda_{t+1,T}) + \sum_{s=t}^{T} \lambda_t^i b_s.
\]

We define

\[
L_t(x_t) = \max_{\lambda_t, \gamma_t \geq 0} L_t(x_t; \lambda_{t,T}). \quad (2.9)
\]

**Theorem 4** We have that

(a) \( L_t(x_t) \geq J_t(x_t) \),

(b) \( L_t(x_t; \lambda_{t,T}) \) is convex and piecewise linear in \( \lambda_{t,T} \).

**Proof.** The inequality \( L_t(x) \geq J_t(x) \) and the convexity of \( L_t(x; \lambda) \) follow from standard Lagrangian theory (see Bertsekas [8]). For the second half of Part (b), we use induction. Note that \( L_T(x_T; \lambda_T) = \sum_{i=1}^{N} L_T^i(x_T^i; \lambda_T) + \lambda_T^i b_T \) is piecewise linear in \( \lambda_T \) for all \( x_T \). Suppose \( L_{t+1}(x_{t+1}; \lambda_{t+1,T}) \) is piecewise linear. Then \( \mathbb{E}[L_{t+1}(x_{t+1}; \lambda_{t+1,T})|x_t, u_t] \) is also piecewise linear. We have

\[
L_t(x_t; \lambda_t) = \max_{u_{i,t} \in U_t, \gamma_{i,t} \in A_t} \sum_{i=1}^{N} \left( R_i^t(x_t^i, u_t^i) - \lambda_t^i A_i u_t^i \right) + \mathbb{E}[L_{t+1}(x_{t+1}; \lambda_{t+1,T})|x_t, u_t]
+ \lambda_t^i b_t
\]

is piecewise linear in \( \lambda_{t,T} \). \( \blacksquare \)
2.3.1 Computation of Lagrangian bound for finite horizon problems

In this section we present two methods for computing $L_0(x_0)$. Our approaches will be similar to the two methods proposed in Section 2.2.1.

**Linear programming**

In this section, we present a linear program for calculating $L_0(x_0)$. Let $R^i_{x_0} = R^i(x_0, u_0^i)$, $\alpha_{x_0} = 1$ if $x_0$ is the starting state of project $i$, and zero otherwise. In addition, we write $P_{x_t|y_{t+1}}^i = P(y_{t+1}|x_t, u_t^i)$ for $t = 0, \ldots, T - 1$. The decision variables are $\lambda_0, \ldots, \lambda_T$, and $U_{x_t}^i$ for all states $x_t^i \in X_t^i$, $t = 1, \ldots, T$, $i \in N$. The linear program follows.

\[
L_0(x_0) = \min_{L_{x_t}^i, \lambda_0, T} \sum_{i \in N} \sum_{x_0^i \in X_0^i} \alpha_{x_0} L_{x_0}^i + \sum_{t=0}^{T} \lambda_t b_t
\]

subject to:

\[
L_{x_t}^i \geq R_{x_t}^i - \lambda_t A_t^i u_{T},
\]

\[\forall u_t^i \in U_t^i, \forall x_t^i \in X_t^i, \forall i \in N\]

\[
L_{x_t}^i \geq R_{x_t}^i - \lambda_t A_t^i u_t^i + \beta \sum_{y_{t+1} \in X_{t+1}^i} P_{x_t|y_{t+1}}^i L_{y_{t+1}}^i, \]

\[\forall u_t^i \in U_t^i, \forall x_t^i \in X_t^i, \forall i \in N, t = 0, \ldots, T - 1,\]

\[
\lambda_t \geq 0, \; t = 0, \ldots, T.
\]

**Stochastic subgradient methods**

A stochastic subgradient method can be used to find $L_0(x_0)$ in a way similar to that proposed in Section 2.2.1. The primary difference is that in estimating the subgradient $\nabla L_0(x_0; \lambda_{0,T}^k)$ at iteration $k$ of the algorithm, sample paths of length $S$ are replaced by sample paths of length $T$. The $m$th estimate of the gradient at point $\lambda_{0,T}^k$ is denoted by

\[
\nabla_m L_0(x_0; \lambda_{0,T}^k) = \sum_{i=1}^{N} \nabla_m L_0^i(x_0; \lambda_{0,T}^k) + \sum_{t=0}^{T} b_{0,T},
\]
where $b'_{0,T} = (b'_0, \ldots, b'_T)$. Then

$$g^k = \frac{1}{M} \sum_{m=1}^{M} \nabla_m L_0(x; \lambda^k_{0,T}),$$

and

$$\lambda^k_{0,T} = [\lambda^k_{0,T} + \gamma^k g^k]^+.$$

We can easily reduce the dimensionality of the Lagrangian problem (2.9) by assuming $\lambda_t = \lambda$ for all $t$ and solving

$$\hat{L}(x_0) = \min_{\lambda \geq 0} L_0(x_0, \lambda).$$

In practice we have found little difference between $L(x_0)$ and $\hat{L}(x_0)$.

### 2.4 Lagrange-based policies for infinite horizon problems

In Sections 2.2 and 2.3, we investigated the results of the Lagrangian relaxation of constraints in certain classes of dynamic optimization problems. We have shown that by setting the Lagrange multipliers constant across all realized states that the resulting problem is much easier to solve. In addition, we have shown that the Lagrange-relaxed problem provides a bound on the cost to go. Our ultimate aim is find a feasible policy derived from our Lagrangian approach. Loosely speaking, our approach is to relax future constraints so that current decisions for each subproblem affect the states and costs of its own subproblem. In this section, we develop three methods for determining a feasible policy based on the Lagrangian decoupled problems. In addition, for the first two methods we suggest two values of the Lagrange multipliers for use their respective algorithms. It should be noted that similar approaches can be taken for finite horizon problems.
We motivate our first two methods by considering the exact solution of the Bellman equation:

\[
J(x) = \max_u \left\{ \sum_{i=1}^{N} R^i(x^i, u^i) + \beta \mathbb{E}[J(y)|x, u] | Au \leq b \right\}.
\]

We assume that the above problem is intractable. In particular, the value of \( \mathbb{E}[J(y)|x, u] \) is difficult to determine. However, as shown in Theorem 1, we have a means of estimating \( J(y) \) quickly using Lagrangian relaxation. Namely, we write

\[
\mathbb{E}[J(y)|x, u] \leq \mathbb{E}[L(y; \lambda)|x, u] = \sum_{i=1}^{N} \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i] + \lambda' b \frac{1}{1 - \beta} \tag{2.10}
\]

Given the estimates \( L^i(y^i; \lambda) \) for a fixed \( \lambda \), we consider the policy

\[
u(x; \lambda) = \arg \max \left\{ \sum_{i=1}^{N} (R^i(x^i, u^i) + \beta \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i] | Au \leq b \right\}. \tag{2.11}
\]

Note that we have removed the constant \( \beta \lambda' b \frac{1}{1 - \beta} \) from Eq. (2.11) since it does not affect \( u(x; \lambda) \). Given \( \lambda \), Eq. (2.11) involves solving a nonlinear separable optimization problem over linear constraints. The interpretation of Eq. (2.11) is that we are ignoring future constraints at a penalty of \( \lambda' A_i \) per decision.

### 2.4.1 Integer optimization

Our first method involves finding an estimate of the cost to go in project \( i \) for every decision \( u^i \). For a given state \( x \) and \( \lambda \), let

\[
g^i_x(u^i) = R^i(x^i, u^i) - \lambda' A_i u^i + \beta \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i].
\]
For every decision vector $u^i$ we associate a binary decision variable $z_{u^i}^i$. Given $g_{x^i}(u^i)$ for all $u^i \in U^i$ for all $i$, we solve

$$\begin{align*}
\max & \sum_{i=1}^{N} g_{x^i}(u^i)z_{u^i}^i \\
\text{s.t.} & A \left( \sum_{i=1}^{N} \sum_{u^i \in U^i} u^i z_{u^i}^i \right) \leq b \\
& \sum_{u^i \in U^i} z_{u^i}^i = 1 \\
& z_{u^i}^i \in \{0, 1\}. 
\end{align*}$$

(2.12)

### 2.4.2 Piecewise approximations

Our second Lagrange-based policy uses a quadratic approximation of the cost to go to determine a feasible policy. This method is used in cases where the integer optimization formulation (2.12) is too large to be calculated, i.e., when $|U^i|$ is large. Our approach involves the quadratic approximation

$$u^i Q_i u^i + f_i u^i + c_i \approx R_i(x^i, u^i) + \beta E[L^i(y^i; \lambda)] |x^i, u^i|.$$

Note that $Q_i$, $f_i$, and $c_i$ are functions of $\lambda$, and $Q_i$ is positive semidefinite. Then the decision made at time $i$ is the solution of

$$\begin{align*}
\max & \sum_{i=1}^{N} u^i Q_i u^i + f_i u^i \\
\text{s.t.} & \sum_{i=1}^{N} A_i u^i \leq b \\
& u^i \in U^i.
\end{align*}$$

### 2.4.3 Choice of Lagrange multipliers

In Sections 2.4.1 and 2.4.2, we introduced methods for choosing a feasible decision vector $u$ based on the information from the decoupled problems for a given vector of Lagrange multipliers $\lambda$. In the next two sections we investigate two choices of $\lambda$. 
Choice of $\lambda$ 1: Set $\lambda$ equal to zero

The first approach is to simply set $\lambda = 0$ for all future constraints. The decision in the current period is

$$u(x; 0) = \arg \max \left\{ \sum_{i=1}^{N} \left( R^i(x^i, u^i) + \beta E[L^i(y^i; 0)|x^i, u^i] \right) \mid Au \leq b \right\}.$$ 

The resulting problem can be solved by integer optimization. The interpretation of this approach is that we are completely ignoring future constraints. In addition, the quantities $R^i(x^i, u^i) + \beta E[L^i(y^i; 0)|x^i, u^i]$ can be computed by fast methods such as policy iteration.

Choice of $\lambda$ 2: Set $\lambda$ equal to $\arg \min$ of Lagrange function

In this section, we choose $\lambda$ based on the solution of $L(x)$. We begin with the assertion that our aim is to find a good approximation of $E[I(y)|x, u]$ that is easy to solve. One method involves finding

$$E \left[ \min_{\lambda \geq 0} L(y; \lambda) \big| x, u \right]. \quad (2.13)$$

Un fortunately this problem is not decomposable into $N$ problems since the minimizing $\lambda$ depends on random state $y$, which is dependent on decision $u$.

We need to choose a static $\lambda$ in this period so that future constraints decouple. In the previous section we selected $\lambda = 0$. Our approach in this chapter is to propose the minimizing $\lambda$ in this period as a candidate. Consider the problem

$$\min \ L(x_0; \lambda) \quad (2.14)$$

$$\text{s.t.} \quad \lambda \geq 0.$$ 

Let $\lambda^*$ be the minimizing Lagrange multiplier. Here $\lambda^*$ is an approximation of the solution to Eq. (2.13) under every realization of $y$. The feasible decision made in this
period is found by solving

\[
\mathbf{u}(\mathbf{x}; \lambda^*) = \arg \max \left\{ \sum_{i=1}^{N} (R^i(x^i, \mathbf{u}^i) + \beta \mathbb{E}[L^i(y^i; \lambda^*) | x^i, \mathbf{u}^i]) \mid A\mathbf{u} \leq b \right\}. \tag{2.15}
\]

In what follows, we give an integer programming approach to computing Eq. (2.15). First, we solve \(L(x)\) using the linear program (2.5). Note that the quantity

\[
R^i_{x_0^i} + \beta \sum_{y^i \in X^i} \rho^i_{x_0^i y^i} L_{y^i} - \lambda' A_i \mathbf{u}^i
\]

is the cost to go in project \(i\) given we make decision \(\mathbf{u}^i\) now, and make optimal decisions in the decoupled problem thereafter. Let \(\rho^i_{x_0^i}\) be the slack variables associated with state \(x_0^i\) in Problem (2.5). Then

\[
\rho^i_{x_0^i} = L_{x_0^i} - (R^i_{x_0^i} + \beta \sum_{y^i \in X^i} \rho^i_{x_0^i y^i} L_{y^i} - \lambda' A_i \mathbf{u}^i). \tag{2.16}
\]

Let \(z^i_{u^i} = 1\) if control \(u^i\) is applied at time \(t = 0\) and zero otherwise. It follows that \(\mathbf{u}(\mathbf{x}; \lambda^*)\) solves the following integer program.

\[
\begin{align*}
\min & \sum_{i=1}^{N} \rho^i_{x_0^i z^i_{u^i}} \\
\text{s.t.} & A \left( \sum_{i=1}^{N} \sum_{u^i \in U^i} u^i z^i_{u^i} \right) \leq b \\
& \sum_{u^i \in U^i} z^i_{u^i} = 1 \\
& z^i_{u^i} \in \{0, 1\}.
\end{align*} \tag{2.17}
\]

Then the policy is found by minimizing the decoupled bound on the cost to go, and finding controls \(u\) that minimizes the sum of the slack variables associated with state \(x_0^i\). It can be seen that Problem (2.17) is the same as Problem (2.12) but with a cost vector defined by \(\lambda = \lambda^*\).
2.4.4 Minimal-lambda feasible policy

The Lagrange-based heuristics of Sections 2.4.1 and 2.4.2 involved finding a feasible decision with reward estimates based on the decoupled problems. That is, first $\lambda$ is chosen, and then constraints are enforced. In this section we take a slightly different approach. The method detailed in this section is to find the minimum Lagrange multipliers such that the current decision is feasible. Let

$$D^i(x^i; \lambda) = \arg \max_{x^i \in \mathcal{U}^i} L^i(x^i; \lambda),$$

(2.18)

and let $D(x; \lambda) = (D^1(x^1; \lambda), \ldots, D^N(x^N; \lambda))$. For a given vector $a$, the policy we propose involves solving

$$\min \ a' \lambda$$

s.t.  $AD(x; \lambda) \leq b$

$$\lambda \geq 0,$$

(2.19)

and setting $u = D(x; \lambda)$. The choice of cost vector $a$ is what defines the policy. We will see in Chapter 3 that such policy can yield optimal or near optimal results. In particular, for multiarmed bandits in Section 3.1, setting $a = 1$ yields the true optimal policy. For restless bandits, Section 3.2, $a = 1$ yields a near optimal policy as well.

The drawbacks of this approach include that the constraint set can be nonlinear, and minimizing $a' \lambda$ subject to the constraints can be difficult. Moreover, there is no guarantee that a feasible solution even exists (see Appendix A for an example of the restless bandit problem where this problem occurs). Assuming Problem (2.19) is feasible, or one has a method dealing with best infeasible solutions, we present a method for limiting $\lambda$ to a ray, rather than the positive orthant. In particular, let $\lambda_r = \lambda_0 + r \nu$ where $r$ is a scalar and $\nu$ and $\lambda_0$ have the same dimension as $\lambda$.  

45
Problem (2.19) becomes

\[
\min_{r} \quad r \alpha' \nu \\
\text{s.t.} \quad AD(x; \lambda_0 + r \nu) \leq b \\
r \geq r_0,
\]

where \( r_0 \) is appropriately defined to guarantee that \( \lambda_r \geq 0 \). Given \( \lambda_0 \) and \( \nu \), the problem as been reduced to the optimization over a line and can be accomplished using binary search.

## 2.5 Conclusions

In this chapter we have defined a class of problems called weakly coupled dynamic optimization problems. This class of problems not only has attractive structural properties, but it is also a good model of a broad class of applications. The results of this chapter are: (i) We have shown how Lagrangian relaxation of the constraints on controls yields a finite number of much smaller problems for finite horizon and infinite horizon problems; (ii) We have introduced methods for quickly determining an upper bound on the true cost to go for finite horizon problems, infinite horizon problems; (iii) We have introduced three Lagrange-based heuristics for determining a feasible control; (iv) We have motivated two values of Lagrange multipliers to use in the Lagrange-based heuristics. In the next chapter we apply the methods developed to several applications.
Chapter 3

Applications

In this chapter we consider applications of the framework we introduced in Chapter 2. The applications we consider are multiarmed bandits, restless bandits, multiclass queueing networks, and supply chain inventory problems. In Section 3.1, we illustrate that the proper selection of Lagrange multipliers results in the Gittins index procedure for multiarmed bandits. We then consider the more general problem of restless problems in Section 3.2 and develop a Lagrange based method that performs very well in computational experimentation. We relate our methods to those in the literature and provide some duality results linking our techniques to those in the literature. In Section 3.3, we investigate the performance of our methods for multiclass queueing networks and provide numerical results. In Section 3.4, we investigate supply chain inventory problems and demonstrate how capacitated problems can be decoupled into single unit-demand pairs. We finish with conclusions in Section 3.5.

3.1 Multiarmed bandits

In this section we apply our Lagrangian results to multiarmed bandits. We show that the Lagrangian policy introduced in Section 2.4.4 is optimal for multiarmed bandits. Furthermore, we show how to optimally choose which bandit to activate by using binary search over values of the Lagrange multiplier.
3.1.1 Introduction

The multiarmed bandit problem is a classic optimization problem of Operations Research. It has been used to model many applications, including job scheduling, sequential random sampling, clinical trials, investment in new products and random search, (see Varaiya et al. [46]). The solution of multiarmed bandits eluded researchers for many years until Gittins and Jones [21], in 1974, established the optimality of an indexing policy. Since that time the Gittins index policy, as the method is known, has been proved to be optimal by many researchers: Gittins [22], Whittle [51], Varaiya et al. [46], Weber [49], Tsitsiklis [43], and Bertsimas and Niño-Mora [10]. Of these, Weber [49], and Whittle, [51], expand on Gittins’ creation of a retirement option to help determine an optimal policy. Bertsimas and Niño-Mora [10] establish the connection between polymatroids, indexable systems and the decomposability of Gittins indices for multiarmed bandits.

3.1.2 Formulation

The discrete space multiarmed bandit problem can be formulated as a weakly coupled dynamic optimization problem with the following parameters.

State The state is \( x_t = (x_t^1, \ldots, x_t^N) \) where \( x_t^i \in X^i, |X^i| \) is finite.

Controls There are two types of controls per bandit \( i \). Setting \( u_t^i = 1 \) sets bandit \( i \) active, and \( u_t^i = 0 \) sets bandit \( i \) passive.

Randomness and Dynamics If \( u_t^i = 0 \) then \( x_{t+1}^i = x_t^i \). Otherwise, the state of bandit \( i \) evolves with known probability \( P(x_{t+1}^i = y|x_t^i, u_t^i = 1) \).

Reward Active bandits receive reward \( R^i(x_t^i, 1) > 0 \) and passive bandits receive zero reward, \( R^i(x_t^i, 0) = 0 \). For this section we will often write \( R^i(x_t^i) \) as shorthand for \( R^i(x_t^i, 1) \). The reward received at time \( t \) is

\[
R(x_t, u_t) = \sum_{i=1}^{N} R^i(x_t^i, u_t^i),
\]
Constraints In every time period the controller must select exactly one bandit to set active and keep the remaining bandits passive. Thus the constraint is $e^t u_t = 1$.

Objective We consider the infinite horizon discounted multiarmed bandit problem. The objective is to solve

$$
\max_{\pi} E \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \beta^t R^i(x^i_t, \pi_t) \right],
$$

for some feasible policy $\pi$.

3.1.3 Optimality of the Lagrange-based policy

The goal of this section is to show the optimality of the Lagrange-based policy introduced in Section 2.4.4 in the context of multiarmed bandits. The policy of Section 2.4.4 is to find the smallest $\lambda$ such that only one of the Lagrangian decoupled problems has $D(x^i; \lambda) = 1$. The policy can be stated as follows:

Choose $\lambda$ such that at state $t = 0$ it is optimal to set at most one bandit active in the decoupled problems.

In particular, we decouple problems with Lagrange multipliers $\lambda$, and then search over values of $\lambda$ until in only one of the subproblems it is optimal to set the bandit active. We show that this policy is optimal because it coincides with Gittins index policy. Moreover, we show that it is a quick solution the problem because we do not have to calculate Gittins index. Rather, we use binary search over $\lambda$ until a feasible decision is found.

The key result of this section is Theorem 5. Before we state it we need to define $D^i(x^i; \lambda)$ and Gittins index.

Having defined the multiarmed bandit problem in the previous section, we proceed by applying the results of Section 2.2. In particular, Eq. (2.3) becomes

$$
L(x; \lambda) = \sum_{i=1}^{N} L^i(x^i; \lambda) + \frac{\lambda}{1 - \beta},
$$
where

\[
L_i^i(x^i; \lambda) = \max_{u^i \in \{0,1\}} \left\{ R_i^i(x^i, u^i) - \lambda u^i + \beta \mathbb{E}[L_i^i(y^i_i; \lambda)|x^i, u^i] \right\}.
\] (3.1)

Note that in contrast to the constraints on the controls of Chapter 2, we have equality, and not an inequality. The only difference this makes is that we now allow \( \lambda \) to take negative values.

Recall the definition of \( D_i^i(x^i; \lambda) \) in Eq. (2.18). For the case of multiarmed bandits it takes the form

\[
D_i^i(x^i; \lambda) = \arg \max_{u^i \in \{0,1\}} L_i^i(x^i; \lambda).
\]

In particular, we have

\[
R_i^i(x^i, 1) - \lambda + \beta \mathbb{E}[L_i^i(y^i_i; \lambda)|x^i, u^i = 1]
\geq R_i^i(x^i, 0) + \beta \mathbb{E}[L_i^i(y^i_i; \lambda)|x^i, u^i = 0] \Rightarrow D_i^i(x^i; \lambda) = 1,
\]

and \( D_i^i(x^i; \lambda) = 0 \) otherwise.

At this point we define Gittins index. Let \( \tau^i \) be a stopping rule. That is, \( \tau^i \) is a random variable that takes values in \( \{0, 1, 2, \ldots, \infty\} \).

\[
\nu^i(x_0^i) \equiv \sup_{\tau^i} \frac{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t R_i^i(x_t^i) \right]}{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t \right]}.
\] (3.2)

The quantity \( \nu^i(x^i) \) is known as Gittins index. The Gittins index policy is to set active the bandit with the greatest Gittins index.

The key result of this section is the following theorem.

**Theorem 5** For a given state \( x \), let \( \nu(1) = \max_i \nu^i(x_0^i) \), the first order statistic, and define \( \nu(2) \) as the second order statistic. Suppose \( \nu(2) < \nu(1) \). Then we have
(a) \[
\sum_{i=1}^{N} D_i^i(x_i^0; \lambda) = \begin{cases} 
0 & \text{for } \lambda > \nu_{(1)}, \\
1 & \text{for } \nu_{(2)} < \lambda \leq \nu_{(1)}, \\
\geq 2 & \text{for } \lambda \leq \nu_{(2)}. 
\end{cases}
\]

(b) For \( \nu_{(2)} < \lambda \leq \nu_{(1)} \), it is optimal to set \( u^i = D^i(x_i^0; \lambda) \) for \( i = 1, \ldots, N \).

The significance of Part (a) is that it corresponds to solving Problem (2.19). More importantly, with Lagrange decomposition of the problem allows for the subproblems \( L^i(\cdot) \) to be solved independently. Part (b) states that over a given range of \( \lambda \), it is optimal to set \( u = D(\lambda) \). The proof of Part (b) relies on the fact that it corresponds to Gittins index policy. The significance of this theorem is that it implies that varying \( \lambda \) until \( \sum_{i \in N} D_i^i(x_i^0; \lambda) = 1 \), and setting \( u^i = D^i(x_i^0; \lambda) \) is optimal. Moreover, for any \( \lambda \), \( D_i^i(x_i^0; \lambda) \) can be calculated in isolation. This is the basis for Algorithm 1 which appears later.

Theorem 5 requires \( \nu_{(1)} > \nu_{(2)} \). The consequence of having \( \nu_{(1)} \equiv \nu_{(2)} \) is that for \( \lambda = \nu_{(1)} \), \( D^i(\lambda) = 1 \) for two bandits (and more if other Gittins indices also equal \( \nu_{(1)} \)). (If \( \nu_{(1)} = \nu_{(2)} \), the optimal decision is found by randomly picking one of the two bandits.) The drawback the scenario with \( \nu_{(1)} = \nu_{(2)} \) is that binary search component of Algorithm 1 will never completely converge.

Our approach in proving Theorem 5 will be to relate \( L^i(x^i; \lambda) \) to a stopping rule dependent on \( \lambda \). We will show that for \( \lambda > \nu^i(x^i) \) it is optimal to stop immediately, i.e., set \( D^i(x^i; \lambda) = 0 \); otherwise, \( D^i(x^i; \lambda) = 1 \).

Before proving Theorem 5, we first prove the following lemmas.

**Lemma 1** \( L^i(x^i; \lambda) \) satisfies the following recursion for all \( x^i, i = 1, \ldots, N \):

\[
L^i(x^i; \lambda) = \max \bigl\{ 0, R^i(x^i) - \lambda + \beta \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i = 1] \bigr\}. 
\] (3.3)
**Proof.** First note that if \( u^i = 0 \), then \( R^i(x^i, u^i) - \lambda u^i = 0 \) and \( \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i] = L^i(x^i; \lambda) \). It follows that subproblem (3.1) can be rewritten as

\[
L^i(x^i; \lambda) = \max \left\{ \beta L^i(x^i; \lambda), R^i(x^i) - \lambda u^i + \beta \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i = 1] \right\}. \tag{3.4}
\]

Define

\[
\hat{L}(x^i; \lambda) \equiv \max \left\{ 0, R^i(x^i) - \lambda + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] \right\}. \tag{3.5}
\]

Our goal is to show that \( \hat{L} \) satisfies Eq. (3.4).

Suppose \( R^i(x^i) - \lambda u^i + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] \geq 0 \). Then

\[
\hat{L}(x^i; \lambda) = R^i(x^i; \lambda) - \lambda + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] \geq 0
\]

\[
\Rightarrow \quad \hat{L}(x^i; \lambda) \geq \beta \hat{L}(x^i; \lambda)
\]

\[
\Rightarrow \quad R^i(x^i; \lambda) - \lambda + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] \geq \beta \hat{L}(x^i; \lambda).
\]

Suppose \( R^i(x^i) - \lambda u^i + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] < 0 \). Then

\[
\hat{L}(x^i; \lambda) = 0 \quad \Rightarrow \quad \hat{L}(x^i; \lambda) = \beta \hat{L}(x^i; \lambda)
\]

\[
\Rightarrow \quad \beta \hat{L}(x^i; \lambda) > R^i(x^i) - \lambda u^i + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1].
\]

It follows that

\[
\hat{L}^i(x^i; \lambda) = \max \left\{ \beta \hat{L}^i(x^i; \lambda), R^i(x^i) - \lambda u^i + \beta \mathbb{E}[\hat{L}^i(y^i; \lambda)|x^i, u^i = 1] \right\}.
\]

We have now transformed \( L^i(\cdot) \) into a form more convenient for stopping problems. This will enable us to characterize the value of \( L^i(x^i_0; \lambda) \) and \( D^i(x^i_0; \lambda) \) for \( \lambda \leq \nu^i(x^i_0) \) and \( \lambda > \nu^i(x^i_0) \).

**Lemma 2** Let \( \tau \) be a stopping rule that takes on values in \( \{0, 1, 2, \ldots, \infty\} \). Then the
following equality holds:

\[ L^i(x_0^i; \lambda) = \sup_{\tau \geq 0} \mathbb{E} \left[ \sum_{t=0}^{\tau-1} \beta^t (R^i_t(x^i_t) - \lambda) \right]. \] (3.6)

For \( \tau = 0 \), we define \( L^i(x_0^i; \lambda) = 0 \).

The proof is due to Tsitsiklis and van Roy [44], for more general stopping problems. We denote the argsup of Eq. (3.6) by \( \tau^i(x_0^i; \lambda) \).

In the next lemma, we show that for the decoupled problems, for \( \lambda \) greater than a bandits Gittins index, the optimal decision is to set \( D^i(x^i; \lambda) = 0 \), and \( D^i(x^i; \lambda) = 1 \), otherwise.

**Lemma 3** We have that

(a) \( \lambda > \nu^i(x_0^i) \Rightarrow D^i(x_0^i; \lambda) = 0 \),

(b) \( \lambda \leq \nu^i(x_0^i) \Rightarrow D^i(x_0^i; \lambda) = 1 \).

**Proof.** First note that \( \tau^i(x^i; \lambda) = 0 \) corresponds to \( D^i(x^i; \lambda) = 0 \). Also, \( \tau^i(x^i; \lambda) \geq 1 \) corresponds to \( D^i(x^i; \lambda) = 1 \). We now proceed with the proof. For Part (a), note that

\[ \lambda > \nu^i(x_0^i) \Rightarrow \lambda \geq \sup_{\tau^i \geq 1} \frac{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t R^i_t(x^i_t) \right]}{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t \right]} \]

\[ \Rightarrow \sup_{\tau^i \geq 1} \frac{\sum_{t=0}^{\tau^i-1} \beta^t} {\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t \right]} < 0 \] (3.7)

\[ \Rightarrow \sup_{\tau^i \geq 0} \frac{\sum_{t=0}^{\tau^i-1} \beta^t} {\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t \right]} = 0. \] (3.8)

Compare Eq. (3.8) to Eq. (3.6). Eq. (3.8) says that \( L^i(x^i; \lambda) = 0 \). Moreover, Eq. (3.7) says that if \( \tau^i \geq 1 \), then \( R^i_t(x^i_t; \lambda) - \lambda + \mathbb{E}[L^i(x^i; \lambda)] < 0 \). In words, \( u^i = 1 \) yields strictly negative expected reward, whereas \( u^i = 0 \) guarantees reward of zero. Thus, the optimal decision is \( D^i(x^i; \lambda) = 0 \).
For Part (b), we have

\[
\lambda \leq \nu^i(x^i_0) \implies \lambda < \sup_{\tau^i \geq 1} \frac{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t R^i(x^i_t) \right]}{\mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t \right]} \\
\implies \sup_{\tau^i \geq 1} \mathbb{E} \left[ \sum_{t=0}^{\tau^i-1} \beta^t (R^i(x^i_t) - \lambda) \right] \geq 0.
\]

The last statement says that we can guarantee a total discounted reward of at least zero by activating bandit \textit{i} at least once. It follows that \( D^i(x^i; \lambda) = 1. \)

\[ \text{Box} \]

**Proof of Theorem 5**

For a given state \( x \), let \( \nu_{(1)} = \max_i \nu^i(x^i) \), the first order statistic, and define \( \nu_{(2)} \) as the second order statistic. We first consider Part (a). Suppose \( \lambda > \nu_{(1)} \). Then from Lemma 3, we have that \( D^i(x^i; \lambda) = 0 \) for \( i = 1, \ldots, N \). Suppose \( \nu_{(2)} < \lambda \leq \nu_{(1)} \).

Then from Lemma 3, the only bandit with \( D^i(x^i; \lambda) = 1 \) is the bandit satisfying \( j = \arg \max_{i \in \mathcal{N}} \nu^i(x^i) \). For \( \lambda \leq \nu_{(2)} \), it follows that there are at least two bandits such that \( D^i(x^i; \lambda) = 1 \). This establishes Part (a).

For Part (b) note that for \( \nu_{(2)} < \lambda \leq \nu_{(1)} \), setting \( u^i = D^i(x^i; \lambda) \) for \( i = 1, \ldots, N \) corresponds to setting the bandit with the greatest Gittins index active, and setting all other bandits passive. This is the Gittins index policy, which is optimal. \[ \text{Box} \]

Algorithm 1 is a Lagrange-based policy motivated by the results of Theorem 5. In effect it uses the policy of Section 2.4.4, which we have proved to be optimal for multiarmed bandits. Algorithm 1 begins with \( \lambda_L = 0 \) and \( \lambda_U = \max_{x^i, \beta \in \mathcal{N}} R^i(x^i) \). Note that \( \nu^i(x^i_0) \leq \max_{x^i, \beta \in \mathcal{N}} R^i(x^i) \) for all \( j \).

There are two important properties of Algorithm 1. The first is that it uses the information from subproblems in order to determine a feasible solution. Second, it uses binary search in order to find a Lagrange multiplier that guarantees optimality of our decision \( u \). Of course, other means of finding a sufficient value of \( \lambda \) can be
used. In particular, in Section 3.1.4 we use linear programming to find a sufficient Lagrange multiplier.

Algorithm 1 Multiarmed bandit Lagrangian heuristic

**Step 0**: Set $\lambda_U = \max_{i \in S} R_i(x^i)$, $\lambda_L = 0$

**Step 1**: Set $\lambda_M = \frac{\lambda_U + \lambda_L}{2}$. Set $p = \sum_{i \in N} D_i(x^i_0; \lambda_M)$;

if $p = 1$ then
  For $i = 1, \ldots, N$ set $u^i = D_i(x^i; \lambda)$, terminate
else if $p = 0$ then
  Set $\lambda_U = \lambda_M$, go to Step 1
else if $p \geq 2$ then
  Set $\lambda_L = \lambda_M$, go to Step 1
end if

3.1.4 Implementing Algorithm 1 using linear programming

In this section we use linear programming to find the optimal bandit to play. The purpose of Algorithm 1 is to quickly find what bandit to set active. A sufficient condition for finding $\nu_2 < \lambda \leq \nu_1$, is to set $\lambda = \nu_1$. Let $k = \arg \max_i \nu^i(x^i_0)$. Then assuming $\nu_2 < \nu_1$, for $\lambda = \nu_1$,

$$R_i(x^i) - \lambda + \mathbb{E}[L_i(y^i; \lambda)|x^i, u^i = 1] < 0 \quad \text{for } i \neq k,$$

$$R_i(x^i) - \lambda + \mathbb{E}[L_i(y^i; \lambda)|x^i, u^i = 1] = 0 \quad \text{for } i = k.$$

In particular, $\nu_1$ is the smallest $\lambda$ such that $R^k(x^k) - \lambda + \mathbb{E}[L^k(y^k; \lambda)|x^k, u^k = 1] = 0$. Furthermore, for $\lambda < \nu_1$, $R_k(x^k) - \lambda + \mathbb{E}[L^k(y^k; \lambda)|x^k, u^k = 1] > 0$. With this in mind, we write a linear program based on Problem (2.5) of Section 2.2.1. The goal is to find the smallest $\lambda$ such that $\alpha_{x^i} L_{x^i} \leq 0$. Recall that $\alpha_{x^i} = 1$ if $x^i = x^i_0$ and zero otherwise (see Section 2.2.1). Combining this result with our formulation of
decoupled dynamic programs in Section 2.2.1, we have

\[
\text{(P) } Z = \min_{L^i_{x}} \sum_{x \in X^i} \sum_{y \in X^i} \alpha_{x^i} L^i_{x^i} + \lambda \frac{1}{1 - \beta} \tag{3.9}
\]

s.t. \( L^i_{x^i} \geq R^i_{x^i} - \lambda + \beta \sum_{y \in X^i} p^{i}_{y^i x^i} L^i_{y^i} \) \tag{3.10}

\( L^i_{x^i} \geq 0 \) \tag{3.11}

\( \alpha_{x^i} L^i_{x^i} \leq 0, \forall i \in \mathcal{N}, x^i \in X^i. \) \tag{3.12}

Eqs. (3.10) and (3.11) are as before. Eq. (3.12) ensures that the penalty \( \lambda \) is large enough so that no project is worth playing at that rate starting from states with \( \alpha_{x^i} = 1 \). Upon termination, \((1 - \beta)\)Z equals the Gittins index for state \( y^i \). Let \( z = \lambda/(1 - \beta) \) and change the cost vector in the objective function (3.9) from \( (\alpha, 1) \) to \((1, \ldots, 1, N)\). Both cost vectors are positive, so the final solutions are identical (see Puterman [39]). The new linear program is the formulation for finding the Gittins index proposed by Chen and Katehakis [17].

Consider the dual of (P) written below.

\[
\text{(D) } \max \sum_{x \in X^i} \sum_{y \in X^i} R^i_{x^i} z^i_{x^i}
\]

s.t. \( z^i_{x^i} \leq \alpha_{x^i}(1 + w_{x^i}) + \beta \sum_{y \in X^i} p^{i}_{y^i x^i} z^i_{y^i} \)

\[
\sum_{x \in X^i} \sum_{y \in X^i} z^i_{x^i} = \frac{1}{1 - \beta}
\]

\( z^i_{x^i}, w_{x^i} \geq 0, \forall i \in \mathcal{N}, x^i \in X^i. \)

It can be seen that the linear program asks us to find to find one bandit with the greatest reward rate. (D) says to play that particular bandit. This is a greedy procedure and was proven optimal by Bertsimas and Niño-Mora [10].
3.1.5 The multiarmed bandit problem with $K > 1$

Consider again the multiarmed bandit problem with $K > 1$ pulls per period. The problem of solving the cost to go at state $x$ can be formulated as

$$J(x) = \max_u \sum_{i=1}^n R_i(x^i, u^i) + \beta \mathbb{E}[J(y)|x, u]$$

s.t. $\sum_{i=1}^N u^i = K$

$u^i \in \{0, 1\}$.

We will again use the constant Lagrangian multiplier $\lambda$ for all states so the bandits decouple and we are left to solve

$$L(x; \lambda) = \sum_{i=1}^N L_i(x^i; \lambda) + K \frac{\lambda}{1 - \beta}$$

where $L_i(x^i; \lambda)$ has the same definition as in Eq. (3.3). The following heuristic for the original problem follows naturally:

At every time stage find $\lambda$ such that $D_i(x^i; \lambda) = 1$ for $K$ bandits and set

$u = D(\lambda)$.

This is equivalent to pulling the bandits of the $K$ highest Gittins indices and will be the basis for the heuristic we propose for restless bandits.

3.2 Restless bandits

In this section we apply our results of Lagrangian relaxation to restless bandits. The results of this section are: (i) A Lagrange-based policy for restless bandits that outperforms existing heuristics; (ii) A dynamic programming interpretation of exact and approximate linear programming formulations of restless bandits; (iii) A new interpretation of a primal-dual heuristic for restless bandits; (iv) Higher-order coupling methods that produce better bounds on the cost to go function; (v) Numerical results
that show the effectiveness of our Lagrange-based policy.

### 3.2.1 Introduction

The restless bandit problem is an extension of the multiarmed bandit problem in which both active and passive rewards are received, and state transition occurs for both passive and active bandits. In addition, all rewards are positive. The problem was introduced by Whittle [53] as a model for a broad class of applications, which include: medical treatment in which the $N$ patients' states change if left untreated, and possibly improve if they were among the $K$ treated; the tracking of $N$ enemy submarines by $K$ aircraft; job scheduling whereby workers output decreases if not allowed to rest. Subsequently, Papadimitriou and Tsitsiklis [37] showed this problem to be PSPACE-hard. A number of heuristics have been developed to solve the problem. Whittle [53], developed an indexing method for the time-average restless bandit problem. Weber and Weiss [50] showed that the heuristic asymptotically approaches the optimal policy under certain conditions. Bertsimas and Niño-Mora [11] provided a first order linear programming relaxation of the problem and devise a heuristic for choosing the optimal policy.

As shown in Section 2.2, weakly coupled $N$-dimensional dynamic optimization problems can be decoupled into $N$ one-dimensional problems using Lagrangian relaxation. In Section 3.1 we showed that the heuristic of choosing the Lagrange multiplier such that we are willing to play a single bandit in the decoupled problem is optimal for the multiarmed bandit problem. This same heuristic will be the basis for the algorithm proposed for solving the restless bandit problem.

### 3.2.2 Structure

In this section we examine the restless bandit problem, which is formulated in Section 3.2.3. In Section 3.2.4, we introduce a Lagrange-based policy based on the policy of Section 2.4.4. In Section 3.2.5, we investigate the primal-dual heuristic of Bertsimas and Niño-Mora [11]. We give a new interpretation of their policy and relate it to our
policy introduced in Section 2.4.1. In Section 3.2.7, we show how restless bandits can be decomposed into groups of bandits and each group can be solved in isolation. For the case of groups of size two, we show the equivalence of this approach to the pairwise linear program established by Bertsimas and Niño-Mora [11]. In Section 3.2.8, we investigate the performance of different policies using computational experiments.

3.2.3 Formulation

We formulate the restless bandit problem as a weakly coupled dynamic optimization problem in the following way.

State The state is \( x_t = (x_1^t, \ldots, x_N^t) \) where \( x_i^t \in X_i \), \( |X_i| \) is finite.

Controls The controls are \( u_t = (u_1^t, \ldots, u_N^t) \) where \( u_i^t \in \{0, 1\} \). As with multiarmed bandits, bandits with \( u_i^t = 1 \) are active at time \( t \), and otherwise they are passive. This is a slight misnomer since passive bandits still go through transition and also accrue reward.

Randomness and Dynamics The transition probabilities \( P(x_{t+1}^i = y|x_t^i, u_t^i) \) are known. In contrast to multiarmed bandits, passive bandits in this problem may change state.

Reward The reward received at time \( t \) is

\[
R(x_t, u_t) = \sum_{i=1}^{N} R^i(x_t^i, u_t^i),
\]

with \( R^i(x_t^i, 1) \geq 0 \) for all states and controls.

Constraints In every time period the controller must select exactly \( K \) bandits to set active and keep the remaining bandits passive. Thus the budget constraint is \( e' u_t = K \).

Objective We consider the infinite horizon discounted restless bandit problem. The
objective is to solve
\[
\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{\infty} \sum_{i=1}^{N} \beta^t R^i(x^i_t, \pi^i_t) \right],
\]
for some feasible policy \( \pi \).

### 3.2.4 A Lagrangian policy for restless bandits

In this section we develop a Lagrange-based policy for restless bandits called INDEX. The algorithm is a special case of the algorithm developed in Section 2.4.1 applied to restless bandits. Motivated by the optimality of Algorithm 1 for multiarmed bandits, we develop INDEX along similar lines. Before we present INDEX, we first define \( L(x^i; \lambda), F(x^i; \lambda) \). We begin by defining the Bellman equation for restless bandits, and then present the Lagrangian-decoupled version.

\[
J(x) = \max \sum_{i=1}^{N} R^i(x^i, u^i) + \beta \mathbb{E}[J(y)|x, u] \\
\text{s.t.} \quad e'u = K \\
\quad u \in \{0, 1\}^N.
\]

It follows that
\[
L(x; \lambda) = \sum_{i=1}^{N} L^i(x^i; \lambda) + \lambda \frac{K}{1 - \beta}, \tag{3.13}
\]

where
\[
L^i(x^i; \lambda) = \max_{u \in \{0, 1\}} \left\{ R^i(x^i, u^i) - \lambda u^i + \beta \mathbb{E}[L^i(y^i; \lambda)|x^i, u^i] \right\}. \tag{3.14}
\]

As in Section 3.1, we must allow \( \lambda \) to take negative values, since the constraints on the controls of the original problem require equality.
Let

\[ F^i_0(x^i, \lambda) = \mathbb{R}^4(x^i, 0) + \beta \mathbb{E}[L^i(x^i; \lambda)|u^i = 0], \]
\[ F^i_1(x^i, \lambda) = \mathbb{R}^4(x^i, 1) - \lambda + \beta \mathbb{E}[L^i(x^i; \lambda)|u^i = 1]. \]

**Lemma 4** \( F^i_u(x^i; \lambda) \) is convex, piecewise linear and non-increasing in \( \lambda \) for \( u = 0, 1 \).

**Proof.** This follows from Theorem 2, since \( L^i(x^i; \lambda) \) are convex and piecewise linear. That \( F^i_0(x^i; \lambda) \) are non-increasing in \( \lambda \) is immediate. 

In Algorithm 1, we compared \( \mathbb{R}^i(x^i_0; u^i) - \lambda + \mathbb{E}[L^i(y^i; \lambda)|x^i_0, u^i = 1] \) to zero to determine \( D^i(x^i_0; \lambda) \). For restless bandits we have

\[ F^i_1(x^i, \lambda) \geq F^i_0(x^i, \lambda) \Rightarrow D^i(x^i_0; \lambda) = 1, \]

and \( D^i(x^i_0; \lambda) = 0 \) otherwise.

The key idea of INDEX is to find a \( \lambda \) such that \( D^i(x^i_0; \lambda) = 1 \) for exactly \( K \) bandits. In other words, find \( \lambda \) such that \( F^i_1(x^i, \lambda) \geq F^i_0(x^i, \lambda) \) for exactly \( K \) bandits. In particular, we would like know that constants \( c^i(x^i_0) \) exist such that

\[ \lambda < c^i(x^i_0) \quad \Rightarrow \quad D^i(x^i_0; \lambda) = 1, \]
\[ \lambda \geq c^i(x^i_0) \quad \Rightarrow \quad D^i(x^i_0; \lambda) = 0. \]

That is, we would like \( D^i(x^i; \lambda) \) to be monotonically decreasing in \( \lambda \). Although \( F^i \) are convex and linear, for general problems there is no guarantee that such thresholds exist. Appendix A provides an example for which there are no such thresholds. However in some cases, even for general Markov chains, monotonicity is guaranteed.

**Theorem 6** \( D^i(\lambda) \) is non-increasing in \( \lambda \) for \( \beta < \frac{1}{2} \).

**Proof.** Note that where defined, \( \frac{\partial F^i_1(x^i; \lambda)}{\partial \lambda} \leq -1 \) whereas

\[ \frac{\partial F^i_0(x^i, \lambda)}{\partial \lambda} \geq -\sum_{t=1}^{\infty} \beta^t = \frac{-\beta}{1 - \beta} > -1. \]
Application of Lemma 4 yields the result.

We are now ready to present our INDEX, shown in Algorithm 2. INDEX uses binary search to find a $\lambda$ such that $\sum_{i=1}^{N} D^i(x^i_0; \lambda) = K$. This is the same approach as in Algorithm 1. One difference is that even if we are not concerned with tie breaks, we are still not guaranteed convergence due to the lack of monotonicity of $D^i(x^i; \lambda)$. Thus, if $\lambda$ converges to a small enough region and $\sum_{i=1}^{N} D^i(x^i_0; \lambda) \neq K$ then we need a means of determining $u$. The function used is RANK($\mathbf{F}$), shown in Algorithm 3. The argument of RANK(·) is $\mathbf{F} = (F^i_{w^i}(x^i; \lambda))_{w^i \in \{0, 1\}, i \in \mathcal{N}}$, and its output is a feasible decision $u$. RANK($\mathbf{F}$) selects the bandits with the $K$ highest values of $F^i_1(x^i_0; \lambda) - F^i_0(x^i_0; \lambda)$ for a given $\lambda$.

Note that INDEX starts with $\lambda_L = -\Lambda$ and $\lambda_U = \Lambda$. The reason for this is that we are guaranteed that $D^i(x^i_0; \lambda) = 0$ for $\lambda > \Lambda$. Likewise, $\lambda < -\Lambda$ guarantees $D^i(x^i_0; \lambda) = 1$.

3.2.5 An integer optimization Lagrange-based policy and the primal-dual heuristic

In this section we present an integer optimization policy based on information from the decoupled Lagrangian functions. This method was first introduced in broad terms in Section 2.4.1, and in Problem (2.17) in particular. We show the equivalence between Problem (2.17) and the primal-dual heuristic for restless bandits proposed by Bertsimas and Niño-Mora [11]. Furthermore, we show the relationship between our decoupling heuristic and the first order linear program of the performance region proposed by Bertsimas and Niño-Mora [11].
Algorithm 2 INDEX heuristic

INPUT: $S = \emptyset, \epsilon, \lambda_L = -\Lambda, \lambda_U = \Lambda, \Lambda = \max_{x^i, u^i, j} R^i(x^i, u^i)$

while $|S| \neq K$ do
  $\lambda_M = \frac{\lambda_L + \lambda_U}{2}$
  for $1 = 1 \ldots$ do
    if $F_1^i(x^i; \lambda) \geq F_0^i(x^i; \lambda)$ then
      $S \leftarrow S \cup \{i\}$
    end if
  end for
  if $|S| < K$ then
    $\lambda_U = \lambda_M$
  else if $|S| > K$ then
    $\lambda_L = \lambda_M$
  end if
  if $\lambda_U - \lambda_L < \epsilon$ then
    $\lambda_M = \frac{\lambda_L + \lambda_U}{2}$, $F = (F_{w^i}(x^i; \lambda))_{u^i \in (0,1), j \in \mathcal{N}}, S = \text{RANK}(F)$
  end if
end while

for $i = 1 \ldots N$ do
  if $i \in S$ then
    $u^i = 1$
  else
    $u^i = 0$
  end if
end for

Algorithm 3 RANK subroutine

INPUT: $F^i_w(x^i_0; \lambda_M), i = 1, \ldots, N, S = \emptyset$
while $|S| < K$ do
  loop
    select $i \in \arg\max\{F_1^i(x^i; \lambda_M) - F_0^i(x^i; \lambda_M)\} | i \in \mathcal{N} \setminus S$
    $S \leftarrow S \cup \{i\}$
  end loop
end while

RETURN $S$

63
Linear programming formulation of the Lagrangian-decoupled problem

We begin with the linear programming formulation of the Lagrangian-decoupled problem.

\[
L(x) = \min \sum_{i \in N} \sum_{x^i \in X^i} \alpha_{x^i} L^i_{x^i} + \lambda \frac{K}{1 - \beta} \\
\text{s.t. } L^i_{x^i} \geq P^1_{x^i} - \lambda \sum_{y^i \in X^i} \beta_{y^i} L^i_{y^i} \tag{3.15} \\
L^i_{x^i} \geq P^0_{x^i} + \beta \sum_{y^i \in X^i} \beta_{y^i} L^i_{y^i}.
\]

For a given \( \lambda \), the integer optimization approach from Section 2.4.1 for restless bandits has the form

\[
\max \sum_{i=1}^{N} g^i(u^i) z^i_{u^i}, \\
\text{s.t. } \sum_{i=1}^{N} \sum_{u^i \in \{0,1\}} u^i z^i_{u^i} = K, \tag{3.16} \\
\sum_{u^i \in \{0,1\}} z^i_{u^i} = 1, \\
z^i_{u^i} \in \{0,1\},
\]

where

\[
g^i(u^i) = R^i(x^i, u^i) - \lambda + \beta E[L^i(x^i, \lambda) | x^i, u^i].
\]

In Section 2.4.3 we considered two possible values for \( \lambda \). In this section we will consider the case where we set \( \lambda^* = \arg \min L(x; \lambda) \), where \( \lambda^* \) solves Problem (3.15). This leads us to the policy proposed by Bertsimas and Niño-Mora [11].

The primal-dual heuristic

In this section we introduce a dynamic programming interpretation of the primal-dual heuristic of Bertsimas and Niño-Mora [11] (see Algorithm 4) and show its equivalence
to our integer programming-based algorithm of Section 2.4.1. The primal-dual heuristic involves first solving a linear program. Then, based on the reduced costs of the solution, it finds a feasible decision. Before we write the linear program formulated by Bertsimas and Niño-Mora [11] we need a few definitions. Define

$$ z_{x^i}^u(\pi) = E_\pi \left[ \sum_{t=0}^{\infty} I_{x^i}^u(t) \beta^t \right], $$

where $\pi \in \Pi$, and $I_{x^i}^u(t) = 1$ if bandit $i$ is in state $x^i$ and uses control $u^i$ at time $t$, and $I_{x^i}^u(t) = 0$ otherwise. The linear program follows.

$$ (P^1) \quad Z^1 = \max \sum_{i \in \mathcal{N}} \sum_{x^i \in X^i} \sum_{u^i \in \{0,1\}} z_{x^i}^{u^i} R_{x^i}^{u^i} $$

s.t. $$ \sum_{u^i \in \{0,1\}} z_{x^i}^{u^i} = \alpha_{x^i} + \beta \sum_{y^i \in \mathcal{X}^i} \sum_{u^i \in \{0,1\}} p_{y^i,x^i}^{u^i} z_{y^i}^{u^i} \quad (3.17) $$

$$ \sum_{i \in \mathcal{N}} \sum_{x^i \in X^i} z_{x^i}^1 = \frac{K}{1 - \beta} \quad (3.18) $$

Eq. (3.17) is the flow conservation constraint on individual bandits. The coupling constraint, Eq. (3.18), states that on average, $K$ bandits are being pulled per period. Let $\gamma_{x^i}^{u^i}$ be the optimal reduced costs for $(P^1)$. The primal-dual heuristic is shown in Algorithm 4.

**Algorithm 4 Primal-dual heuristic**

Step 0 For a given state $x$, solve (LP$^1$).

Step 1 Set $p = |\{z_{x^i}^1 : z_{x^i}^1 > 0, \alpha_{x^i} = 1, i \in \mathcal{N}\}|$

if $p \leq K$ then

set active the projects with positive active primal

end if

if $p < K$ then

set active $K - p$ projects with smallest active reduced costs $\gamma_{x^i}^1$

else if $p > K$ then

set active $K$ projects with largest passive reduced costs $\gamma_{x^i}^0$

end if

65
It can be seen that \((P^1)\) is the dual of Problem (3.15). Moreover, we have

\[
\begin{align*}
\gamma^0_{x_t^i} &= L_{x_t^i} - (R^0_{x_t^i} + \beta E[L_{y_t^i}, x_t^i, 0]). \\
\gamma^1_{x_t^i} &= L_{x_t^i} - (R^1_{x_t^i} - \lambda^* + \beta E[L_{y_t^i}, x_t^i, 1])
\end{align*}
\]  
(3.19)\hspace{1cm} \text{(3.20)}

It follows that

\[
\gamma^1_{x_t^i} - \gamma^0_{x_t^i} = F^i_0(\lambda^*) - F^i_1(\lambda^*).
\]

**Remark 1** At most one of \(\gamma^0_{x_t^i}, \gamma^1_{x_t^i}\) is positive.

Thus, ranking the smallest \(\gamma^1_{x_t^i}\) and the largest \(\gamma^0_{x_t^i}\), as the primal-dual heuristic does, is equivalent to ranking \(\gamma^0_{x_t^i} - \gamma^1_{x_t^i}\) from largest to smallest and choosing the \(K\) greatest of these.

There are three things to note:

1. \(L(x) = Z^1\). In particular, \((P^1)\) and Problem (3.15) are dual linear programs of each other.

2. The primal-dual heuristic solves a special case of our minimizing lambda integer optimization policy (2.17) of Section 2.4.4 (rewritten for general \(\lambda\) for restless bandits in Problem (3.16). To see this we first note that \(\gamma^1_{x_t^i}\) and \(\gamma^0_{x_t^i}\) of Eqs. (3.20) and (3.19), respectively, are equivalent to Eq. (2.16). That is, they are the slack variables found in solving \(L(x)\). We then write Problem (2.17) in terms of \(\gamma\).

\[
\begin{align*}
\min & \quad \sum_{i=1}^{N} \gamma^0_{x_t^i} z_0^i + \gamma^1_{x_t^i} z_1^i \\
\text{s.t.} & \quad \sum_{i=1}^{N} z_1^i = K \\
& \quad z_0^i + z_1^i = 1 \\
& \quad z_0^i, z_1^i \in \{0, 1\}.
\end{align*}
\]  
(3.21)

If \(z_1^i = 1\), then cost \(\gamma^1_{x_t^i}\) is added. Otherwise, \(z_0^i = 0\) and cost \(\gamma^0_{x_t^i}\) is added. It
follows that Problem (3.21) is equivalent to
\[
\min \sum_{i=1}^{N} (\gamma_{x^i}^1 - \gamma_{x^i}^0) w^i + \sum_{i=1}^{N} \gamma_{x^i}^0,
\]
\[
\text{s.t. } \sum_{i=1}^{N} w^i = K,
\]
\[
w^i \in \{0, 1\}. \tag{3.22}
\]

The solution to Problem (3.22) is the vector \( w \) that finds the \( K \) lowest values of \((\gamma_{x^i}^1 - \gamma_{x^i}^0)\). This is precisely the solution that primal dual heuristic in Algorithm 4 finds.

3. The primal-dual heuristic has the following interpretation: Activate the \( K \) bandits with greatest value of \( F_1^i(x^i; \lambda^*) - F_0^i(x^i; \lambda^*) \). In words, activate the \( K \) bandits that yield the greatest expected difference of reward between activity and passivity for \( \lambda = \lambda^* \).

### 3.2.6 An M-order coupling approach to restless bandits

In this section we will derive a method of coupling \( M \) restless bandits together with an appropriate set of Lagrange multiplier. Our aim is to find a means of decoupling the weakly constrained dynamic optimization problems into \( \binom{N}{M} \) groupings of \( M \) subprocesses solved together. The benefits of this is that minimizing over all Lagrange multipliers will yield a lower upper bound of the true cost to go. The central result of this section is the following theorem.

**Theorem 7** For a given \( M < N \), there exist Lagrange multipliers \( \lambda = (\lambda^1, \ldots, \lambda^N) \), \( \gamma = (\gamma^1, \ldots, \gamma^K) \) such that

(a)
\[
L_M(x; \lambda, \gamma) = \sum_{M \subseteq N} L^M_{x, M; \lambda, \gamma} + \lambda \gamma^{K/N} / (1 - \beta) + \sum_{T=0}^{K} \gamma^T \frac{(K)(N-K)/M}{(1 - \beta)},
\]

67
where

\[
L^M(x_M; \lambda, \gamma) = 
\max_{u \in \mathcal{K}} \left\{ \sum_{i \in \mathcal{M}} \left( \frac{R_i(x^i, u^i)}{N} - \lambda u^i \right) - \gamma e' u + \beta E[L^M(y^M; \lambda, \gamma)|x_M, u] \right\}.
\]

(b) \( \min_{\lambda, \gamma} L_M(x; \lambda, \gamma) \geq J(x) \).

(c) \( L_M(x; \lambda, \gamma) \) is convex and piecewise linear in \( \lambda, \gamma \).

This theorem says that restless bandits can be decoupled into problems of size \( M \), with all \( \binom{N}{M} \) problems solved in isolation.

**Proof.** The proof of Part (a) involves first formulating the restless bandit problem as a sum of \( \binom{N}{M} \) linked dynamic programs, each of size \( M \). Lagrangian relaxation of the problem yields the result. The proofs of Part (b) and (c) closely follow the proof of Theorem 2 and are omitted.

For Part (a), we begin by writing the exact formulation of the restless bandit problem:

\[
J(x) = \max_{u, i \in \mathcal{N}} \left\{ \sum_{i \in \mathcal{N}} R(x^i, u^i) + \beta E[J(y)|x, u]|e'u = K \right\}.
\]
We group bandits into sets $\mathcal{M}$ of size $M < K$ and denote by $u^i_M$ the control applied to bandit $i$ in grouping $\mathcal{M}$. 

$$J(x) = \max_{u_M^i, w_M^i, v^i} \sum_{\mathcal{M} \in \mathcal{N}} \sum_{i \in \mathcal{M}} \frac{R(x^i, u^i_M)}{N} + \beta E[J(y)| x, u]$$

s.t. $u^i_M = v^i, \; i \in \mathcal{M}, \; \forall \mathcal{M} \in \mathcal{N}$ \hfill (3.23)

$$e'v = K$$ \hfill (3.24)

$$\sum_{i \in \mathcal{M}} w^i_M = \sum_{T=0}^{K} Tw^T_M$$ \hfill (3.25)

$$\sum_{T=1}^{K} w^T_M \leq 1$$ \hfill (3.26)

$$\sum_{\mathcal{M} \in \mathcal{N}} w^T_M = \left(\begin{array}{c} K \\ T \end{array}\right) \left(\begin{array}{c} N - K \\ M - T \end{array}\right)$$ \hfill (3.27)

$w^T_M \in \{0, 1\}$

$v \in \{0, 1\}^N$.

The number of decision variables is $O(N + \binom{N}{M}(M + K))$. The decision variable $w^T_M = 1$ if the sum of decisions in set $\mathcal{M}$ is $T$, and is zero otherwise. Note that we require $T \leq K$. Eq. (3.23) ensures for all subsets $\mathcal{M}$ that contain bandit $i$ that the same decision for that bandit is made. Eq. (3.24) ensures exactly $K$ bandits are activated. Eqs. (3.25) and (3.26) state together that the sum of the decisions in set $\mathcal{M}$ is less than $K$. With $M < K$ these constraints are redundant. Eq. (3.27) is the key coupling constraint. It ensures that across all sets of size $M$ only the correct total number of decisions are made.

We replace Eq. (3.23) with $\sum_{\mathcal{M} \in \mathcal{N}} u^i_M = \binom{N}{M} v^i$ since $u^i_M \in \{0, 1\}$. Then Eq. (3.24) can be replaced by

$$\sum_{i \in \mathcal{N}, \mathcal{M} \in \mathcal{N}} u^i_M = K \binom{N}{M}.$$ \hfill (3.28)

We proceed by applying the same approach as used in Theorem 1. In particular, we attach Lagrange multiplier $\lambda$ to Constraint (3.28), and Lagrange multipliers $\gamma^T$
to Constraints (3.27). In addition, we will completely relax Constraints (3.23). This allows us to consider groupings of size $M$ in isolation.

Each subproblem $L^M(x; M, \gamma)$ has $O(|X|^M)$ states and $O(|U|^M)$ possible decisions per state. For a given set of Lagrange multipliers, the overall problem requires the solution of $\binom{N}{M}$ subproblems.

Let $L_1(x; \lambda)$ be the Lagrangian function defined in Eq. (3.13) and $L_1(x) = \min\lambda L_1(x; \lambda)$. Similarly, let $L_M(x; \lambda, \gamma) = \min\lambda, \gamma L_M(x; \lambda, \gamma)$. Bertsimas and Niño-Mora [11] consider the dual version of $\min\lambda, \gamma L_M(x; \lambda, \gamma)$, though they do not explicitly write it down.

The benefits of higher order grouping procedures is that correlations in the data can be accounted for. In particular, if two projects have close to zero correlation then they can be treated independently. However, if the magnitude of the correlation among a group of projects is great, then one should consider solving them together exactly (and assume the remaining projects behave independently).

**An $M$-order Lagrange-based policy**

In this section we present an integer optimization method for determining a feasible policy based on information from decoupled Lagrangian functions of Section 3.2.6. The integer optimization formulation closely resembles the one produced in Section 2.4.1. In particular, for given $\lambda, \gamma$, we define

$$g^M(u, M) = \sum_{i \in M} \left( \frac{R^i(x^i, u^i)}{N} - \lambda u^i \right) - \gamma c^u + \beta E[L^M(y^M; \lambda, \gamma)|x, u]$$
The integer optimization problem is to solve

$$\max_{\nu_{uM}^M} \sum_{M \in \mathcal{N}} \sum_{uM} \nu_{uM}^M g^M(u_M)$$

(3.29)

s.t. $$\sum_{i \in \mathcal{N}} \sum_{M \ni i} \sum_{uM} \nu_{uM}^M \frac{u_M}{N} = K$$

(3.30)

$$\sum_{uM} \nu_{uM}^M = 1$$

(3.31)

$$\nu_{uM}^M u_M e^i = \nu_{uK}^K w_K e^i \forall M, K \subset \mathcal{N}, M, K \ni i$$

(3.32)

$$\nu_{uM}^M \in \{0, 1\}, \forall \{u_M \in \{0, 1\}^M | e' u_M \leq K\},$$

where $e^i$ is an $M$ dimensional vector that equals 1 in the $i$th position, and is zero otherwise. Decision variable $\nu_{uM}^M = 1$ if in grouping $\mathcal{M}$, decision $u_M$ is made, and is zero otherwise. The objective (3.29) is to maximize the expected cost to go, as calculated by the different groupings. Constraint (3.30) ensures that the total decisions made sum to $K$. Constraint (3.31) ensures that only one grouped decision $u_M$ is made per grouping $\mathcal{M}$. Constraint (3.31) ensures that the same decision for a particular project $i$ is the same for all groupings that contain $i$.

The benefit of the above algorithm is that for problems that are tractable for $M \geq 2$, the above formulation pools more information into the decision making process. This algorithm is used in Chapter 5 and produces high quality results.

### 3.2.7 Linear programming results and pairwise couplings

In this section we present two linear programming results relating the formulations of Bertsimas and Niño-Mora [11] with dynamic programs. The first result is a theorem stating that the linear programming formulation of restless bandits presented by Bertsimas and Niño-Mora [11] is the dual problem of the linear programming formulation for the standard Bellman equation for restless bandits. In the next subsection we extend these results to pairwise coupled bandits. Note that the equivalence of the first order approximation (P1) and Problem (3.15) was established in Section 3.2.5.

We begin by defining the linear programming formulation for restless bandits
derived by Bertsimas and Niño-Mora [11]. Let \( \Pi \) be the set of stationary admissible policies. Let \( I^u_x(t) \) be 1 if the state is \( x \) and decision \( u \) is made at time \( t \), and 0 otherwise, under a stationary policy \( \pi \in \Pi \). Also, let \( R^u_x = \sum_{i=1}^{N} R^i(x^i, u^i) \) and \( p_{xy}^u = \prod_{i=1}^{N} P(x_{i+1}^i = y^i|x_i^i, u_i^i) \) The decision variable in the linear program is \( z^u_x \), the discounted time spent in state \( x \) making decision \( u \) (we are restricting ourselves to stationary policies). Formally \( z^u_x \) is defined as

\[
z^u_x(\pi) = \mathbb{E}_x \left[ \sum_{t=0}^{\infty} I^u_x(t) \beta^t \right].
\]

The linear program follows.

\[
(P) \quad Z^* = \max \sum_{x \in X} \sum_{u \in U_x} z^u_x R^u_x \\
\text{s.t. } \sum_{u \in U_x} z^u_x = \alpha_y + \beta \sum_{x \in X} \sum_{u \in U_x} p_{xy}^u z^u_x \quad (3.33)
\]

Eq. (3.33) is then the flow conservation constraint according to Bertsimas and Niño-Mora [11], and the objective is to maximize the expected discounted reward subject to using a stationary feasible policy.

**Theorem 8** For a given initial state \( x_0 \),

\[
Z^* = J(x_0).
\]

Furthermore, the dual of \( (P) \) is equivalent to the linear program formulation of restless bandits derived represented as a dynamic program.

**Proof.** The Bellman equation for restless bandits is

\[
J(x) = \max_{u \in \{0,1\}^N} \left\{ \sum_{i=1}^{N} R^i(x^i, u^i) + \beta \mathbb{E}[J(y)|x, u]|c'u = K \right\}.
\]
As in Section 2.2.1 we formulate the above problem as the following linear program.

\[
\text{(D) } J(x_0) = \min \sum_{x \in X} \alpha_x J_x \\
\text{s.t. } J_x \geq R^u_x + \beta \sum_{y \in X} p_{xy}^u J_y, \forall u \in U_f, \forall x.
\]

The dual result follows. By strong duality we have \( Z^* = J(x_0) \). ☐

**A second order LP relaxation**

In Section 3.2.7, we established a duality connection with the linear programming version of the dynamic programming formulation and the linear programming formulation of restless bandits proposed by Bertsimas and Niño-Mora [11]. In addition, we presented a connection between the linear programming relaxation of bandits of Bertsimas and Niño-Mora [11], and the Lagrangian-decoupled bound of the cost to go presented in Section 2.2.1. In this section we prove a duality connection between the second order linear programming relaxation of Bertsimas and Niño-Mora [11] and the higher-order coupling of bandits presented in Section 3.2.6 for the case \( M = 2 \). This result is stated in Theorem 9.


\[
z^{u^i, u^j}_{x^i, x^j}(\pi) = E_u \left[ \sum_{t=0}^{\infty} I_{x^i}^u(t) I_{x^j}^u(t) \beta^t \right].
\]

They define the second order feasible action space as

\[A^2 \equiv \{ (u^i, u^j) \in Z^2 | u^i + u^j \leq K \}.
\]

Let

\[Q^2_{ij} = \{ z_{i,j} = (z^{u^i, u^j}_{x^i, x^j}(\pi))_{x^i \in X^i, x^j \in X^j, (u^i, u^j) \in A^2 | \pi \in \Pi} \}.
\]

73
The authors propose that a full formulation of $Q^2_{i,j}$ is given by
\[
\sum_{(w',w) \in A^2} z_{w'w}^{i,j} = \alpha^i \alpha^j + \beta \sum_{x^i \in X^i, x^j \in X^j \atop (w',w) \in A^2} p_{x'y'}^{i,j} z_{w'w}^{x'y'}.
\]
\[(y^i, y^j) \in X^i \times X^j, z_{w'w}^{i,j} \geq 0, (x^i, x^j) \in X^i \times X^j, (w',w) \in A^2.\]


\[(P^2): \quad Z^2 = \max \sum_{1 \leq i < j \leq N} \sum_{x^i \in X^i} \sum_{x^j \in X^j} \sum_{(w',w) \in A^2} (\hat{R}_{x^i}^{w'} + \hat{R}_{x^j}^{w'}) z_{w'w}^{x^i x^j}
\]
\[\sum_{1 \leq i < j \leq N} \sum_{x^i \in X^i} \sum_{x^j \in X^j} z_{00}^{i,j} = \frac{(N-K)}{1-\beta}, \quad (3.34)\]
\[\sum_{1 \leq i < j \leq N} \sum_{x^i \in X^i} \sum_{x^j \in X^j} (z_{10}^{i,j} + z_{01}^{i,j}) = \frac{K(N-K)}{1-\beta}, \quad (3.36)\]
\[\sum_{1 \leq i < j \leq N} \sum_{x^i \in X^i} \sum_{x^j \in X^j} z_{11}^{i,j} = \frac{(K)}{1-\beta}, \quad (3.37)\]
\[\sum_{i=1}^{N} \left( \sum_{x^i \in X^i} \sum_{w^i \in A^2} \sum_{(u',u) \in A^2} z_{w'u}^{i,i} \right) + \sum_{i<j} \sum_{x^i \in X^i} \sum_{x^j \in X^j} \sum_{w^i \in A^2} \sum_{(u',u) \in A^2} z_{w'u}^{i,j} = \frac{K(N-1)}{1-\beta}, \quad (3.38)\]
\[z_{w'w}^{i,j} \geq 0, \quad x^i \in X^i, x^j \in X^j, 1 \leq i < j \leq N.\]

**Theorem 9** For a given state $x_0$, we have
\[Z^2 = \min_{\lambda, \gamma} L_2(x_0; \lambda, \gamma),\]

where $L_2(x; \lambda, \gamma)$ is defined in Theorem 7. Furthermore, the linear programming formulation for solving $L_2(x_0)$ is the dual problem of $P^2$.  

74
Proof. We begin by writing the Lagrangian functions from Theorem 7 for $M = 2$.

\[
L_2(x; \lambda, \gamma) = \sum_{(i,j) \in N} L^{(i,j)}(x_{(i,j)}; \lambda, \gamma) + \lambda \frac{K(N)}{1 - \beta} + \sum_{t=0}^{K} \gamma^T \left( \frac{K}{T} \frac{(N-K)}{(2-T)} \right),
\]

where

\[
L^{(i,j)}(x^{(i,j)}, \lambda, \gamma) = \max_{u \in u \leq K} \left\{ \frac{R_i(x^i, u^i)}{\binom{N}{2}} - \lambda u^i + \frac{R_j(x^j, u^j)}{\binom{N}{2}} - \lambda u^j - \gamma^{e^t} + \beta E[L^{(i,j)}(y^{(i,j)}; \lambda, \gamma)|x^{(i,j)}, u] \right\}.
\]

The goal in relating this pairwise solution is to show that the linear programming formulation for solving $L_2(x)$ is the dual of the second order linear programming formulation proposed by Bertsimas and Niño-Mora [11]. Let

\[
\tilde{R}_i^{u^i} = \frac{R_i(x^i, u^i)}{\binom{N}{2}},
\]

and define $\alpha_{x^i}$ and $p_{x^i}^{u^i}$ as in Section 2.2.1. Also, let

\[
\gamma^{e^t} = \begin{cases} 
\gamma^{00} & \text{for } e^t = 0, \\
\gamma^{10,01} & \text{for } e^t = 1, \\
\gamma^{11} & \text{for } e^t = 2.
\end{cases}
\]

It follows that the problem $\min_{\lambda, \gamma} L_2(x; \lambda, \gamma)$ is equivalent to the following linear
program.

\[(D^2) \quad L_2(x_0) = \min \quad \sum_{i \leq j \leq N} \sum_{x^j \in X^j} \sum_{z \in \mathcal{X}} \alpha_{x^j} z \cdot L_{x^j}^{(i,j)} + \frac{K(N - 1)}{1 - \beta} \gamma_{0,0,0} + \frac{K(N - K)}{1 - \beta} \gamma_{1,0,1} + \frac{(K)}{1 - \beta} \gamma_{1,1} \]

\[\text{s.t.} \quad L_{x^j}^{(i,j)} - \beta \sum_{y^i \in \mathcal{X}^i, y^j \in \mathcal{X}^j} P_{x^j y^i} P_{x^j y^j} L_{y^i y^j}^{(i,j)} + \gamma_{0,0,0} \geq \tilde{R}_{x^j}^0 + \tilde{R}_{x^j}^0 \]

\[L_{x^j}^{(i,j)} - \beta \sum_{y^i \in \mathcal{X}^i, y^j \in \mathcal{X}^j} P_{x^j y^i} P_{x^j y^j} L_{y^i y^j}^{(i,j)} + \gamma_{1,0,1} + \lambda \geq \tilde{R}_{x^j}^1 + \tilde{R}_{x^j}^1 \]

\[L_{x^j}^{(i,j)} - \beta \sum_{y^i \in \mathcal{X}^i, y^j \in \mathcal{X}^j} P_{x^j y^i} P_{x^j y^j} L_{y^i y^j}^{(i,j)} + \gamma_{1,0,0} + \lambda \geq \tilde{R}_{x^j}^1 + \tilde{R}_{x^j}^0 \]

\[L_{x^j}^{(i,j)} - \beta \sum_{y^i \in \mathcal{X}^i, y^j \in \mathcal{X}^j} P_{x^j y^i} P_{x^j y^j} L_{y^i y^j}^{(i,j)} + \gamma_{1,1} + 2\lambda \geq \tilde{R}_{x^j}^1 + \tilde{R}_{x^j}^1, \text{ if } (1,1) \in \mathcal{A}^2.\]

It can easily be verified that \(D^2\) is the dual of \(\Gamma^2\) and it follows by strong duality that \(Z^2 = L_2(x_0)\). ■

### 3.2.8 Empirical results

We test our policy, INDEX, in a restless bandit setting against a number of other policies. For each problem, there is a specified number of projects, \(N\), and uniform number of states for each project \(|\mathcal{X}|\). The data for the problems was randomly generated. Both passive and active rewards are drawn from a Uniform(0,10) distribution, and transition probabilities are drawn from a Uniform(0,1) and subsequently normalized to sum to one. We used the same data for Problems 3-6. The heuristics implemented follow.

\(Z_{\text{Greedy}}\): Estimated (through simulations) expected value of the greedy heuristic which at each time activates the \(K\) projects with the highest immediate activated reward.
| Problem | \((N, |X|^K)\) | \(\beta\) | \(Z_{\text{Greedy}}\) | \(Z_{\text{Greedy}}\) | \(Z_{\text{PDDH}}\) | \(Z_{\text{INDEX}}\) | \(Z^*\) | \(Z^1\) |
|---------|----------------|-------|----------------|----------------|----------------|----------------|--------|--------|
| Problem 1 | (10,7,5) | 0.20 | 70.7 | 71.9 | 71.96 | 71.72 | 72.17 |
|         |             | 0.50 | 117.67 | 121.46 | 121.62 | 121.15 | 122.44 |
|         |             | 0.90 | 620.13 | 650.19 | 651.88 | 646.46 | 659.23 |
| Problem 2 | (10,7,1) | 0.20 | 74.30 | 74.60 | 74.62 | 74.58 | 74.76 |
|         |             | 0.50 | 106.92 | 110.74 | 110.83 | 110.69 | 111.59 |
|         |             | 0.90 | 563.02 | 585.63 | 588.78 | 588.89 | 597.20 |
| Problem 3 | (5,3,1) | 0.20 | 31.59 | 39.62 | 39.62 | 39.63 | 39.63 | 39.91 |
|         |             | 0.50 | 54.16 | 64.13 | 64.16 | 64.16 | 64.16 | 65.20 |
|         |             | 0.90 | 291.45 | 324.09 | 324.78 | 324.80 | 324.83 | 332.71 |
| Problem 4 | (5,3,2) | 0.20 | 39.00 | 42.96 | 42.96 | 42.96 | 42.96 | 43.27 |
|         |             | 0.50 | 61.33 | 67.72 | 67.80 | 67.82 | 67.82 | 69.06 |
|         |             | 0.90 | 350.56 | 358.23 | 358.63 | 358.23 | 358.74 | 362.11 |
| Problem 5 | (5,3,3) | 0.20 | 42.11 | 42.44 | 42.44 | 42.44 | 42.44 | 42.54 |
|         |             | 0.50 | 63.89 | 65.37 | 65.44 | 65.45 | 65.45 | 65.97 |
|         |             | 0.90 | 284.02 | 294.22 | 294.32 | 293.84 | 294.38 | 297.04 |
| Problem 6 | (5,3,4) | 0.20 | 38.28 | 38.64 | 38.67 | 38.64 | 38.64 | 38.89 |
|         |             | 0.50 | 57.12 | 58.64 | 58.74 | 58.74 | 58.74 | 59.74 |
|         |             | 0.90 | 273.45 | 283.82 | 284.10 | 284.10 | 284.10 | 286.45 |

Table 3.1: Results of restless bandit heuristics.

- \(Z_{\text{Greedy}}\): Estimated (through simulations) expected value of the greedy heuristic which at each time activates the \(K\) projects with the highest immediate active minus passive reward.
- \(Z_{\text{PDDH}}\): Estimated (through simulations) expected value of the primal-dual heuristic defined in Bertsimas and Niño-Mora [11].
- \(Z_{\text{INDEX}}\): Estimated (through simulation) expected value of INDEX.
- \(Z^*\): Optimal value as calculated using dynamic programming.
- \(Z^1\): Optimal value obtained by solving \(\min_\lambda L(x; \lambda)\). \(Z^1 = L(x)\).

The size of Problems 1 and 2 do not permit the calculation of \(Z^*\). Note that \(Z^*\) is not monotonic, which is not surprising since there are active as well as passive rewards.
and we require $e'u = K$. The results are shown in Table 3.1 and they indicate the following:

1. INDEX and PDH both perform almost optimally.

2. Greedy$_b$ is a very good heuristic; it has scores similar to INDEX and PDH.

3. PDH outperforms Greedy$_a$, which is consistent with the findings of Bertsimas and Niño-Mora [11].

4. In our experiments, INDEX chose the optimal $\lambda$ very quickly.

### 3.3 Multiclass queueing networks

Consider a set of stations that must service the arrival of different classes of customers in discrete time. Each class of customers has its own arrival rate and service time. Given the routes the customers follow through the network, the problem is to determine a sequencing policy of the classes at the service stations that minimizes a linear combination of the sojourn times of each customer class. In this problem, the coupling constraints occur at service stations in the form of the sequencing decisions. This problem arises in packet-switching communication networks, job shop manufacturing systems, scheduling of multi-processors and multi-programmed computer systems, (see Avram et al. [2]).

The first main result of this section is that multiclass queueing networks can be formulated as weakly coupled dynamic optimization problems. Once this is established we computationally investigate the effectiveness of our Lagrange-based policies and bounds in this setting. We find that our policies perform near optimal and the bounds on the true cost to go are strong.

#### 3.3.1 Structure

The section is structured as follows. In Section 3.3.2, we formulate multiclass queueing networks. In Section 3.3.3 we show that a finite state version of the problem is a
weakly coupled dynamic optimization problem. In Section 3.3.4 we use the results of Section 3.3.3 to find the Lagrangian decoupled formulation of multiclass queueing networks. Finally in Section 3.3.5 we present computational results of our Lagrange-based policies and bounds on two queueing networks.

3.3.2 Formulation

The following formulation closely follows the formulation provided by Avram et al. [2]. A queueing network has $M$ stations and $N$ job classes ($N \geq M$) and $K$ sources. A job class is specific to a station; as a particular job flows through the network, it defines a different class in each station. A single station may service multiple job classes, each with its own service rate and holding cost. The arrivals for each job class either come from another station or from outside the system. Each class $i$ that has arrivals from another station has a unique previous class $p(i)$. After the jobs are processed, they either become members of a new class or they leave the system. If jobs of class $i$ do not leave the system after processing, they have a unique next class $n(i)$, otherwise $n(i) = 0$. For $i = 1, \ldots, N$, let $x_i^t$ denote the number of jobs of class $i$ at time $t$ and $s(i)$ be the index of the station processing jobs of class $i$. The cost per unit time for holding a job of class $i$ is $c_i$. Finally, the control variables, $u_i^t$ will indicate whether or not station $s(i)$ is serving class $i$ at time $t$ ($u_i^t \in \{0, 1\}$). If $u_i^t = 1$, then the first customer of class $i$ at station $s(i)$ leaves the queue, as indicated by the random variable $q_i^t$. The departure times of jobs leaving a class are geometrically distributed, conditional on the controller acting on them. In addition, we allow for pre-emption; the controller can stop working on a class at any time and can resume working on a class at any time.

An example queueing network is shown in Figure 3-1. The left hand box represents Station 1, and the right hand box represents Station 2. In each time period, and for both stations, there is a probability of 0.08 that a new customer arrives. $\mu_1, \ldots, \mu_4$ represent the probability of customer departure given the customer is being served for queueing classes 1 through 4 respectively. For example, if the controller serves queue 1, there is a probability of 0.12 that the job will finish in this period. The coupling
constraint is that the controller can serve at most one queue at each station.

A mathematical formulation of discrete time multiclass queueing networks follows.

**State** The state of the system at time $t$ is $x_t = (x^1_t, \ldots, x^N_t)$ where $x^i_t$ is the number of customers of class $i$ in queue and being served. Time is discrete.

**Controls** The controls are $u_t = (u^1_t, \ldots, u^N_t)$ where $u^i_t = 1$ if the server $s(i)$ is serving class $i$, and is zero otherwise.

**Randomness** The randomness at time $t$ is $q_t$ where $q^i_t = 1$ if a customer of class $i$ leaves the queue. If $p(i)$ is a source, then $q^i_t = 1$ is the event that a customer joins queue $i$. The distribution of the elements of $q_t$ follow.

$$u^i_t = 0 \implies q^i_t = 0$$
$$u^i_t = 1 \implies \begin{cases} q^i_t = 1 & \text{with probability } \mu_i \\ q^i_t = 0 & \text{with probability } 1 - \mu_i. \end{cases}$$

For the case that $p(i)$ is a source, we write

$$q^{p(i)}_t = \begin{cases} 1 & \text{with probability } \gamma_i \\ 0 & \text{with probability } 1 - \gamma_i. \end{cases}$$

In some cases we will denote the arrivals from sources by the random variable $b^i_t = q^{p(i)}_t$.

**Dynamics** The dynamics of the system are

$$x^i_{t+1} = \begin{cases} 0 & \text{if } x^i_t - q^i_t + q^{p(i)}_t \leq 0 \\ x^i_t - q^i_t + q^{p(i)}_t & \text{otherwise}. \end{cases}$$
Let $A$ be the node-node incidence matrix with $A_{ij} = 1$ if class $i$ feeds into class $j$, $A_{ij} = -1$ if class $i$ is fed by class $j$, and $A_{ij} = 0$, otherwise. For example, the network illustrated in Figure 3-1 has

$$A = \begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 \\
0 & 0 & 1 & 0
\end{bmatrix}$$

In the matrix above, the first column corresponds to customers arriving with probability $\lambda_1$. Likewise, the fourth column corresponds to customers arriving with probability $\lambda_2$. It follows that the transitions can be expressed as $x_{t+1} = [x_t + Aq_t + b_t]^+$, where $[\cdot]^+$ represents the positive component of its argument. We will often suppress the reference to time $t$, and write the new state of the system as $y = f(x, u, q)$.

**Constraints** The sum of the control variables $u_i^t$ for all the classes processed at the same station must be less than one. This constraint is represented by the $m \times n$ binary matrix $D$, where

$$D_{ij} = \begin{cases}
1 & \text{if } s(j) = i \\
0 & \text{otherwise.}
\end{cases}$$

For example, the network illustrated in Figure 3-1 has

$$D = \begin{bmatrix}
1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0
\end{bmatrix}$$

In the constraint matrix above, the first row corresponds to the constraints at the left-most station in Figure 3-1. The constraints are written as $Du_t \leq e$. There is an additional constraint that $x_i^t = 0 \Rightarrow u_i^t = 0$. 

81
**Objective** Our objective is to solve

\[
\min_{\pi} \mathbb{E}\left[ \sum_{t=0}^{\infty} \beta^t c'x_t \right],
\]

for a feasible policy \( \pi \).

The above problem can be solved by finding a solution to the Bellman equation

\[
J(x) = \min_{u \in \{0,1\}^N} \left\{ c'x + \beta \mathbb{E}\left[ J([x + Aq + b]^+)|u] \right] | Du \leq e \right\}.
\]

A shortfall of this approach is the possibly infinite range of \( x \). For this reason we will consider a variation of the problem in which there is a maximum allowable queue size.

### 3.3.3 Buffered multiclass queueing networks as weakly coupled dynamic optimization problems

In this section we relate our formulation of multiclass queueing networks to weakly coupled dynamic optimization problems, as defined in Section 2.1.1. This is achieved by first restricting multiclass queueing networks to buffered multiclass queueing networks. The dynamic of the network change in that customers that arrive to a queue length equal to the buffer size immediately leave the system. We show that work flows of jobs can be treated as independent subproblems that are linked by the constraints at stations. The will lead us to the main result of this section, Theorem 10, which says that buffered multiclass queueing networks are weakly coupled dynamic optimization problems, as defined in Section 2.1.1. The significance of this result is that it allows us to use the policies and bounds of Chapter 2.

**Theorem 10** Under the formulation of Section 3.3.2, with the addition of buffers, multiclass queueing networks are weakly coupled dynamic optimization problems, as defined in Section 2.1.1.
Before we prove this result we need a few definitions and other results. We begin by defining work flows of jobs as projects, i.e., subproblems that can be solved in isolation were it not for the constraints. For \( k \in K = \{1, ..., K\} \), let \( i_k \) denote the class of which the source is external and \( p(i_k) \) does not exist. Let \( p^1(i) = p(i) \) and \( p^n(i) = p(p^{n-1}(i)) \). Then

\[
S_k = \{ i \in \mathcal{N} | \exists n \geq 1, p^n(i) = i_k \}
\]

is the chain of classes that originate at source class \( i_k \). For example, in Figure 3-1 \( S_1 = \{1, 2\} \) and \( S_2 = \{3, 4\} \). We will use the notation \( x_k, u_k \) and \( c_k \) to denote controls, queues and costs associated with all classes in chain \( k \). For example, \( x_k = (x^i)_{i \in S_k} \).

Let \( q_k = (q)_{i \in S_k} \). We have the following result.

**Lemma 5** We have that

\[
P(q|u) = \prod_{k=1}^{K} P(q_k|u_k).
\]

**Proof.** This clearly follows from the definition of the dynamics of \( q \). \( \blacksquare \)

The following corollary is immediate.

**Corollary 1** We have that

\[
P(y|x, u) = \prod_{k=1}^{K} P(y_k|x_k, u_k).
\]

**Proof.** Clearly the state dynamics \( y_k = f(x, u, q) \) can be written as \( y_k = f(x_k, u_k, q_k) \).

Application of Lemma 5 yields the result. \( \blacksquare \)

We have now established the most important feature linking our problem to weakly coupled dynamic optimization problems. The next result involves buffers and it will serve two purposes: First, the presence of buffers guarantees that our state space is finite which a requirement of weakly coupled dynamic optimization problems; Second, it will make computational analysis significantly easier.
We denote by $B$ the buffer, which is the maximum allowable queue size. The effect of the buffer is that customers that arrive at a queue of size $B$ immediately leave the system; otherwise the dynamics of the system are the same as before. The adjusted dynamic equations are

$$
x^i_{t+1} = \begin{cases} 
0 & \text{if } x^i_t + q^i_t - q^{p(i)}_t \leq 0, \\
B & \text{if } x^i_t + q^i_t - q^{p(i)}_t \geq B, \\
x^i_t + q^i_t - q^{p(i)}_t & \text{otherwise.}
\end{cases} \tag{3.38}
$$

The dynamics will often be written as $y = f_B(x, u, q)$. We denote by $J_B(x)$ the optimal cost of the problem with buffer $B$:

$$J_B(x) = \min_{u \in \{0,1\}^N} \left\{ c^T x + \beta \mathbb{E}[J_B(f_B(x, u, q)|x, u)|Du \leq e]\right\}.$$

Note that Corollary 1 still holds with the presence of a buffer. We are now ready to prove the main result.

**Proof of Theorem 10**

The proof involves writing the elements of Section 3.3.2 as they appear in Section 2.1.1. This is a discrete time problem that satisfies the following conditions.

**State** The state of the system is $x = (x_1, ..., x_K)$ where $x_k = (x^i)_{i \in S_k}$, $x_k \in X_k = \{0, ..., B\}^{S_k}$.

**Controls** The controls are $u = (u_1, ..., u_K)$ where $u_k = (u^i)_{i \in S_k}$, $u^i \in \{0,1\}$.

**Randomness** The randomness per period is $q = (q_1, ..., q_K)$, $q^i \in \{0,1\}$. From Lemma 5 for buffered networks, the randomness satisfies

$$P(q|u) = \prod_{k=1}^{K} P(q_k|u_k).$$

**Dynamics** The dynamics can be written as $y_k = f(x_k, u_k, q_k)$. Furthermore, from
Corrolary 1 we have

$$P(y|u, x) = \prod_{k=1}^{K} P(y_k|u_k, x_k).$$

Costs The cost accrued each period is $c'x = \sum_{k=1}^{K} c'_k x_k$.

Constraints The constraints linking the controls in each period satisfy $Du \leq e$.

Objective This is an infinite horizon problem with discount factor $\beta$ with the following objective:

$$\max_{\pi} E \left[ \sum_{t=0}^{\infty} \sum_{k=1}^{K} \beta^t c'_k x_k(t) \right].$$

In the next section we apply the Lagrangian results for weakly coupled dynamic optimization problems to buffered multiclass queueing networks.

3.3.4 Lagrangian decoupled formulation and LP methods

An examination of the exact formulation of the discrete multiclass queueing network reveals the following similarity to weakly coupled dynamic optimization problems. Chains of sequential queues must share station $i$'s resources, represented by constraint $D_i u \leq 1$. In this section, we apply the Lagrange decoupling results of Chapter 2 to the queueing network. In particular, we have one Lagrange multiplier per station. The astute reader will note that it is possible to have two classes from the same chain $k$ at a particular station, i.e., there exist $i, j \in S_k$ with $s(i) = s(j)$. The subproblem $L^k_\beta(\cdot)$ is affected in the following way: We enforce $u^i + u^j \leq 1$ and penalize decisions at rate $u^i \lambda_{s(i)}$ and $u^j \lambda_{s(j)}$ respectively. Effectively we are restricting $u_k \in U_k$.

The key result is that the Lagrangian relaxed problem involved solving chains of
work flows in isolation. The Lagrangian decoupling follows. We write

$$L_B(x; \lambda) = \sum_{k \in K} L^k_B(x_k; \lambda) - e' \lambda \frac{1}{1-\beta},$$

(3.39)

where

$$L^k_B(x_k; \lambda) = \min_{u_k} \left\{ c_k^L x_k - \sum_{i \in S_k} \lambda_{x(i)} u^i + \beta E_{q_k} \left[ L^k_B(f_B(x_k, u_k, q_k); \lambda) \right] \right\}.$$

Note that there is a negative sign in front of $e' \lambda \frac{1}{1-\beta}$ in Eq. (3.39) due to the fact that this is a minimization problem. This formulation is consistent with the results of Theorem 1. We solve the problem of minimizing $L(x; \lambda)$ using linear programming.

In the notation used we suppress all reference to $B$ for aesthetic reasons. We use the following notation: $\alpha_{x_k}^k = 1$ if the initial state of $x_i$ for $i \in S_k$ is $x_k$; $c_{x_k}^k = c_k^L x_k$; $p_{x_k,y_k}^k = P(y_k \mid x_k, k)$. The decision variables are $\lambda$ and $L^k_{x_k}$ for $x_k \in X_k, k = 1, \ldots, K$. The formulation follows.

$$L_B(x) = \max \sum_{k \in K} \sum_{x \in X_k} \alpha_{x_k}^k L^k_{x_k} - \frac{1}{1-\beta} e' \lambda$$

s.t. $L^k_{x_k} \leq c_{x_k}^k - u' \lambda + \beta \sum_{y_k \in X_k} p_{x_k,y_k}^k L^k_{y_k}$,

(3.40)

$$\forall u \in U, \forall x \in X_k, \forall k \in K.$$

It follows that $L_B(x) \leq J_B(x)$.

3.3.5 Numerical results

In this section, we apply our Lagrangian methods to two networks. In particular, we test the accuracy of the Lagrangian bounds and we test the performance of the integer optimization Lagrange-based policy of Section 2.4.3. The first example shows that for small networks our policies are very close to optimal. The second example demonstrates that our Lagrange-based policy not only can solve intractable problems, but also produces high quality results.
Figure 3-2: The decoupled queuing network from Example 1. Here, $c, \mu$ are the same as in Fig. 3-1, but in addition there is a cost $\lambda_1$ for working on classes 1 and 4, and a cost of $\lambda_2$ for working on classes 2 and 3.

Figure 3-3: The results of the exact and approximate algorithms for the network shown in Fig. 3-1, as a function of the buffer.

Example 1

Consider the network in Fig. 3-1. This is the discrete time version of a network analyzed in de Farias and van Roy [20]. We set $x_0 = (2, 2, 2, 2)$ and use $c = (1, 1, 1, 1)$ and $\beta = 0.95$. $L(x_0)$ versus $J(x_0)$ as a function of the buffer size are shown in Fig.3-3. The results of the polices used is shown in Table 3.2.

Example 2

Consider the network in Fig. 3-4. We set $x_0 = 2e$ and $c = e$ and $\beta = 0.95$. This problem was too large to solve exactly. This is a high dimension problem; for a buffer size of $B$, the number of possible states is $O(B^8)$. The decoupled problem consists of four problems, each with $O(B^2)$ states. Specifically, the decoupled problem has
four chains, each chain with two stations. The results of the policies used is shown in Table 3.2.

Simulation results

We simulated the results of three policies in the queueing networks of Examples 2 and 3. For both examples a buffer size of 15 was used. In addition to the three policies, we have the two bounds on the cost to go: one from the exact solution with a buffer size of 12, and one from $Z_0^H$. The description of the policies and bounds follows.

$Z_{\text{Greedy}}$: Estimated (through simulations) expected value of greedy heuristic which at each station works on the class of greatest length-weighted cost $c^T x^i$.

$Z_{\text{Lag}}$: Estimated (through simulations) expected value of Lagrange-based policy with $\lambda = 0$ using integer optimization, as described in Section 2.4.1. In particular, we set $\lambda = 0$ and solve the individual chains of work flow in isolation. The decision in the current period is the feasible maximizer of the cost to go based on the value estimates of the decoupled problems.

$Z_{\text{ML}}$: Estimated (through simulations) expected value of policy found by integer optimization as described in Section 2.4.1. In this case we set $\lambda$ equal to the
<table>
<thead>
<tr>
<th></th>
<th>$Z_{\text{Greedy}}$</th>
<th>$Z_{LZ}$</th>
<th>$Z_{\text{ML}}$</th>
<th>$Z^*$</th>
<th>$Z^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>162.01</td>
<td>136.33</td>
<td>132.56</td>
<td>132.45</td>
<td>128.05</td>
</tr>
<tr>
<td>Example 2</td>
<td>306.68</td>
<td>287.83</td>
<td>287.75</td>
<td></td>
<td>270.55</td>
</tr>
</tbody>
</table>

Table 3.2: Simulation results of policies for different queuing networks

argmax of (LP$_{\text{approx}}$). As mentioned in Section 2.4.3, this is equivalent to solving (LP$_{\text{approx}}$), and then choosing the choosing controls that have minimum slack.

$Z^*$ A lower bound on the true cost to go using a buffer size of 12. $Z^* = J_B(\mathbf{x})$

$Z^1$ A lower bound on the true cost to go found by solving Problem (3.40). $Z^1 = L_B(\mathbf{x})$. The results of 1000 simulations are shown in Table 3.2. Because of the large size of the network in Example 2, $Z^*$ was not computable. Note that for Example 2, the results of $Z_{LZ}$ and $Z_{\text{ML}}$ were not statistically different. The results of Tables 3.2 indicate the following:

1. $Z_{\text{ML}}$ is a very good policy and is very close to optimal.

2. $Z_{LZ}$ is a good policy despite its greater ease of computation.

The key result of these numerical experiments is that the Lagrange-based policies are close to optimal. This is clearly true in Example 1. For Example 2, extrapolation from Example 1 suggests that $Z_{ML}$ is closer to $Z^*$ than $Z^1$ is, and thus the Lagrange policies are very strong. Also important is the significant decrease in the size of the decoupled formulations. In Example 1, calculation of the exact cost to go, $J_B(\mathbf{x})$, for a buffer size of 10 took roughly 25 hours of computation, whereas computation of $L_B(\mathbf{x})$ for $B = 25$ took less than 30 minutes.

### 3.4 Supply chain inventory problems

Consider a serial supply chain consisting of a finite number of stations. At one end of the chain there is a production station where goods are produced, and at the other
end there is an outlet station where demand is received. The objective is to ship goods from the production station to the outlet station while accounting for holding costs, back order costs, and lead times, so as to minimize overall system costs. For the case that holding costs are linear, and lead times are deterministic, it has been shown by Clark and Scarf [18] that an echelon base-stock policy is optimal. Muharremoglu and Tsitsiklis [34] allow for random lead times, so long as the sequential ordering of units is preserved, and show that the problem can be decomposed into unit-customer pairs. The results of [18] and [34] are both for uncapacitated serial supply chains and unfortunately there are no known results for the capacitated case. In this section we present a Lagrangian approach to decoupling the capacitated problem into an uncapacitated problem and suggest a Lagrange-based heuristic for solving the problem.

3.4.1 Standard formulation

The uncapacitated serial supply chain problem can be described as follows. There are $N$ stations organized in a serial manner. Demands are received at station 1 and goods are produced at station $N + 1$. The goal of the supply chain manager is to minimize the sum of holding costs and back order costs in meeting customer demand over a horizon of $T$ periods. Goods from station $i$ are shipped to station $i - 1$, for $i = 2, \ldots, N + 1$ to meet the demand, and unmet demand is back-logged at station 1. There is a deterministic lead time of $B_i$ between stations $i + 1$ and $i$. In the capacitated problem there are constraints on the number of goods that can be held at particular stations, $C_i$. The standard formulation has the following variables.

\[ x_i^t \] the inventory level of station $i$ at time $t$. $x_i^1 < 0$ indicates a backlog of orders at station 1.
\[ y_{l,i}^t \] the number of units, at time $t$, to arrive at station $i$, $l$ periods from time $t$, for $l = 1, \ldots, B_i$.
\[ z_i^t \] the inventory level plus all demands on order for station $i$ at time $t$,
\[ z_i^t = x_i^t + \sum_{l=1}^{B_i} y_{l,i}^t. \]
Note that we have made a distinction between goods at a station, and the goods being shipped to a station. The reason is that in deciding how much to ship to the next station in a particular time period, one must know the number of goods immediately available. In the literature, it is common to ascribe holding costs at a station to the goods at that station plus all goods being delivered to the station. Thus, $z^i_t$ will be convenient for shorthand notation when dealing with holding costs. Furthermore, we will often use the following vector notation:

$$x_t = (x^1_t, \ldots, x^N_t)$$

$$y_t = (y^1_{t,1}, \ldots, y^1_{t,B_1}, y^2_{t,1}, \ldots, y^N_{t,B_N}).$$

In what follows we define the serial supply chain problem in mathematical detail.

**State** $(x_t, y_t)$

**Controls** $u_t = (u^1_t, \ldots, u^{N+1}_t)$ where $u^i_t \geq 0$ represents the number of units transported from station $i$ to station $i-1$ at time $t$ for $i = 2, \ldots, N+1$. $u^i_t$ represents the units given to back-logged demand and demand received in period $t$.

**Randomness** $d_t$ is the demand received at station 1 during time period $t$. $P(d_t)$ is known and is independent of previous demand and the controller’s actions.

**Dynamics**

$$x^1_{t+1} = x^1_t + y^2_{t,1} - d_t$$

$$x^i_{t+1} = x^i_t + y^i_{t,1} - u^i_t, \quad i = 2, \ldots, N,$$

$$y^i_{t+1,t} = y^i_{t+1,t+1},$$

$$y^i_{t+1,B_i} = u^{i+1}_t.$$

These dynamics will sometimes be expressed as $(x_{t+1}, y_{t+1}) = f(x_t, y_t, u_t, d_t)$.

**Constraints** The following constraints are enforced:

- $x^i_t + \sum_{i=1}^{B_i} y^i_{t,i} \leq C_i, \quad i = 2, \ldots, N,$
\( (x^1_t)^+ + \sum_{i=1}^{B_i} y^1_{i,t} \leq C_1 \).

This is an inventory constraint. Note that back orders do not count against the inventory at other locations in the system. We will denote by \( U_t(x_t, y_t) \) the set of all \( u_t \in \mathbb{N}^N \) that satisfy the above constraints.

**Costs** There are two costs incurred:

- **holding costs** \( h = (h_1, ..., h_N) \), where \( h_i \) denotes the cost per unit held and on order at station \( i \).
- **backorder cost** \( b \), the cost per unit of demand back-logged at station 1.

Costs are incurred at the beginning of period \( t = 1, ..., T \). We will often write the sum of these costs as

\[
g(x_t, y_t) = \sum_{i=2}^{N} h_i (x^i_t + \sum_{i=1}^{B_i} y^i_{i,t}) + h_1 \sum_{i=1}^{B_i} y^1_{i,t} + h_1 (x^1_t)^+ + b(x^1_t)^-.
\]

**Objective**

\[
\min_{\pi \in \Pi} \mathbb{E}_{d_t \sim \pi} \left[ \sum_{t=0}^{T} g(x_t, y_t) \right], \tag{3.41}
\]

where \( \Pi \) is a feasible policy.

In our model, the system events occur in the following order given time \( t \): holding costs and back order costs are accrued; orders \( u_t \) are executed; demand \( d_t \) is received; states evolve as time becomes \( t + 1 \). At station 1 orders are filled up to what station inventory will allow and unmet demand is back-logged. We assume that the current state satisfies the constraints, but that the next state, under all realizations of \( d_t \), must satisfy the constraints as well.

Having defined the state for this problem as we have, the natural procedure for this problem would be to formulate it as a Bellman equation and iteratively solve the problem for \( T \) periods to determine the optimal policy and expected value of the best policy.
Bellman equation

\[ J_t(x_t, y_t) = \min \ g(x_t, y_t) + E[J_{t+1}(x_{t+1}, y_{t+1})|u_t] \]
\[ \text{s.t. } x_{t+1}^i + \sum_{l=1}^{B_t} y_{t+1,i,l} \leq C_i, \ i = 2, ..., N \]
\[ (x_{t+1}^1)^+ + \sum_{l=1}^{B_t} y_{t+1,1,l} \leq C_1, \ \forall d_t. \]

Note that the constraint is in terms the next state. For the uncapacitated problem it has been shown that (i) the optimal policy is an echelon base stock policy, meaning the problem can be solved in smaller stages, Clark and Scarf [18], and (ii) the problem can be decomposed into single unit customer pairs, Muharremoglu and Tsitsiklis [34]. Unfortunately such methods do not work for the capacitated version. Our approach is to decouple the capacitated problem into an uncapacitated problem, and then use standard methods to arrive at an optimal policy for the Lagrangian relaxed problem. By selecting the right Lagrange multipliers, we hope to find an ordering policy that does not violate the original capacity constraints. First, we will present a decomposed formulation of the capacitated serial supply chain problem and show that it is in fact a weakly coupled dynamic optimization problem. This will motivate using the Lagrange-based policies developed in Chapter 2.

### 3.4.2 Decomposed formulation

In this section we consider the decomposed formulation that is introduced by Muharremoglu and Tsitsiklis [34]. Our approach to the formulation is slightly different. We begin by attaching a unique label \( k \geq 1 \) to all of the units that could possibly be shipped. In addition, we index every unit of demand placed in each period by an index \( m \).
We begin by defining the states.

\[
x^i_t(k) = \begin{cases} 
1 & \text{if unit } k \text{ is at station } i \text{ at time } t, \\
0 & \text{otherwise.}
\end{cases}
\]

\[
y^i_t,l(k) = \begin{cases} 
1 & \text{if unit } k \text{ is } l \text{ time stages from station } i \text{ at time } t, \\
0 & \text{otherwise.}
\end{cases}
\]

We introduce a station 0 where all units and demand are stored after matching up. We denote by \( x^0_t(k) = 1 \) that unit \( k \) has met a single demand by time \( t \) and is at station 0, and 0 otherwise. Note that

\[
\sum_{k=1}^{\infty} x^i_t(k) = x^i_t,
\]

\[
\sum_{k=1}^{\infty} y^i_t,l(k) = y^i_t,l
\]

\[
\sum_{i=0}^{N+1} x^i_t(k) + \sum_{i=1}^{N} \sum_{l=1}^{B_i} y^i_t,l(k) = 1.
\]

Analogous to the previous section, we define \( z^i_t(k) = x^i_t(k) + \sum_{l=1}^{B_i} y^i_t,l(k) \).

Clearly, no matter what demand arrives, it is always possible to consistently give priority of shipping to goods of lower index. Muharremoglu and Tsitsiklis [34] refer to this property as *monotonicity*. In our formulation monotonicity can be stated as

\[
k_1 < k_2 \Rightarrow x^{N+1}_t(k_1) = \sum_{i=j+1}^{N} \left( x^i_t(k_1) + \sum_{l=1}^{B_i} y^i_t,l(k_1) \right) + \sum_{l=0}^{L_i} y^i_t,l(k_1) \\
\leq x^{N+1}_t(k_2) = \sum_{i=j+1}^{N} \left( x^i_t(k_2) + \sum_{l=1}^{B_i} y^i_t,l(k_2) \right) + \sum_{l=0}^{L_i} y^i_t,l(k_2), \ j \geq 0.
\]

Having broken the state space into binary variables indicating the position of individual units we proceed by decomposing demand units \( d_t \) into individual units. In the previous section the random variable of interest was the number of goods that will arrive in a given time period. Our focus now will be on the time of arrival of the
\( m \)th unit of demand. We define
\[
d_t(m) = \begin{cases} 
1 & \text{if the } m \text{th unit of demand arrives at time } t, \\
0 & \text{otherwise.}
\end{cases}
\]

It follows that \( \sum_{m=1}^{\infty} d_t(m) = d(t) \), and \( \sum_{t=1}^{T} d_t(m) \leq 1 \). Knowing when demand \( m \) arrives is dependent on the number of units to arrive before it. We therefore define
\[
D_t = \text{the number of units of demand received before the start of time period } t,
\]
\[
= \sum_{s=1}^{t-1} \sum_{m=1}^{\infty} d_s(m).
\]

We have
\[
P(d_t(m) = 1|D_t) = \begin{cases} 
1 & \text{if } m \leq D_t, \\
P(d_t \geq m - D_t) & \text{if } m > D_t.
\end{cases}
\]

Of course, demands in each period are considered independent of each other, so one could simply augment states by \( m - D_t \). However, for sake of notation clarity we will refrain from doing this.

Having redefined the states we proceed by formalizing the specifics of the system.

**Formulation**

**State** \((x_t, y_t, D_t) = (x_t(1), y_t(1), x_t(2), y_t(2), \ldots, D_t)\)

**Randomness** \(d_t(m)\), where \(P(d_t(m) = 1|D_t)\) is known.

**Controls** \(u_t = (u_t^1(1), u_t^1(2), \ldots, u_t^{N+1}(1), u_t^{N+1}(2), \ldots)\) where
\[
u_t^i(k) = \begin{cases} 
1 & \text{if unit } k \text{ is released from station } i \text{ at time } t, \\
0 & \text{otherwise.}
\end{cases}
\]

As with previous variables, \( \sum_{k=1}^{\infty} u_t^i(k) = u_t^i \).
Costs

\[ g(x_t, y_t) = \sum_{i=2}^{N} h_i \sum_{k=1}^{\infty} z_i^1(k) + h_1 \sum_{i=1}^{B_1} \sum_{k=1}^{\infty} y_{i,t}^1(k) + h_1 \left( \sum_{k=1}^{\infty} x_i^1(k) \right)^+ + b \left( \sum_{k=1}^{\infty} x_i^1(k) \right)^- . \]

Dynamics

\[ x_{t+1}^i(k) = x_t^i(k) + y_{i,t}^1(k) - u_t^i(k), \quad i = 2, \ldots, N, \]

\[ x_{t+1}^1(k) = x_t^1(k) + y_{i,t}^2(k) - d_t(m), \quad \text{for} \ m > D_t \text{ and unique pairing } (k, m), \]

\[ x_{t+1}^0(k) = \begin{cases} 1 \cdot x_t^1(k) + y_{i,t}^2(k) = 1 \text{ and there exists demand } d_t(m) = 1 & \\
0 & \text{for unique pairing } (k, m), \end{cases} \]

\[ y_{t+1,i}^1(k) = y_{i,t+1}^1(k) \]

\[ y_{t+1,i}^j(k) = u_t^{i+1}(k) \]

\[ D_{t+1} = D_t + \sum_{k=1}^{\infty} d_t(k). \]

Constraints

\[ \sum_{k=1}^{\infty} x_i^1(k) + \sum_{l=1}^{B_1} \sum_{k=1}^{\infty} y_{i,t}^1(k) \leq C_t, \quad i \geq 2 \]

\[ \left( \sum_{k=1}^{\infty} x_i^1(k) \right)^+ + \sum_{l=1}^{B_1} \sum_{k=1}^{\infty} y_{i,t}^1(k) \leq C_1, \]

\[ u_t^i(k) \leq x_t^i(k). \]

The third constraint says that unit \( k \) cannot be released from station \( i \) unless it is already at station \( i \).

Objective

\[ \min_{\pi \in \Pi} E \left[ \sum_{t=1}^{T} g(x_t, y_t) \right] \]
Bellman equation

\[
J_t(x_t, y_t, D_t) = \min \ g(x_t, y_t) + E[J_{t+1}(x_{t+1}, y_{t+1}, D_{t+1})|u_t, D_t]
\]

s.t. \[\sum_{k=1}^{\infty} x_t^i(k) + \sum_{i=1}^{B_t} \sum_{k=1}^{\infty} y_{i,t}^i(k) \leq C_t, \ i \geq 2\]

\[\left(\sum_{k=1}^{\infty} x_t^i(k)\right)^+ + \sum_{i=1}^{B_t} \sum_{k=1}^{\infty} y_{i,t}^i(k) \leq C_j, \ \forall d_t,\]

\[u_t^i(k) \leq x_t^i(k).\]

Decomposition of the uncapacitated problem

The key result of Muhtarremercoglu and Tsitsiklis [34] is the decomposition of the serial supply chain problem into unit-demand pairs. The result relies on two properties: monotonicity of the state space, which we have confirmed of our formulation; and the optimality of committed policies. This second property requires that the kth unit will always be delivered to the kth unit of demand. As we have formulated the problem, the unit k arrives before unit k + 1. Similarly, demand k is made before demand k + 1.

Any matching other than a committed one could result in unnecessary back order or holding costs. So clearly committed policies are optimal.

It follows that the state dynamics can be written as

\[
x_{t+1}^i(k) = x_t^i(k) + y_{t,1}^i(k) - u_t^i(k), \ i = 2, ..., N,
\]

\[
x_{t+1}^1(k) = x_t^1(k) + y_{t,1}^2(k) - d_t(k)
\]

\[
x_{t+1}^0(k) = \begin{cases} 
1 & x_t^1(k) + y_{t,1}^2(k) = 1 \text{ and there exists demand } d_t(k) = 1 \\
0 & \text{otherwise,}
\end{cases}
\]

\[
y_{t+1,1}^i(k) = y_{t,1}^i(k)
\]

\[
y_{t+1,1}^i(k) = u_t^{i+1}(k)
\]

\[
D_{t+1} = D_t + \sum_{k=1}^{\infty} d_t(k).
\]

To fully achieve decomposition, we must show that the costs can be decomposed
into unit-demand pairs. We begin by writing the costs as first defined. First note that from monotonicity

\[ x_t^1(k_1) = 1 \Rightarrow x_t^1(k_2) \in \{0, 1\}, \; k_2 \neq k_1 \]
\[ x_t^1(k_1) = -1 \Rightarrow x_t^1(k_2) \in \{-1, 0\}, \; k_2 \neq k_1. \]

It follows that the cost per stage can be written as

\[
g(x_t, y_t) = \sum_{i=2}^{N} h_i \sum_{k=1}^{B_1} x_t^i(k) + h_1 \sum_{i=1}^{B_1} \sum_{k=1}^{B_1} y_{t,i}(k) + h_1 \left( \sum_{k=1}^{\infty} x_t^1(k) \right)^+ + h_1 \left( \sum_{k=1}^{\infty} x_t^1(k) \right)^- \\
= \sum_{k=1}^{\infty} \left( \sum_{i=2}^{N} h_i x_t^i(k) + h_1 \sum_{i=1}^{B_1} y_{t,i}(k) + h_1 (x_t^1(k))^+ + h_1 (x_t^1(k))^+ \right) \\
= \sum_{k=1}^{\infty} g(x_t(k), y_t(k)).
\]

Muharremoglu and Tsitsiklis [34] show that in the absence of capacity constraints that the problem decouples in the sense that the individual unit-demand pairings can be optimized independently. That is

\[
\min_{\pi \in \Pi} \mathbb{E} \left[ \sum_{t=1}^{T} g(x_t, y_t) \right] = \min_{\pi \in \Pi} \mathbb{E} \left[ \sum_{t=1}^{T} \sum_{k=1}^{\infty} g(x_t(k), y_t(k)) \right] \\
= \sum_{k=1}^{\infty} \min_{\pi \in \Pi} \mathbb{E} \left[ \sum_{t=1}^{T} g(x_t(k), y_t(k)) \right].
\]

This problem can be solved using dynamic optimization. The Bellman equation follows.

\[
J_t(x_t, y_t, D_t) = \sum_{k=1}^{\infty} J_t'(x_t(k), y_t(k), D_t),
\]

98
where

\[ J^k_t(x_t(k), y_t(k), D_t) = \]
\[ \min g(x_t(k), y_t(k)) + E[J^k_{t+1}(x_{t+1}(k), y_{t+1}(k), D_{t+1})|u_t(k), D_t] \]
\[ \text{s.t. } u_t^i(k) \leq x_t^i(k). \]

In contrast to the weakly coupled problems considered in Chapter 2, the randomness of the problem is not unique to each decoupled problem. Note that the number of demands already received by some time \( t \), \( D_t \), affects the time at which \( d_t(k) = 1 \) for all \( k \). However, this fact does not affect the decoupling since the it still holds that future costs to go can be computed independently.

**Decomposed Bellman equation**

We write the Bellman equation for the decomposed formulation of the capacitated supply chain inventory problem.

\[ J_t(x_t, y_t, D_t) = \min \sum_{k=1}^{\infty} g(x_t(k), y_t(k)) + E[J_{t+1}(x_{t+1}, y_{t+1}, D_{t+1})|u_t, D_t] \]
\[ \text{s.t. } \sum_{k=1}^{\infty} \left( x_{t+1}^i(k) + \sum_{l=1}^{B_t} y_{t+1,l}(k) \right) \leq C_t, \ i \geq 2 \]
\[ \sum_{k=1}^{\infty} \left( (x_{t+1}^i(k))^+ + \sum_{l=1}^{B_t} y_{t+1,l}(k) \right) \leq C_1, \ \forall d_t, \]
\[ u_t^i(k) \leq x_t^i(k). \]

The significance of this formulation is that one can easily see the weak coupling of constraints and the additivity of the costs. Moreover, we know that the uncapacitated problem is decomposable. It follows that the capacitated problem is a weakly coupled dynamic optimization problem and that the Lagrangian decoupling approach developed in Chapter 2 applies.
3.4.3 Lagrangian decoupling

Having defined the capacitated supply chain inventory problem as a weakly coupled dynamic optimization problem, the obvious heuristic approximating $J_1(x_1, y_1, 0)$ is to use the Lagrangian decoupling approach. At this time it will be convenient to change the time at which costs are accrued. In particular, costs for the inventory as a result of our controls $u_t(k)$ and the randomness $d_t$ will be added in the current stage. The reason for this adjustment is to ease notation when we penalize the constraints on the inventory with Lagrange multipliers. We begin by rewriting the new Bellman equation.

$$
J_t(x_t, y_t, D_t) = \min \mathbb{E} \left[ \sum_{k=1}^{\infty} g(x_{t+1}(k), y_{t+1}(k)) + J_{t+1}(x_{t+1}, y_{t+1}, D_{t+1}) | u_t, D_t \right]
$$

s.t. $\sum_{k=1}^{\infty} \left( x_{i+1}^i(k) + \sum_{l=1}^{B_i} y_{i+1,l}(k) \right) \leq C_i, \quad i \geq 2$

$\sum_{k=1}^{\infty} \left( (x_{i+1}^i(k))^+ + \sum_{l=1}^{B_i} y_{i+1,l}^l(k) \right) \leq C_1, \quad \forall d_t,$

$u_t^i(k) \leq x_t^i(k).$

Having defined the problem in this way, we have put emphasis of constraint compliance on the states of the next phase, instead of the decisions made in the current phase. Because of the close link between state and controls both methods yield the same results. Note that the current state is feasible. We proceed by attaching a Lagrange multiplier $\lambda_{t+1} = (\lambda_{t+1}^1, \ldots, \lambda_{t+1}^N)$ to the constraints that arise in the next time stage. It is clear that for the resulting cost vector has components $h_t + \lambda_{t+1}^i$ for $i = 1, \ldots, N$. For shorthand, we will write the new costs, as a function of state as

$$
g_\lambda(x_t, y_t) = \sum_{k=1}^{\infty} g_\lambda(x_t(k), y_t(k)).$$
We can then define the following Lagrangian function.

\[
L_t(x_t, y_t, D_t; \lambda_{t+1,T}) = \\
\min \ E \left[ \sum_{k=1}^{\infty} g(x_{t+1}(k), y_{t+1}(k)) + E[L_{t+1}(x_{t+1}, y_{t+1}, D_{t+1}; \lambda_{t+1,T})|u_t, D_t] \right] \\
- \sum_{i=1}^{N} \lambda_i C_i \\
s.t. \quad u_t^i(k) \leq x_t^i(k).
\]

From earlier results it is apparent that the above problem can be decomposed into separate Lagrangian functions. We write these explicitly.

\[
L_t(x_t, y_t, D_t; \lambda_{t+1,T}) = \sum_{k=1}^{\infty} L_t^k(x_{t+1}(k), y_{t+1}(k), D_{t+1}; \lambda_{t+2,T}) - \sum_{s=t+1}^{T} \sum_{i=1}^{N} \lambda_i^s C_i,
\]

where

\[
L_t^k(x_t(k), y_t(k), D_t) = \\
\min \ E \left[ g(x_{t+1}(k), y_{t+1}(k)) + L_{t+1}^k(x_{t+1}(k), y_{t+1}(k), D_{t+1}; \lambda_{t+2,T})|u_t(k), D_t \right] \\
s.t. \quad u_t^i(k) \leq x_t^i(k).
\]

The key result here then is that Lagrangian decomposition can be applied to a weakly coupled dynamic optimization problem a the successful heuristics of earlier sections can be applied.

### 3.5 Conclusions

In this chapter we have investigated how our decoupling heuristic can be applied to a variety of problems. We have used the concept of decomposition to produce bounds on the cost to go that are easy to calculate for otherwise intractable problems. The results we obtain are: (i) A Lagrange based policy for multiarmed bandits that reproduces Gittins index policy; (ii) A Lagrange-based policy for restless bandits that
outperforms existing heuristics; (iii) A dynamic programming interpretation of an existing exact and approximate linear programming formulations of restless bandits; (iv) A new interpretation of an existing heuristic for restless bandits; (v) A new interpretation of a linear programming formulation for the problem of determining Gittins index; (vi) Higher-order coupling methods that produce better bounds on the cost to go than full decoupling; (vii) Decoupling multiclass queueing networks results in serial queueing networks with one decision per station and additional cost per work on queuing class; (viii) Numerical results that show the strong performance of our Lagrange-based policies and Lagrangian bounds for multiclass queueing networks; (ix) The formulation of capacitated serial supply chains as weakly coupled dynamic optimization problems. Future work in this area could include more complex schemes for achieving better bounds, and feasible heuristics based on the decomposed solutions.
Chapter 4

Extensions to Dynamic Linear Systems under Constraints

In this chapter we investigate a class of non-decomposable problems, that is, a class of dynamic optimization problems that cannot be decoupled by Lagrange multipliers. For many of these problems the addition of constraints makes the problem no more difficult than the original. However, for some problems the unconstrained version is easily solved, and in some cases permits a closed form solution. In particular, we investigate quadratic-cost linear systems. We take a Lagrangian approach to these problems and reduce them to an unconstrained version. This will allow us to derive a closed-form solution as a function of the Lagrange multipliers.

4.1 Formulation

Consider the system

\[ x_{t+1} = A_t x_t + B_t u_t + w_t, \quad t = 0, 1, \ldots, T - 1, \]

and the quadratic cost

\[ \mathbb{E}_{w_i} \left[ x_T' Q_T x_T + \sum_{t=0}^{T-1} (x_t' Q_t x_t + u_t' R_t u_t) \right]. \]
In addition, at every time \( t = 0, \ldots, T - 1 \) there are constraints

\[ F_t u_t \leq g_t. \]

The following definitions are from Bertsekas [7] p. 130. In these expressions, \( x_t \) and \( u_t \) are vectors of dimension of \( n \) and \( m \), respectively, and the matrices \( A_t, B_t, Q_t, R_t, F_t \) and vector \( g_t \) are given and have appropriate dimension. We assume matrices \( Q_t \) are positive semidefinite symmetric, and the matrices \( R_t \) are positive definite symmetric. The disturbances \( w_t \) are independent random vectors with given probability distributions that do not depend on \( x_t \) and \( u_t \). Furthermore, each \( w_t \) has zero mean and finite second moment.

Writing the Bellman equation, we have

\[ J_T(x_T) = x_T^T Q_T x_T, \]

\[ J_t(x_t) = \min_{u_t} \left\{ E_x [x_t' Q_t x_t + u_t' R_t u_t + J_{t+1}(A_t x_{t+1} + B_t u_{t+1} + w_t)] \mid F_t u_t \leq g_t \right\}. \]

The goal is to find \( J_0(x_0) \). For the unconstrained version, there is a closed form solution of the problem that involves solving the discrete-time Ricatti equations, (see Bertsekas [7]). However, for the constrained problem there is no known closed form solution. Indeed, at every time stage one must solve a linearly constrained non-linear optimization problem. We proceed by using Lagrangian relaxation of the constraints.

### 4.2 Lagrangian decomposition

As in Section 2.3 we take a Lagrangian approach to decomposing the dynamic program computing \( J(x_t) \) into an easier problem. Let

\[ L_T(x_T) = x_T^T Q_T x_T, \]

\[ L_t(x_t, \lambda_{t+1}) = \min_{u_t} \left\{ E_x [x_t' Q_t x_t + u_t' R_t u_t + 2\lambda_{t+1}' (F_t u_t - g_t)] + L_{t+1}(x_{t+1}, \lambda_{t+1, t+1})\right\}. \]
As in Section 2.3, we have used the notation $X_{t,T} = (\lambda_t', \ldots, \lambda_T')$. We have relaxed the constraint with $2\lambda_t$ to ease notation in future equations.

**Theorem 11**

$$L_t(x_t; \lambda_{t,T}) = x'_t K_{t+1} x_t + \sum_{s=t}^{T-1} \mathbb{E}[w'_s K_{s+1} w_s] - \lambda'_{t,T} H_{t,T} \lambda_{t,T} + 2(D_t x_t - G_t)' \lambda_{t,T}, \quad (4.1)$$

where

$$K_T = Q_T,$$

$$K_t = A_t'(K_{t+1} - K_{t+1} B_t M_t B'_t K_{t+1}) A_t + Q_t,$$

$$D_{T-1} = -F_{T-1} M_{T-1} B_{T-1}' Q_T A_{T-1},$$

$$D_t = \begin{bmatrix} -F_t M_t B'_t K_{t+1} A_t \\ (D_{t+1} - D_{t+1} B_t M_t B'_t K_{t+1}) A_t \end{bmatrix},$$

$$H_{T-1} = F_{T-1} M_{T-1} F_{T-1},$$

$$H_t = \begin{bmatrix} F_t M_t F'_t & F_t M_t B'_t D'_{t+1} \\ D_{t+1} B_t M_t F'_t & H_{t+1} + D_{t+1} B_t M_t B'_t D'_{t+1} \end{bmatrix},$$

$$M_t = (R_t + B'_t K_{t+1} B_t)^{-1}.$$ 

Furthermore, $M_t$ and $K_t$ are positive definite and $H_t$ is positive semidefinite.

**Proof.** We will prove the case for $t = T$ and proceed by induction. Consider the single-stage relaxed version of this problem with $t = T - 1$ defined by

$$L_{T-1}(x_{T-1}; \lambda_{T-1}) = \min_{u_{T-1}} \mathbb{E} \left[ x'_{T-1} Q_{T-1} x_{T-1} + u'_{T-1} R_{T-1} u_{T-1} + (A_{T-1} x_{T-1} + B_{T-1} u_{T-1} + w_{T-1})' Q_T (A_{T-1} x_{T-1} + B_{T-1} u_{T-1} + w_{T-1}) \right] + 2\lambda'_{T-1} (F_{T-1} u_{T-1} - g_{T-1}).$$
Using the fact that \(E[w'_{T-1}Q_T(A_{T-1}x_{T-1} + B_{T-1}u_{T-1})] = 0\) we can eliminate the term \(E[w'_{T-1}Q_T(A_{T-1}x_{T-1} + B_{T-1}u_{T-1})]\) and obtain

\[
L_{T-1}(x_{T-1}; \lambda_{T-1})
= \min_{u_{T-1}} \left\{ -2\lambda'_{T-1}g_{T-1} + x'_{T-1}Q_{T-1}x_{T-1} + u'_{T-1} (R_{T-1} + B'_{T-1}Q_TB_{T-1})u_{T-1} \\
+ x'_{T-1}A'_{T-1}Q_TA_{T-1}x_{T-1} + 2(x'_{T-1}A_{T-1}Q_TB_{T-1} + \lambda'_{T-1}F_{T-1})u_{T-1} \\
+ E[w'_{T-1}Q_Tw_{T-1}] \right\}.
\]

By differentiating with respect to \(u_{T-1}\) and by setting the derivative equal to zero, we have

\[
(R_{T-1} + B'_{T-1}Q_TB_{T-1})u_{T-1} = -(B'_{T-1}Q_TA_{T-1}x_{T-1} + F'_{T-1}\lambda_{T-1}).
\]

The matrix multiplying \(u_{T-1}\) on the left is positive definite since \(R_{T-1}\) is positive definite and \(B'_{T-1}Q_TB_{T-1}\) is positive semidefinite. It follows that the minimizing control of this Lagrangian problem is

\[
u^*_{T-1} = -(R_{T-1} + B'_{T-1}Q_TB_{T-1})^{-1}(B'_{T-1}Q_TA_{T-1}x_{T-1} + F'_{T-1}\lambda_{T-1}).
\]

By substitution into the expression for \(L_{T-1}\), we have

\[
L_{T-1}(x_{T-1}; \lambda_{T-1})
= -2\lambda'_{T-1}g_{T-1} + E[w'_{T-1}Q_Tw_{T-1}] + x'_{T-1}(Q_{T-1} + A'_{T-1}Q_TA_{T-1})x_{T-1} \\
-(B'_{T-1}Q_TA_{T-1}x_{T-1} + F'_{T-1}\lambda_{T-1})'(R_{T-1} + B'_{T-1}Q_TB_{T-1})^{-1}(B'_{T-1}Q_TA_{T-1}x_{T-1} + F'_{T-1}\lambda_{T-1}) \\
-x'_{T-1}A'_{T-1}Q_TB_{T-1}(R_{T-1} + B'_{T-1}Q_TB_{T-1})^{-1}B'_{T-1}Q_TA_{T-1}x_{T-1} \\
-x'_{T-1}A'_{T-1}Q_TB_{T-1}(R_{T-1} + B'_{T-1}Q_TB_{T-1})^{-1}F'_{T-1}\lambda_{T-1} \\
-2x'_{T-1}A'_{T-1}Q_TB_{T-1}(R_{T-1} + B'_{T-1}Q_TB_{T-1})^{-1}F'_{T-1}\lambda_{T-1}.
\]
We can rewrite this last expression as

\[
L_{T-1}(x_{T-1}; \lambda_{T-1}) = x'_{T-1}K_{T-1}x_{T-1} + \mathbb{E}[w'_{T-1}Q_T w_{T-1}]
- \lambda'_{T-1}H_{T-1}\lambda_{T-1} + 2(D_{T-1}x_{T-1} - g_{T-1})'\lambda_{T-1},
\]

where

\[
K_{T-1} = A'_{T-1}(Q_T - Q_T R_{T-1} (R_{T-1} + B'_{T-1}Q_T B_{T-1})^{-1}B'_{T-1}Q_T)A_{T-1} + Q_{T-1}
\]

\[
H_{T-1} = F_{T-1}(R_{T-1} + B'_{T-1}Q_T B_{T-1})^{-1}F'_{T-1}
\]

\[
D_{T-1} = -F_{T-1}(R_{T-1} + B'_{T-1}Q_T B_{T-1})^{-1}B'_{T-1}Q_T A_{T-1}.
\]

\(K_{T-1}\) is known as the Ricatti equation. We assume the result is true for \(t + 1\) and show it is correct for stage \(t\).

\[
L_t(x_t; \lambda_{t,T}) \]
\[
= \min_{u_t} \mathbb{E}\left[x'_tQ_t x_t + u'_t R_t u_t + L_{t+1}(A_t x_t + B_t u_t + w_t; \lambda_{t+1,T}) + 2\lambda'_t(F_t u_t - g_t)\right]
\]
\[
= \min_{u_t} \left\{ \mathbb{E}\left[x'_tQ_t x_t + u'_t R_t u_t + (A_t x_t + B_t u_t + w_t)' K_{t+1}(A_t x_t + B_t u_t + w_t)
+ 2\lambda'_{t,T} \left( D_{t+1}(A_t x_t + B_t u_t + w_t) - G_{t+1} \right) \right] - \lambda'_{t+1,T} H_{t+1} \lambda_{t+1,T} \right\}
\]
\[
+ 2\lambda'_{t,T} (F_t u_t - g_t) + \sum_{s=t+1}^{T-1} \mathbb{E}[w'_s K_{s+1} w_s]
\]
\[
= \min_{u_t} \left\{ x'_tQ_t x_t + u'_t R_t u_t + x'_t A'_{t+1} K_{t+1} A_t x_t + \sum_{s=t}^{T-1} \mathbb{E}[w'_s K_{s+1} w_s]
+ u'_t B'_{t+1} K_{t+1} B_t u_t + 2x'_t A'_{t+1} K_{t+1} B_t u_t + 2\lambda'_{t,T} F_t u_t + 2\lambda'_{t+1,T} D'_{t+1} B_t u_t
- \lambda'_{t+1,T} H_{t+1} \lambda_{t+1,T} + 2x'_t A_{t} D_{t+1} \lambda_{t+1,T} - 2\lambda'_{t+1,T} G_{t+1} \right\}.
\]

By differentiating with respect to \(u_t\) and by setting the derivative equal to zero, we obtain

\[
(R_t + B'_{t+1} K_{t+1} B_t) u_t = -(B'_{t+1} A_t x_t + F'_{t} \lambda_t + B'_{t+1} D'_{t+1} \lambda_{t+1,T}).
\]
The matrix multiplying \( u_t \) on the left is positive definite since \( R_t \) is positive definite and \( B_t'K_{t+1}B_t \) is positive semidefinite. It follows that the minimizing control of this Lagrangian problem is

\[
\begin{align*}
  \mathbf{u}_t^* &= -(R_t + B_t'K_{t+1}B_t)^{-1}(B_t'K_{t+1}A_tx_t + F_t'\mathbf{\lambda}_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T}).
\end{align*}
\]

By substitution into the expression for \( L_t \), we have

\[
\begin{align*}
  L_t(x_t, \mathbf{\lambda}_{t,T}) &= x_t'Q_xt_t + x_t'A_t'K_{t+1}A_tx_t + \sum_{s=t}^{T-1} \mathbb{E}[w_s'K_{s+1}w_s] \\
  &\quad - \mathbf{\lambda}_{t+1,T}'H_{t+1,T} + 2x_t'A_t'D_{t+1}\mathbf{\lambda}_{t+1,T} - 2\mathbf{\lambda}_{t+1,T}'G_{t+1} \\
  &\quad -(B_t'K_{t+1}A_tx_t + F_t'\mathbf{\lambda}_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T})'M_t \\
  &\quad (B_t'K_{t+1}A_tx_t + F_t'\mathbf{\lambda}_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T}) \\
  &= x_t'K_{t+1}x_t + \sum_{s=t}^{T-1} \mathbb{E}[w_s'K_{s+1}w_s] - \mathbf{\lambda}_{t,T}'H_{t,T}\mathbf{\lambda}_{t,T} + 2(D_{t+1}x_t - G_t)'\mathbf{\lambda}_t.
\end{align*}
\]

We have established the positive-definiteness of \( M_t \). The positive-definiteness of \( K_t \) clearly follows by induction. For the positive semidefiniteness of \( H_t \), note that \( H_{T-1} \) is positive semidefinite. Suppose \( H_{t+1} \) is positive semidefinite and consider pre and post multiplication of \( H_t \) by vector \( z_{t,T} \) of appropriate dimension.

\[
\begin{align*}
  z_{t,T}'H_{t,T}z_{t,T} &= z_{t,T}'F_tM_tF_t'z_t + z_{t+1,T}'D_{t+1}B_tM_tF_t'z_t \\
  &\quad + z_{t+1,T}'D_{t+1}B_tM_tB_t'D_{t+1}\mathbf{\lambda}_{t+1,T} + z_{t+1,T}'H_{t+1}\mathbf{\lambda}_{t+1,T} \\
  &\quad + z_{t+1,T}'D_{t+1}B_tM_tB_t'D_{t+1}\mathbf{\lambda}_{t+1,T} \\
  &= z_{t,T}'F_tM_tF_t'z_t + z_{t+1,T}'H_{t+1}\mathbf{\lambda}_{t+1,T} \\
  &\quad + z_{t+1,T}'D_{t+1}B_tM_t(F_t'z_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T}) \\
  &\quad + z_{t+1,T}'D_{t+1}B_tM_t(F_t'z_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T}) \\
  &= z_{t,T}'F_tM_tF_t'z_t + z_{t+1,T}'H_{t+1}\mathbf{\lambda}_{t+1,T} \\
  &\quad +(F_t'z_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T})'M_t(F_t'z_t + B_t'D_{t+1}\mathbf{\lambda}_{t+1,T}).
\end{align*}
\]
It follows that $H_t$ is positive semidefinite.  ■

We define

$$L_t(x_t) = \max_{\lambda_t, \tau \geq 0} L_t(x_t; \lambda_{t, \tau}).$$

**Theorem 12** (a) $L_t(x_t) \leq J_t(x_t)$,

(b) $L_t(x_t; \lambda_{t, \tau})$ is quadratic in $\lambda_{t, \tau}$.

**Proof.** For Part (a), note that $L_T(x_T) = J_T(T)$. Assume the result is true for $t + 1$. For a given state $x_t$, let $u_t^*$ be the argmin of $J_t(x_t)$. Note that for any $\lambda_{t, T} \geq 0$

$$L_t(x_t; \lambda_{t, T}) = \min_{u_t \in U_t} \{ x_t' Q_t x_t + u_t' R_t u_t + 2\lambda_t' (F_t u_t - g_t) + E[L_{t+1}(x_{t+1}; \lambda_{t+1, T})|x_t, u_t] \}$$

$$\leq \min_{u_t \in U_t} \{ x_t' Q_t x_t + u_t' R_t u_t + 2\lambda_t' (F_t u_t - g_t) + E[J_{t+1}(x_{t+1})|x_t, u_t] \}$$

$$\leq x_t' Q_t x_t + u_t' R_t u_t + 2\lambda_t' (F_t u_t^* - g_t) + E[J_{t+1}(x_{t+1})|x_t, u_t^*].$$

$$\leq J_t(x_t). \quad (4.2)$$

Eq. (4.2) follows since $F_t u_t^* \leq g_t$ and $\lambda_t \geq 0$. The result follows. Part (b), follows from the positive semidefiniteness of $-H_t$.  ■

**4.2.1 Analytical comparison of bounds**

In this section we compare three bounds for estimating the true cost to go. Note that $L_0(x_0; \lambda_{0, T})$ comprises deterministic components and a random component, independent on one another. Suppose the problem is deterministic and $w_t = 0$ for all $t$. We define

$$J_0^D(x_0) = \min_{u_0, \ldots, u_{T-1}} x_T' Q_T x_T + \sum_{t=0}^{T-1} (x_t' Q_t x_t + u_t' R_t u_t),$$

$$109$$
with the same linear dynamics and constraints. For given \( \lambda_{0,T} \) we define the optimal bound achieved by the Lagrangian problem as

\[
L_0^D(x_0; \lambda_{0,T}) = x_0'K_{T+1}x_T - \lambda_{T,T}'H_t\lambda_{T,T} + 2(D_t x_t - G_t)'\lambda_{T,T},
\]

and \( L_0^D(x_0) = \max_{\lambda_{0,T}} L_0^D(x_0; \lambda_{0,T}) \). In Appendix B we show that \( J_0^D(x_0) \) is the solution to a quadratic minimization problem in the controls \( u_0, \ldots, u_T \). If follows that for the deterministic case strong duality holds and \( L_0^D(x_0) = J_0^D(x_0) \) (see Bertsekas [8]). Then, for general \( w_t \) with zero mean and finite second moment we have the relation

\[
L_0^D(x_0) = J_0^D(x_0) + \sum_{t=0}^{T-1} \mathbb{E}[w_t'K_{t+1}w_t].
\]

Let

\[
J_0^{NC}(x_0) = \min_{\pi} \mathbb{E}_{\pi} \left[ x_T'Q_T x_T + \sum_{t=0}^{T-1} (x_t'Q_t x_t + u_t'R_t u_t) \right],
\]

the unconstrained version of the problem defined in Section 4.1. Bertsekas [7] shows that

\[
J_0^{NC}(x_0) = x_0'K_0 x_0 + \sum_{t=0}^{T-1} \mathbb{E}[w_t'K_{t+1}w_t],
\]

where \( K_0, \ldots, K_T \) are defined as above. It can be seen that the additional maximization over \( \lambda_{0,T} \) in solving Eq. 4.1 is an approximation at the additional cost incurred by having constraints. We have the following lemma.

**Lemma 6** We have that

\[
L_0(x_0) = J_0^D(x_0) + J_0^{NC}(x_0) - x_0'K_0 x_0.
\]

The significance of this result is that it tells us the added value of the bound we achieved by using Lagrangian relaxation over simply ignoring the constraints. That amount is \( J_0^D(x_0) - x_0'K_0 x_0 \).
4.2.2 Partial decomposition

The drawback of the previous section is that for large $T$, the resulting quadratic maximization problem is very difficult to solve. One approximation that can be made is to set $\lambda_0 = \cdots = \lambda_{T-1}$. The Lagrangian relaxed problem is

$$L_t(x_t; \lambda) = \min_{u_t} \left\{ \mathbb{E}[x_t'Q_t x_t + u_t'R_t u_t + L_{t+1}(A_t x_t + B_t u_t + w_t, \lambda)] + 2\lambda'(F_t u_t - g_t) \right\},$$

To denote the Lagrangian relaxed problem with the multipliers specified as mentioned above. The following result is obtained.

**Theorem 13** We have that

$$L_0(x_0; \lambda) = \sum_{t=0}^{T-1} \mathbb{E}[w_t'K_{t+1} w_t] + x_0'K_0 x_0 - \lambda'w_0 \lambda + 2(S_0 x_0 - \sum_{t=0}^{T-1} g_t)'\lambda,$$

where

$$K_T = Q_T,$$

$$K_t = A_t'(K_{t+1} - K_{t+1}B_tM_tB_t'K_{t+1})A_t + Q_t,$$

$$S_T = 0,$$

$$S_t = (S_{t+1} - (F_t + S_{t+1}B_t)M_tB_t'K_{t+1})A_t,$$

$$V_T = 0,$$

$$V_t = V_{t+1} + (F_t + S_{t+1}B_t)M_t(F_t + S_{t+1}B_t)'$$

$$M_t = (R_t + B_t'K_{t+1}B_t)^{-1}.$$

Furthermore, $V_t$ are symmetric positive semidefinite, and $V_t$ are symmetric positive definite.

The proof is similar to that of Theorem 11 and can be found in Appendix C.

Define $\bar{L}_0(x_0) = \max_{\lambda \geq 0} L_0(x_0, \lambda)$. It follows that $\bar{L}_0(x_0) \leq L_0(x_0) \leq J_0(x_0)$.
Extension

In deriving this relaxed problem, the assumption $\lambda_1 = \cdots = \lambda_{T-1}$ has resulted in an easy minimization (and maximization) problem. However, we have placed equal weight on constraints of each stage. In this extension we select weighted penalizers. Let $\alpha = (\alpha_0, \ldots, \alpha_{T-1})$. We define

$$ L_0(x_0; \lambda, \alpha) = \sum_{t=0}^{T-1} E[w_t^i K_{t+1} w_t] + x_0^i K_0 x_0 - \lambda^i V_0 \lambda + 2(S_0 x_0 - \sum_{t=0}^{T-1} \alpha_t g_t)^i \lambda, $$

where

$$ K_T = Q_T $$
$$ K_t = A_t'(K_{t+1} - K_{t+1} B_t M_t B_t' K_{t+1}) A_t + Q_t $$
$$ S_T = 0 $$
$$ S_t = (S_{t+1} - (\alpha_t F_t + S_{t+1} B_t) M_t B_t' K_{t+1}) A_t $$
$$ V_T = 0 $$
$$ V_t = V_{t+1} + (\alpha_t F_t + S_{t+1} B_t) M_t (\alpha_t F_t + S_{t+1} B_t)' $$
$$ M_t = (R_t + B_t' K_{t+1} B_t)^{-1}. $$

We set $\alpha_0 = 1$. Given $\alpha$, the aim is to solve $\max_{\lambda \geq 0} L_0(x_0; \lambda, \alpha)$.

Define $\tilde{L}_0^\alpha(x_0) = \max_{\alpha \geq 0} \max_{\lambda \geq 0} L_0(x_0, \alpha \lambda)$. It follows that $\tilde{L}_0(x_0) \leq \tilde{L}_0^\alpha(x_0) \leq L_0(x_0) \leq J_0(x_0)$.

4.3 A Lagrange-based policy

In this section we develop a Lagrange based policy for controls $u_t$ at time $t$. The approach we take is similar to the approach taken in Section 2.4.1. In particular, we approximate the future cost to go by relaxing future constraints with Lagrange multipliers. This enables the controller to have an easily calculated estimate of the cost to go given a feasible decision in the current period.
We motivate our method by considering the exact solution of what action to take at time $t$ as the solution to

\[(DP) \quad J_t(x_t) = \max \left\{ u_t^R_t u_t + E[J_{t+1}(A_t x_t + B_t u_t + w_t)] \mid F_t u_t \leq g_t \right\}.\]

As the calculation of $J_{t+1}(x_{t+1})$ is intractable, we would like to approximate it by some other means. One method would be to use $L_{t+1}(x_{t+1})$. This approach would require determining a new $\lambda^*_{t+1,T}$ for every occurrence of $x_{t+1}$. However, because $u_t$ is continuous, one can not exhaustively solve to find $L_{t+1}(x_{t+1})$ for every control $u_t$. Instead, we use the approximation

$$L_{t+1}(x_{t+1}) \approx L_{t+1}(x_{t+1}; \bar{\lambda}_{t+1,T}),$$

where $(\lambda'_t, \bar{\lambda}'_{t+1,T}) = \lambda^*_{t,T} = \lambda^*_{t,T}'. $ That is, instead of minimizing $L_{t+1}(x_{t+1}; \lambda_{t+1,T})$ over $\lambda_{t+1,T}$, we use the Lagrange multipliers found at time $t$ to estimate $L_{t+1}(x_{t+1})$. The same approach is taken in Section 2.4.3.

$$E[L_{t+1}(x_{t+1}; \lambda_{t+1,T})|x_t, u_t] =$$

$$x_t^A_t K_{t+1} A_t x_t + \sum_{s=t}^{T-1} E[w_s' K_{s+1} w_s] + u_t^B_t K_{t+1} B_t u_t$$

$$+ 2x_t^A_t K_{t+1} B_t u_t + 2\lambda_{t+1,T}' D_{t+1} B_t u_t$$

$$- \bar{\lambda}'_{t+1,T} H_{t+1} \lambda_{t+1,T} + 2x_t^A_t D_{t+1} \lambda_{t+1} - 2\lambda_{t+1,T}' G_{t+1}.$$

Then, given $\bar{\lambda}_{t,T}$, we solve

$$\min \quad x_t^Q_t x_t + u_t^R_t u_t + x_t^A_t K_{t+1} A_t x_t + \sum_{s=t}^{T-1} E[w_s' K_{s+1} w_s]$$

$$+ u_t^B_t K_{t+1} B_t u_t + 2x_t^A_t K_{t+1} B_t u_t + 2\lambda_{t+1,T}' D_{t+1} B_t u_t$$

$$- \bar{\lambda}'_{t+1,T} H_{t+1} \bar{\lambda}_{t+1,T} + 2x_t^A_t D_{t+1} \bar{\lambda}_{t+1,T} - 2\lambda_{t+1,T}' G_{t+1}$$

s.t. $F_t u_t \leq g_t$. 

113
We can remove the extraneous terms and solve

\[
(\text{DP}_A) \quad \min u_t'(R_t + B_t'K_{t+1}B_t)u_t + 2(B_t'K_{t+1}A_t x_t + B_t'D_{t+1}\lambda_{t+1,T})'u_t \\
\text{s.t. } F_t u_t \leq g_t.
\]

The policy (DP\(_A\)) refers to the decision \(u_t\) selected by the above quadratic program.

In the next two sections we present two heuristics that will serve as benchmarks for the Lagrange-based policy above.

### 4.3.1 Greedy heuristic

This greedy-type heuristic considers only immediate costs. Effectively, it assumes \(E[J(x_{t-1})|x_t, u_t] = 0\). The decision taken at time \(t\) is the solution to the following dynamic program.

\[
(\text{DP}_{\text{Greedy}}) \quad \min u_t'R_tu_t \\
\text{s.t. } F_t u_t \leq g_t.
\]

### 4.3.2 Deterministic approximation heuristic

The second heuristic assumes the problem is deterministic. This method is used to find \(J_0^D(x_0)\) and its full formulation is derived in Appendix B.

\[
(\text{DP}_{\text{Greedy}}) \quad J_0^D(x_0) = \\
\min_{u_t, \ldots, u_{t-1}} \quad x_t'Q_t x_t + \sum_{s=t+1}^{T} \left( \sum_{n=t}^{s-1} D_{s-1,n+1}B_n u_n + D_{s-1,t} x_t \right)'Q_s \\
\quad \cdot \left( \sum_{m=t}^{s-1} D_{s-1,m+1}B_m u_m + D_{s-1,t} x_t \right) \\
\text{s.t. } F_s u_s \leq g_s, \ s = t, \ldots, T - 1.
\]

114
4.4 Numerical results

In this section we test the performance our Lagrange based bounds against other bounds. In addition, we test our Lagrange-based policy against two other heuristics. For the remainder of this section we define the following for notational convenience:

\[
Z_D = J^D_0(x_0) \\
Z_{NC} = J^{NC}_0(x_0) \\
Z_A = L_0(x_0) \\
Z_{\lambda} = \bar{L}_0(x_0).
\]

4.4.1 Comparison of bounds

Matrices \( A_t, B_t, F_t \) and vector \( g_t \) are randomly drawn from a Uniform(-5,5) distribution. A quick test ensures the constraints permit a feasible solution. Matrices \( R_t \) and \( Q_t \) are random positive definite matrices, scaled by a factor of 10. Vectors \( w_t \) are drawn from a Uniform(0,1) distribution, then rescaled for a zero mean and are then rescaled by a factor of 10. \( P_t \) are Uniform(0,1). The starting \( x_0 \) was drawn from Uniform(-5,5) \( n \) dimensional distribution. Some sample results are shown in Table 4.1. Column \(|w|\) represents the number of possible random scenarios. We also simulated the results over a 1000 instances of the problem from random starting points. The mean ratio of \( Z_A / \max \{ Z_D, Z_{NC} \} \) is 1.063 with a 95% confidence interval of (1.052, 1.074). We also fixed the starting point to \( x_0 = (0, \ldots, 0) \). In this case the average ratio was 1.139 with a 95% confidence interval of (1.124,1.154).

4.4.2 Comparison of heuristics

We test our heuristic, DP\(A\) against two other benchmarks explained in Section 4.3. As in the previous section, matrices \( A_t, B_t, F_t \) and vector \( g_t \) are drawn from randomly from a Uniform(-5,5) distribution. Matrices \( R_t \) and \( Q_t \) are random positive definite matrices, scaled by a factor of 10. Vectors \( w_t \) are drawn from a Uniform(0,1) distribution, then rescaled to have a zero mean and are rescaled again by a factor of 10.
\[ (|x|, |u|, |g|, w, T) \] | \( Z_{NC} \) | \( Z_\lambda \) | \( Z_D \) | \( Z_\lambda \) | \( Z_\lambda / \max\{Z_D, Z_{NC}\} \) 
\hline
(3,3,5,3,5) | 1531 | 1548 | 2879 | 4409 | 1.53 \\
(3,3,5,3,5) | 1083 | 1113 | 41061 | 42145 | 1.03 \\
(3,3,5,3,5) | 694 | 694 | 855 | 1549 | 1.81 \\
(5,5,7,2,10) | 7107 | 7107 | 1546 | 8653 | 1.22 \\
(5,5,7,2,10) | 22845 | 22846 | 1498 | 24344 | 1.07 \\
(5,5,7,2,10) | 7251 | 7252 | 14269 | 21628 | 1.52 \\
(7,5,5,10,5) | 14367 | 14379 | 97759886 | 97774253 | 1.00 \\
(7,5,5,10,5) | 5368 | 5372 | 542846 | 548213 | 1.01 \\
(7,5,5,10,5) | 6868 | 6871 | 39895205 | 44311849 | 1.11 \\
\hline

Table 4.1: Numerical results of bounds for constrained quadratic-cost linear systems.

\( P_t \) are Uniform(0,1). The starting \( x_0 \) was drawn from Uniform(-5,5) \( n \) dimensional distribution. The heuristics implemented follow.

\( Z_\text{Greedy} \): Estimated (through simulations) expected value of the greedy heuristic (DP\text{Greedy}). At each time ignores future costs subject to \( F_t u_t \leq g_t \).

\( Z_\text{Greedy}_d \): Estimated (through simulations) expected value of the greedy heuristic which at each time assumes the problem is deterministic. The control used is the solution to (DP\text{Greedy}_d).

\( Z_{DP\lambda} \): The estimated expected value of the heuristic that uses the solution to (DP\lambda) detailed in Section 4.3.

Each heuristic was simulated 5000 times. The results are shown in Table 4.2. 95% confidence intervals are shown in italics.

### 4.4.3 Summary

The results of the experiments are summarized below.

1. Table 4.1 shows that \( Z_\lambda \) provides the best bounds for long horizon problems, relative to the other closed form bounds.

2. \( Z_\lambda \) is only marginally better than \( Z_D \). This is because the quadratic maximization problem had relatively small optimal value.
| (|x|, |u|, |g|, |w|, T) | \(Z_A\) | \(Z_{\text{Greedy}}\) | \(Z_{\text{Greedy}^\text{Dyn}}\) | \(Z_{\text{DPA}}\) |
|---|---|---|---|---|
| (5,5,8,2,3) | 2.9219e+08 | 5.2680e+08 | 3.8198e+08 | 5.3457e+08 |
| | | (5.2056, 5.3395) | (3.7839, 3.8558) | (5.2837, 5.4078) |
| (4,4,6,2,5) | 7.8498e+07 | 9.0849e+07 | 8.5201e+07 | 8.5135e+07 |
| (2,2,5,10,5) | 2.4260e+05 | 1.4676e+05 | 1.439e+05 | 1.4158e+05 |
| | | (1.5079, 1.5482) | (1.4791, 1.5185) | (1.4553, 1.4947) |
| (3,3,4,10,5) | 4.1840e+11 | 2.8496e+12 | 3.3427e+12 | 2.7090e+12 |
| | | (2.8326, 2.8667) | (3.3222, 3.3631) | (2.6941, 2.7239) |
| (2,2,5,10,5) | 1.6980e+06 | 1.9991e+07 | 1.8576e+07 | 1.3469e+07 |
| | | (2.0534, 2.1076) | (1.4111, 1.4752) | (1.9159, 1.9742) |
| (4,4,6,2,4) | 5.1330e+07 | 1.8090e+08 | 1.797e+08 | 1.3654e+08 |
| | | (1.8160, 1.8343) | (1.8277, 1.8463) | (1.3791, 1.3987) |

Table 4.2: Numerical results of heuristics for constrained quadratic-cost linear systems. Italicized numbers in parenthesis are 95% confidence intervals of the mean.

3. Table 4.2 shows that \(Z_{\text{DPA}}\) outperforms both greedy heuristics.

4. The difference between \(Z_A\) and the best of the three heuristics is usually quite large. Either the heuristics are weak or the bound needs improvement.

### 4.5 Conclusions

In this chapter we have investigated the role of Lagrangian relaxation in problems other than weakly coupled dynamic optimization problems. The problems investigated are quadratic cost linear systems with linear constraints. We have shown:

(i) Lagrange relaxation of the constraints yields a bound on the cost to go that is quadratic in the multipliers; (ii) The Lagrangian bound is closely related to the bound achieved by assuming no constraints, and to the bound achieved by assuming no randomness; (iii) The Lagrange-based policies that developed in Chapter 2 apply and produce good results. The key result of this chapter is that we have shown that our Lagrange-based bounds and policies apply to classes of problems other than weakly coupled dynamic optimization problems. This result establishes that future research in Lagrange-based dynamic policies can lead to broader applicability.
Chapter 5

Optimal bidding in on-line auctions

5.1 Introduction

On-line auctions have become established as a convenient, efficient, and effective method of buying and selling merchandise. The largest of the consumer-to-consumer on-line auction web sites is eBay which has over 42 million registered users and was the host of over $9.3 billion worth of goods sold\textsuperscript{1} in over 18,000 categories, ranging from consumer electronics and collectibles to real estate and cars. Because of the ease of use, the excitement of participating in an auction, and the chance of winning the desired item at a low price, the auctions hosted by eBay attract a wide variety of bidders in terms of experience and knowledge concerning the item for auction. Indeed, even for standard items like personal digital assistants we have observed a large variance in the selling price, which illustrates the uncertainty one faces when bidding.

In this chapter we construct a dynamic model of on-line auctions that can be constructed using publicly available data. In the context of this model we use dynamic optimization to determine an optimal bidding policy for a bidder with a set budget. We extend these results to find the optimal policy for a bidder bidding in multiple simultaneous on-line auctions with a fixed budget. Several approximation schemes are implemented to solve this high dimensional problem, including a variation of our

\textsuperscript{1}http://www.shareholder.com/ebay/annual/2001_annual_10K.pdf
Lagrangian heuristic.

5.1.1 Overview of on-line auctions

eBay auctions have a finite duration (3, 5, 7, or 10 days). The data available to bidders during the duration of the auction include: the item’s description, the number of bids, the ID of all the bidders and the time of their bid, but not the amount of their bid (this becomes available after the auction has ended), the ID of the current highest bidder, the time remaining until the end of the auction, whether or not the reserve price has been met, the starting price of the auction, and the second highest price of the item, referred to as the listed price. The auction ends when time has expired, and the item goes to the highest bidder at a price equal to a small increment above the second highest bid.

eBay publishes on the web the bidding history of all of the auctions completed through its website from the past thirty days. The bidding history includes the starting and ending time of the auction, the amount of the minimum opening bid set by the seller, the price for which the item was sold and, apart from the winning bid of the auction, the amount of every bid, and when and by whom it was submitted. For the winning bid of the auction only the identity of the bidder and submission date are revealed. In addition, if the auction was a reserve auction, then an indication of whether or not the reserve price was met. However, eBay does not publish the reserve price set by the sellers, and without this information we felt we could not properly model reserve price auctions. As a result we only consider auctions without a reserve price.

5.1.2 Literature review

The literature for traditional auctions is extensive. For a survey of auction theory see Klemperer [26], Milgrom [32], and McAfee and McMillan [30]. The mechanism for determining a winner in an eBay auction is similar to that of a second-price sealed bid auction, also known as a Vickrey auction, see Vickrey [47]. In such auctions
the optimal action, regardless of what the opponents are doing, is at some point to submit a bid equal to one’s valuation of the item. The primary difference between a Vickrey auction and an eBay auction is that eBay reveals the identity of bidders and the value of the highest bid to date. In addition, the end of an auction on eBay is fixed in advance (i.e., there is a hard stop time). This makes it possible for bidders to submit bids close enough to the ending time of the auction and as a result, to not allow for competitors to respond. Such a strategy, known as sniping, has become so popular that a number of web sites exist to assist bidders in sniping (for example, see www.esnipe.com). In fact, we have found that for a personal digital assistant, model Palm Pilot III, the bids received per second in the final ten seconds is over 100 times greater than those received in the final day.

On-line auctions allow bidders to participate in many auctions at once, or perhaps in many auctions in a short time span. The popularity of on-line auctions has motivated both theoretical and empirical investigations of bidding strategies. Taking into account network congestion, response time, and potentially other factors, Roth and Ockenfels [40] (see also Ockenfels and Roth [35]) provide evidence that there is a small but significant probability that a bid placed at the last seconds of an auction will not register on eBay’s web site. This is an effect that our proposed algorithm explicitly accounts for. Roth and Ockenfels [40] show that if one is not certain that a submitted bid will be accepted, then there is no dominate bidding strategy. Furthermore, they argue that it is an undominated strategy to submit multiple bids. Bajari and Hortacsu [3] show that in a common value environment, sniping is an equilibrium behavior. Late bidding in on-line auctions has attracted a lot of interest from both practitioners and academics. Landsburg [27], suggests submitting bids late and bidding multiple times in order to keep others from learning and out-bidding him. Hahn [23] provides evidence that late bidding makes up 45% of all bids, but also that there is also a substantial amount of early bidding. Nonetheless, Hasker et al. [24] statistically reject that bidders commonly use a “Jump-call” strategy (a derivative of “Jump-bidding” from Avery [1] for English auctions), but also a “Snipe-or-war” strategy. Mizuta and Steiglitz [33] simulate a bidding a bidding environment with
early bidders and snipers and find out that early bidders win at a lower price but win fewer times on average. There is also evidence that bidders react to the ratings of sellers (see Lucking-Reiley et al. [28] and Dewan and Hsu [19]). In this chapter we ignore this effect.

There has also been work done on bidding in multiple auctions. Oren and Rothkopf [36] consider the effects of bidding in sequential auctions against intelligent competitors and derive an infinite horizon optimal bidding strategy. Boutillier et al. [14] develop a piece-wise linear dynamic programming approximation scheme for bidding in multiple sequential auctions with complementaries and substitutability. Zheng [54] finds empirical evidence from eBay that bidders bid across multiple auctions simultaneously and that they tend to bid for the item with the lowest listed price. They also show that such a bidding strategy is a Nash equilibrium and results in lower payments for winners. Stone and Greenwald [42] consider a number of automated trading agents programmed to bid in multiple simultaneous auctions for complementary and substitutable goods. Bapna et al. [4] provide an empirical and theoretical study of observed bidding strategies in on-line auctions with multiple identical items.

5.1.3 Philosophy and contributions

Our objective in this chapter is to construct algorithms that determine the optimal bidding policy for a given utility function for a single item in an on-line auction, as well as multiple items in multiple simultaneous or overlapping on-line auctions. In order to explain our modeling choices (see Section 5.2 for more details), we require that the model we build for optimal bidding for a potential buyer, called the agent throughout the chapter, satisfies the following requirements:

(a) It captures the essential characteristics of on-line auctions.

(b) It leads to a computationally feasible algorithm that is directly usable by bidders.

(c) The parameters for the model can be estimated from publicly available data.

To achieve our goals we have taken an optimization, as opposed to a game theoretic approach. The major reason for this is the requirement of having a computationally
feasible algorithm that is based on data and is directly applicable by bidders. Furthermore, our goal is to impose as few behavioral assumptions as possible and yet come up with bidding strategies that work well in practice (see also Sections 5.2.9, 5.2.10, 5.3.6 for some empirical evidence). Given that auctions evolve dynamically, in this chapter we adopt a dynamic programming framework. We model the rest of the bidders as generating bids from a probability distribution which is dependent on the time remaining in the auction and the listed price, and can be directly estimated using publicly available data. Furthermore, we have tested our approach in a setting where there is a population and an additional competitor. We intend to show that by incorporating other strategies into the population bidding distribution (i.e., the agent is aware that the population may be also using "smarter" strategies) the approach suggested in this chapter performs better when competing against other strategies. Finally, the first author has applied the algorithms in this chapter many times in a real world setting to buy stamps and collections of stamps. The author's findings are that the algorithm is highly effective in that it both increases the chances of winning and decreases the amount paid per win. As a result, we feel that a dynamic programming approach gives rise to practical, realistic and directly applicable bidding strategies.

We feel that this chapter makes the following contributions:

1. We propose a model for on-line auctions that satisfies requirements (a)-(c), mentioned above. The model gives rise to an exact optimal algorithm for a single auction based on dynamic programming.

2. We show in simulation using real data from 1772 completed auctions for personal digital assistants and 4208 completed auctions for stamp collections that the proposed algorithm outperforms simple, but widely used static heuristic rules.

3. We extend our methods to multiple simultaneous or overlapping on-line auctions. We provide five approximate algorithms, based on approximate dynamic programming and integer programming. The strongest of these methods is based on combining the value functions of single auctions found by dynamic programming using an integer programming framework. We provide computational evi-
vidence that the method produces high quality solutions fast and reliably. To the best of our knowledge, this method is new and may have wider applicability to high dimensional dynamic programming problems.

4. We test our algorithm in a multi-bidder environment against widely used bidding heuristics for both single and multiple simultaneous auctions. We show how our algorithms can be improved by incorporating different bidding strategies into the probability distribution of the competing bids.

5.1.4 Structure of the chapter

The chapter is structured as follows. In Section 5.2, we present our formulation and algorithm for a single item on-line auction. In Section 5.3, we present several algorithms based on approximate dynamic programming and integer programming for the problem of optimally bidding on multiple simultaneous auctions, and in Section 5.4, we consider multiple overlapping on-line auctions. The final section summarizes our contributions.

5.2 Single item auction

In this section, we outline the model in Section 5.2.1, the process we used to estimate the parameters of the model in Section 5.2.8, and the empirical results from the application of the proposed algorithm in Section 5.2.9.

5.2.1 The model

The length of the auction is discretized into $T$ periods during which bids are submitted and where the winner, the highest bidder, is declared in period $T + 1$. As the majority of the activity in an eBay auction occurs near the end of the auction, see Section 5.2.8 and [35], we have used the following $T = 13$ periods to indicate the time remaining in the auction: 5 days, 4 days, 3 days, 2 days, 1 day, 12 hours, 6 hours, 1 hour, 10 minutes, 2 minutes, 1 minute, 30 seconds, and 10 seconds remaining in the auction.
These periods are indexed by \( t = 1, \ldots, 13 \) respectively. These time intervals were selected for two reasons: First, they were chosen in decreasing size in order to match the increasing intensity of intensity of bids as the auction draws to a close. Second, the periods were chosen at times which are naturally convenient for bidders to follow.

### 5.2.2 State

A key modeling decision is the description of the state. We define the state to be \((x_t, h_t)\) for \( t = 1, \ldots, T + 1 \), where

\[
x_t = \text{listed price at time } t,
\]

\[
h_t = \begin{cases} 
\text{the agent's proxy bid if the highest bidder at time } t, \\
0, \text{ otherwise.}
\end{cases}
\]

We will often use the indicator \( w_t = 1 \) if \( h_t > 0 \), and zero, otherwise, to indicate if the agent is the highest bidder or not.

### 5.2.3 Control

The control at time \( t \) is the amount \( u_t \) the agent bids. We assume that the agent has a maximum price \( A \) up to which he is willing to bid for. Clearly, \( u_t \in F_t = \{0\} \cup \{u_t | x_t \leq u_t \leq A\} \) if \( w_t = 0 \), and \( u_t \in F_t = \{u_t | h_t \leq u_t \leq A\} \) if \( w_t = 1 \).

### 5.2.4 Randomness

There are three elements of randomness in the model:

(a) How the other bidders (the population) will react. In order to model the population's behavior, we let \( q_t \) be the population's bid. Note that \( q_t = 0 \) means that the population does not submit a bid at time \( t \). We assume that \( P(q_t = j|x_t, h_t) \) is known and estimated from available data, as described in Section 5.2.8.

(b) The proxy bid \( \overline{h}_t \) at time \( t \) which is the highest bid to date if \( w_t = 0 \) (the agent is not the highest bidder). If, however, \( w_t = 1 \), then \( \overline{h}_t \) is defined to be zero.
The reason for this is that in this case, the proxy bid is known to the agent and is part of the state (denoted at $h_t$). In an eBay auction bidders know the listed price, but not the value of the proxy bid, unless of course they are the highest bidder. If a submitted bid is higher than the proxy bid, then the new listed price becomes equal to the old proxy bid plus a small increment. The exception to this is if a bidder out-bids his own proxy bid, in which case the listed price remains unchanged. For a given listed price, the minimum allowable bid is a small increment above the current listed price. We assume that if the agent is not the highest bidder, then the distribution of the proxy bid $P(\tilde{h}_t = j|x_t, h_t = 0)$ is known and estimated from available data, as described in Section 5.2.8.

(c) Whether or not the bid will be accepted. As we have mentioned, near the last seconds in the auction, that is for $t = T$, there is evidence (see [35]) that a bid will be accepted with probability $p < 1$. This models increased congestion due to increased activity, low speed connections, network failures, etc. In all other times $t = 1, \ldots, T - 1$ the bid will be accepted. We use the random variable $v_t$, which is equal to one if the bid is accepted, and zero, otherwise. From the previous discussion, $P(v_t = 1) = 1$, for $t = 1, \ldots, T - 1$, and $P(v_T = 1) = p$.

5.2.5 Dynamics

The dynamics of the model are of the type

$$x_{t+1} = f(x_t, h_t, u_t, v_t, q_t, \tilde{h}_t)$$
$$h_{t+1} = g(h_t, u_t, v_t, q_t, \tilde{h}_t),$$
where the functions \( f(\cdot) \), \( g(\cdot) \) are as follows:

\[
\begin{align*}
    w_t = 0, & \quad q_t \geq u_t \geq \overline{h}_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = u_t, \quad h_{t+1} = 0, \quad (5.1)\\
    w_t = 0, & \quad q_t \geq \overline{h}_t \geq u_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \overline{h}_t, \quad h_{t+1} = 0, \quad (5.2)\\
    w_t = 0, & \quad \overline{h}_t \geq q_t \geq u_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \max(q_t, x_t), \quad h_{t+1} = 0, \quad (5.3)\\
    w_t = 0, & \quad u_t > q_t \geq \overline{h}_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = q_t, \quad h_{t+1} = u_t \quad (5.4)\\
    w_t = 0, & \quad u_t > \overline{h}_t \geq q_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \overline{h}_t, \quad h_{t+1} = u_t \quad (5.5)\\
    w_t = 0, & \quad \overline{h}_t \geq u_t \geq q_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \max(u_t, x_t), \quad h_{t+1} = 0, \quad (5.6)\\
    w_t = 0, & \quad q_t \geq \overline{h}_t, \quad v_t = 0 \quad \Rightarrow \quad x_{t+1} = \overline{h}_t, \quad h_{t+1} = 0, \quad (5.7)\\
    w_t = 0, & \quad \overline{h}_t \geq q_t, \quad v_t = 0 \quad \Rightarrow \quad x_{t+1} = \max(q_t, x_t), \quad h_{t+1} = 0, \quad (5.8)\\
    w_t = 1, & \quad q_t \geq u_t \geq \overline{h}_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = u_t, \quad h_{t+1} = 0, \quad (5.9)\\
    w_t = 1, & \quad u_t > q_t \geq \overline{h}_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = q_t, \quad h_{t+1} = u_t \quad (5.10)\\
    w_t = 1, & \quad u_t \geq \overline{h}_t > q_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \max(q_t, x_t), \quad h_{t+1} = u_t \quad (5.11)\\
    w_t = 1, & \quad u_t > \overline{h}_t \geq q_t, \quad v_t = 1 \quad \Rightarrow \quad x_{t+1} = \max(q_t, x_t), \quad h_{t+1} = u_t \quad (5.12)\\
    w_t = 1, & \quad q_t \geq \overline{h}_t, \quad v_t = 0 \quad \Rightarrow \quad x_{t+1} = \overline{h}_t, \quad h_{t+1} = 0, \quad (5.13)\\
    w_t = 1, & \quad h_t > q_t, \quad v_t = 0 \quad \Rightarrow \quad x_{t+1} = \max(q_t, x_t), \quad h_{t+1} = h_t \quad (5.14)
\end{align*}
\]

Eqs. (5.1)-(5.8) are for the case when \( w_t = 0 \). Eqs. (5.1)-(5.3) address the case that the population's bid is higher than the agent's bid, and the agent's bid is accepted. In Eq. (5.1), both the population and the agent bid above the proxy bid at time \( t \), and thus the next listed price is \( u_t \), and the agent is not the highest bidder. In Eq. (5.2) the highest price at time \( t \) is between the population's and the agent's bid, and thus the next listed price will be \( h_t \), and the agent is not the highest bidder. In Eq. (5.3) both the population and the agent bid lower than the proxy bid at time \( t \), and thus the next listed price is \( q_t \), and the agent is not the highest bidder. Note that the max operator in Eqs. (5.3) and (5.6) cover the case that neither the population nor the agent bids (\( q_t = u_t = 0 \)).

Eqs. (5.4)-(5.6) address the case that the population's bid is lower than the agent's
bid, and the agent’s bid is accepted, analogously to Eqs. (5.1)-(5.3). Finally, Eqs. (5.7), (5.8) cover the case that the agent’s bid is not accepted. Note that the max operator in Eq. (5.8) covers the case that the population does not bid \( q_t = 0 \).

Eqs. (5.9)-(5.14) address the case that \( w_t = 1 \), that is the agent is the highest bidder and hence has a proxy bid. In Eq. (5.9) both the population and the agent bid above the proxy bid at time \( t \) and the population bids higher, and thus the next listed price is \( u_t \), and the agent is not the highest bidder. In Eq. (5.10) the agent bids higher than the population, and thus the next listed price is \( q_t \), and the agent is the highest bidder. In Eqs. (5.11) and (5.12), the agent bids higher than the proxy bid, and thus the proxy bid is equal to \( u_t \), while the listed price is updated to \( \max(q_t, x_t) \). Note that we use strict inequalities to ensure that the agent is the highest bidder, since in the case of ties, the population is the highest bidder. Finally in Eqs. (5.13)-(5.14) \( v_t = 0 \) and so the populations bid is competing against the agent’s proxy bid.

5.2.6 Objective

We assume that the agent wants to maximize the expected utility

\[
\text{maximize } E[U(x_{T+1}, h_{T+1})].
\]

We will focus on the utility function

\[
U(x_{T+1}, h_{T+1}) = w_{T+1}(A - x_{T+1}).
\]  

(5.15)

The utility (5.15) implies that the agent will not bid for an item beyond his budget \( A \), he wants to win the auction at the lowest possible price, and he is indifferent between not winning the auction and winning it at the budget \( A \).

The choice of this particular model is guided by the requirements (a)-(c) outlined in the Introduction. We could include a more intricate state; for example we could include the number of bids at time \( t \) as an indicator of the auction’s interest; however, the tractability of the model would decrease, but most importantly the estimation of
the relevant probability distributions would become substantially more difficult given
the sparsity of the data.

5.2.7 Bellman equation

The problem of maximizing the expected utility in a single item auction can be solved
using the Bellman equation:

\[ J_{t+1}(x_{t+1}, h_{t+1}) = U(x_{t+1}, h_{t+1}) \]

If \( w_t = 0 \) then

\[
J_t(x_t, h_t) = \max_{u_t \in F_t(x_t, h_t)} \mathbb{E}_{q_t, v_t, \bar{h}_t}[J_{t+1}(x_{t+1}, h_{t+1})], \quad t = 1, \ldots, T,
\]

\[ = \max_{u_t \in F_t(x_t, h_t)} \sum_{q=0}^{A} \sum_{v=0}^{A} \sum_{h=x_t}^{h} J_{t+1}(f(x_t, h_t, u_t, q, v, \bar{h}), g(h_t, u_t, q, v, \bar{h})) \]

\[ \cdot \mathbb{P}(q_t = q, \bar{h}_t = \bar{h}|x_t) \mathbb{P}(v_t = v). \tag{5.16} \]

If \( w_t = 1 \) then

\[
J_t(x_t, h_t) = \max_{u_t \in F_t(x_t, h_t)} \mathbb{E}_{q_t, v_t}[J_{t+1}(x_{t+1}, h_{t+1})], \quad t = 1, \ldots, T,
\]

\[ = \max_{u_t \in F_t(x_t, h_t)} \sum_{q=0}^{A} \sum_{v=0}^{A} \sum_{h=0}^{h} J_{t+1}(f(x_t, h_t, u_t, q, v, 0), g(h_t, u_t, q, v, 0)) \]

\[ \cdot \mathbb{P}(q_t = q|x_t) \mathbb{P}(v_t = v). \tag{5.17} \]

Note that in Eq. (5.16), when the agent does not have a proxy bid, the expectation
is taken over \( q_t, \bar{h}_t \) and \( v_t \), whereas in Eq. (5.17), when the agent has a proxy bid, the
expectation is taken over only \( q_t \) and \( v_t \) \((\bar{h}_t = 0)\). We set \( \mathbb{P}(q_t = A, \bar{h}_t = \bar{h}|x_t) = \mathbb{P}(q_t = q, \bar{h}_t = A|x_t) \) equal to \( \mathbb{P}(q_t \geq A, \bar{h}_t = \bar{h}|x_t) \) and \( \mathbb{P}(q_t = q, \bar{h}_t \geq A|x_t) \) respectively,
since if a bid from the population is ever greater than or equal to \( A \) then the agent
cannot win. If the agent has a proxy bid, then we set \( \mathbb{P}(q_t = A|x_t) = \mathbb{P}(q_t \geq A|x_t). \)
5.2.8 Estimation of parameters

As we have mentioned, perhaps the most important guiding principle for the current model, is that the model’s parameters should be estimated from the data that is publicly available from eBay. eBay publishes the history of auctions, and thus the prices \( h_t \) are readily available, with the exception of \( h_{T+1} \), which is not publicized. Given this information, and the time of bids and identity of bidders, we calculate the listed price reported to the bidder when their bid was submitted. We can thus find the empirical distribution for \( P(q_t = j|x_t, w_t) \) and \( P(h_t = j|x_t, w_t) \). We have found no dependence on \( w_t \), and thus we calculated \( P(q_t = j|x_t) \) and \( P(h_t = j|x_t) \). To reduce the size of the estimation problem, and to eliminate having to deal with extremely sparse distribution matrices, we round up bids \( u_t \) and listed prices \( x_t \) to \([u_t/10]\) and \([x_t/10]\). For example, an observed listed price of $45 at time \( t \) is counted as \( x_t = 5 \).

Since we are modeling only a single competing bid from the population and not the many that could arrive during a given time period, we calculate the distribution of the maximum bid to occur for a given \((x_t, t)\). Let \( q_t \) be an actual bid at a real time \( s \), and similarly for \( x_t \), and let \( \hat{s}_t \) be the actual time, in seconds, at which period \( t \) begins. Thus, we calculate \( \max_{\hat{s}_t \leq s < \hat{s}_{t+1}} \hat{q}_s \). Then,

\[ P(q_t = q|x_t) = P\left(q_t = \left\lfloor \frac{\max_{\hat{s}_t \leq s < \hat{s}_{t+1}} \hat{q}_s}{10} \right\rfloor \middle| x_t = \left\lfloor \frac{\hat{x}_t}{10} \right\rfloor \right), \]

where the right hand side is calculated empirically.

We have calculated the empirical bidding distribution, adjusted as described above, for personal digital assistants (PDAs) and stamp collections, in an attempt to capture the effect of private and common value auctions, respectively. For PDAs, we looked at the Palm Pilot III model, whose final selling price was between $70 and $200. In total, there were 22478 bids in 1772 auctions over a two week period, with the mean auction lasting 5 days and receiving bids from just over 4 unique bidders on average. As an example, Figure 5-1 presents the empirical distribution of bids submitted between 6 and 12 hours from the end of the auction. Note that for a given listed price, bids are either zero (no bid), or they are distributed at values above the
Figure 5-1: The empirical distribution of the population's bid $q_t$ for Palm Pilot IIIs for a given listed price $x_t$. Here $t$ represents the time period of 6 to 12 hours remaining in the auction. Note that since the agent's budget is $A = 150$, bids by the population above $160$ are counted as $160$.

listed price (the 22478 recorded bids do not include 'zero bids'). Similarly, we have also calculated the bidding distribution of the population for stamp collections with final selling prices ranging from $100$ to $1000$. The data was taken from 4208 completed auctions with 50766 total bids during the same period, with the mean auction lasting 7.5 days and receiving bids from 3 different bidders on average. For this set of data, bid increments of $50$ were used. The empirical distribution of $P(h_t = j|x_t)$ has been calculated similarly.

As noted earlier, Roth and Ockenfels [40] observed that the number of bids increases as auctions near their end, and that the distribution of the arrival time of bids in the final seconds obeys a power law. In order to capture this phenomenon for the data we used, we consider different time horizons, denoted by $S$, before the end of the auction: 3 days, 6 hours, 10 minutes and 1 minute. For each separate $S$, we partition the time interval $[0, S]$ into ten subintervals $a_1 = [0, 0.1S]$, $a_2 = [0.1S, 0.2S], \ldots, a_{10} = [0.9S, S]$, so that $a_1$ represents the final tenth of an inter-
val of total length $S$. For example, when $S = 10$ minutes, $a_1$ is the final minute of the auction. For each interval $a_i$, $i = 1, \ldots, 10$ we record the fraction of all the bids in $[0, S]$ that arrived within this period. Figure 5-2 shows the fraction of bids in each interval $a_i$ as a function of the percentage of the respective time scale, that is $0.1 \times i$, for all the four values of $S$ for the data for Palm Pilots III and stamp collections. Figure 5-2 suggests that the distribution of the timing of these bids is identical for the times $S$ equal to 3 days, 6 hours and 10 minutes. For $S$ equals 1 minute, it is still the same for all but the first interval $a_1$, that is, within 6 seconds, before the end of the auction. An explanation of this phenomenon is to assume that due to network congestion and other phenomena, there is a probability $p$ of a bid being accepted during the last seconds of an auction. An approximate estimate of $p$ is then given as follows.

We first make a distinction between submitted bids, and accepted bids. The former are bids intended to be submitted, and the latter is what registers on the website. For all subintervals except that of $a_1$ for interval $S=1$ minute, submitted bids are accepted bids. However, for the final subinterval for $S=1$ minute, submitted bids are accepted bids with probability $p$. We assume that the distribution of submitted bids over a particular interval $S$ is the same for all intervals. For $S = 1$ minute, the observed fraction of bids arriving in interval $a_1$ is

$$P(\text{accepted in } a_1 | \text{accepted}) = \frac{P(\text{accepted in } a_1)}{P(\text{accepted})}$$

$$= \frac{P(\text{accepted in } a_1 | \text{submitted in } a_1)P(\text{submitted in } a_1)}{\sum_{i=1}^{10} P(\text{accepted in } a_i | \text{submitted in } a_i)P(\text{submitted in } a_i)}.$$

(5.18)

$$P(\text{accepted in } a_i | \text{submitted in } a_i) = p \text{ for } i = 1 \text{ and equals } 1, \text{ otherwise. Figure 5-2 suggests that } 0.41 = (0.45 + 0.42 + 0.36)/3 = P(\text{submitted in } a_1). \text{ Figure 5-2 also suggests that for } S=1 \text{ minute } P(\text{accepted in } a_1 | \text{submitted in } a_1) = 0.16. \text{ Then, from Eq. (5.19) we have } 0.41p/(0.41p + (1 - 0.41)) = 0.16, \text{ leading to an estimate of } p \approx 0.27. \text{ We have tested } DP \text{ over a broad range of } p \text{ and found that for the different}$$

132
values there is no qualitative difference in the results. In our experiments, we use a 
p-value of 0.8, which is based roughly on experience.

5.2.9 Empirical results

Having estimated its parameters, we have applied the model as follows:

(a) For bidding for a Palm Pilot III, we used a utility of the form (5.15) with a 
budget of $150. Since we have clustered the data into $10 increments the utility 
function becomes \( U(x_{T+1}, h_{T+1}) = 10(A - x_{T+1})w_{T+1} \) with \( A = 15 \) and where \( x_t \) 
measures the listed price in tens of dollars. We set \( T = 13 \) using the time steps 
described earlier.

(b) For bidding for stamp collections, we used a budget of $500, $50 increments, and 
a utility function \( U(x_{T+1}, h_{T+1}) = 50(A - x_{T+1})w_{T+1} \) with \( A = 10 \), to represent 
the budget of $500.

To test the performance of the algorithm in simulation we first compute the op-
timal cost to go and optimal decision for every state \((x_t, h_t)\), for \( t = 1, \ldots, T \) using 
Eq. (5.16). For the purposes of the simulation experiment, bids are drawn from the 
same distribution for which the algorithm was constructed and upon arriving in a
new state of the auction, the optimal bid is determined following the dynamic programming algorithm. The next states are computed using update rules (5.1)-(5.8) and the auction proceeds. At the end of the auction, period $T + 1$, the winner is declared and the appropriate utility is received. The following reported results are based on 10,000 simulations.

The optimal bidding policy depends on the estimated data. In Tables 5.1 and 5.2, we report the empirically observed optimal bidding policy for the estimated data for $p = 0.27$ and $p = 0.8$ respectively. Bidding in the early stages of the auction is not optimal since it can only lead to higher listed prices later on in the auction. However, because bids submitted in period $T$ are not guaranteed to be accepted, it is optimal to submit a bid in $T - 1$. As bids in period $T - 1$ also lead to a higher listed price in $T$, a trade-off emerges between having a proxy bid and causing the listed price to be too large. As expected, the tables show that it is optimal to bid more when $p$ is smaller in time period $T - 1$. Note that the algorithm suggests bidding more when the agent is the highest bidder in $T - 1$. This is due to the reduction of uncertainty one faces when he is the highest bidder. These two tables show that qualitatively, there is a difference in bidding strategy resulting from the two values of $p$ tested, but it is very small.

Tables 5.3 and 5.4 show the effect $p$ has on the performance of $DP$ for Palm Pilots
<table>
<thead>
<tr>
<th>p-value</th>
<th>Win %</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27</td>
<td>69.0</td>
<td>30.5</td>
<td>105.8</td>
</tr>
<tr>
<td>0.7</td>
<td>69.5</td>
<td>30.7</td>
<td>105.8</td>
</tr>
<tr>
<td>0.8</td>
<td>69.6</td>
<td>30.7</td>
<td>105.8</td>
</tr>
<tr>
<td>0.9</td>
<td>69.4</td>
<td>31.0</td>
<td>105.4</td>
</tr>
<tr>
<td>1.0</td>
<td>70.2</td>
<td>32.0</td>
<td>104.4</td>
</tr>
</tbody>
</table>

Table 5.3: Performance of DP Policy for Palm Pilot IIIIs for a range of p-values.

<table>
<thead>
<tr>
<th>p-value</th>
<th>Win %</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27</td>
<td>98.9</td>
<td>332.7</td>
<td>163.6</td>
</tr>
<tr>
<td>0.7</td>
<td>98.9</td>
<td>332.7</td>
<td>163.6</td>
</tr>
<tr>
<td>0.8</td>
<td>98.9</td>
<td>332.7</td>
<td>163.6</td>
</tr>
<tr>
<td>0.9</td>
<td>98.9</td>
<td>332.8</td>
<td>163.5</td>
</tr>
<tr>
<td>1.0</td>
<td>99.0</td>
<td>333.7</td>
<td>163.0</td>
</tr>
</tbody>
</table>

Table 5.4: Performance of DP Policy for Stamps for a range of p-values.

and stamp collections, respectively. In both cases the effects are small.

Table 5.4 shows the varying effect p has on the average utility and winning percentage when bidding for stamps. The high budget of the seller relative to what the market is willing to pay means that the effect of p is small. The conclusion drawn from Tables 5.1 - 5.4 is that the effect of the value of p on the performance of DP is small. For the remainder of this chapter we will use p = 0.8.

Table 5.5 shows the results of the algorithm after 10,000 simulations with A = 15, for four different bidding strategies for stamp collections: (a) The dynamic programming policy; Bidding the budget A (b) at time t = 0 (the beginning of the auction); (c) at time t = T - 1; (d) at time t= T. The dynamic programming based policy was clearly the best. Although it didn't lead to wins as often as bidding A at t = 0 or t = T - 1, the average utility was far larger. Note that the average utility is equal to the probability of winning times 150 minus the average spent per win.

The reason for dynamic programming's success is that it is not restricted to making bids at specified times, but can instead manipulate the auction and bid when required. On average, the agent spent $105.8 per win using the dynamic programming based policy. We implemented this algorithm using similar data to bid for a Palm Pilot III in an on-line auction and the item was won for $92.
<table>
<thead>
<tr>
<th>Policy</th>
<th>Win %</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>69.6</td>
<td>30.7</td>
<td>105.8</td>
</tr>
<tr>
<td>Bid A at $t = 0$</td>
<td>51.8</td>
<td>23.1</td>
<td>105.4</td>
</tr>
<tr>
<td>Bid A at $t = T - 1$</td>
<td>55.9</td>
<td>24.3</td>
<td>106.5</td>
</tr>
<tr>
<td>Bid A at $t = T$</td>
<td>46.5</td>
<td>20.8</td>
<td>105.3</td>
</tr>
</tbody>
</table>

Table 5.5: Performance of bidding strategies for Palm Pilot III.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Win %</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>98.9</td>
<td>332.7</td>
<td>163.6</td>
</tr>
<tr>
<td>Bid A at $t = 0$</td>
<td>93.9</td>
<td>285.0</td>
<td>196.6</td>
</tr>
<tr>
<td>Bid A at $t = T - 1$</td>
<td>98.2</td>
<td>325.9</td>
<td>168.4</td>
</tr>
<tr>
<td>Bid A at $t = T$</td>
<td>78.6</td>
<td>260.0</td>
<td>169.3</td>
</tr>
</tbody>
</table>

Table 5.6: Performance of bidding strategies for stamp collections.

Table 5.6 shows the results of the algorithm after 10,000 simulations with $A = 10$, for different bidding strategies for stamp collections. In this case, the listed price and all bids were rounded to $50$ increments. Again the optimal policy is the clear winner. Not only does it win 99% of the time, it spends $163.6$ per win, versus 98% winning percentage and $168.4$ per win for the next closest policy. We have used this algorithm to win over one thousand stamp collections and individual stamps in eBay.

### 5.2.10 Bidding against multiple competitors

In this section we consider an agent bidding against both the population’s bid and an additional competitor’s bid. The purpose of this analysis is to illustrate the robustness of the DP as well as show how information about the competitor’s strategy can be used to improve the performance of the DP.

Tables 5.7 and 5.8 show the results of bidding against a competitor of different budgets for Palm Pilots when the agent’s budget is 150. In Table 5.7, the strategy of the competitor is to bid his budget at time $T - 1$. Table 5.8 applies to the case that the competitor’s strategy is also a DP strategy solved for the particular budget. In both cases the competitor’s utility is equal to his budget minus the price paid if he wins the object, and zero otherwise. Both tables show that as the competitor’s budget increases the agent’s expected utility decreases only slightly, except for when both
### Table 5.7: Performance of bidding against Policy ‘Bid A at T-1’ for Palm Pilot IIIIs, the agent’s budget is 150.

| Competitor’s Budget | DP | | | Competitor | | | |
|---------------------|----|---|----------------|----------------|
|                     | Win % | Avg. Utility | Avg. Spent per Win | Win % | Avg. Utility | Avg. Spent per Win |
| 100                 | 67.8  | 25.7          | 112.0              | 1.0   | 0.1          | 89.7              |
| 110                 | 68.5  | 22.7          | 116.9              | 2.0   | 0.2          | 98.6              |
| 120                 | 65.5  | 17.5          | 123.3              | 3.4   | 0.5          | 105.0             |
| 130                 | 65.7  | 12.6          | 130.8              | 4.4   | 0.9          | 109.3             |
| 140                 | 58.5  | 5.8           | 140.0              | 7.4   | 1.4          | 120.8             |
| 150                 | 6.5   | 0.0           | 150.0              | 58.0  | 2.5          | 145.7             |

### Table 5.8: Performance of bidding against DP Policy for Palm Pilot IIIIs, the agent’s budget is 150.

| Competitor’s Budget | DP | | | Competitor | | | |
|---------------------|----|---|----------------|----------------|
|                     | Win % | Avg. Utility | Avg. Spent per Win | Win % | Avg. Utility | Avg. Spent per Win |
| 100                 | 60.1  | 20.2          | 116.3              | 0.0   | 0.0          | 95.2              |
| 110                 | 60.5  | 18.2          | 119.9              | 0.1   | 0.0          | 90.0              |
| 120                 | 59.5  | 15.4          | 124.1              | 0.3   | 0.0          | 107.8             |
| 130                 | 59.4  | 11.6          | 130.5              | 0.6   | 0.1          | 115.4             |
| 140                 | 45.7  | 4.9           | 139.3              | 3.8   | 0.1          | 136.6             |
| 150                 | 22.0  | 0.8           | 146.5              | 22.0  | 0.8          | 146.5             |


<table>
<thead>
<tr>
<th>Probability of Competitor's Entrance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win %</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.25</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.75</td>
</tr>
<tr>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 5.9: Performance of bidding against Policy ‘Bid A at T-1’ with competitor’s budget of 140 for Palm Pilot IIIIs, for different anticipated entrance probabilities.

parties have the same budget. Note that because the DP’s objective is to maximize expected utility, and not the probability of winning, the strategy employed by the agent allows the competitor to sometimes win even with a smaller budget.

In the case of a Palm Pilot, Table 5.9 shows the results of bidding against a competitor, when the agent’s budget is 150 (this involves first solving the Bellman Equations (5.16) and (5.17) with the population bids $q_t$ and $h_t$ and the competitor’s bid). However, the competitor’s bid was present only with a given probability to reflect the agent’s uncertainty as to whether the competitor would be present or not later in the auction. Nevertheless, in the simulations the competitor did bid in every auction. The agent’s anticipated probability of the competitor bidding is reflected in the column ‘Probability of Competitor’s Entrance’. The simulations use a competitor with a budget of 140. The poor performance of the DP for entrance probabilities of less than one, is a result of the DP attempting to win at low prices without anticipating the scenario in which there is a competitor bidding.

5.3 Multiple auctions

We consider an agent interested in participating in $N$ simultaneous auctions all ending at the same time. In each auction $i=1,\ldots,N$, the agent is willing to bid no more than $A_i$, and no more than $A$ over all auctions.

For $t = 1, \ldots, T+1$, and $i = 1, \ldots, N$ the state of each auction is $(x^i_t, h^i_t)$; the control is $u^i_t$; randomness is given by the vector $(q^i_t, v^i_t, \overline{h}^i_t)$. We denote the corresponding
vectors by \((x_t, h_t), u_t\) and \((q_t, v_t, \overline{h}_t)\). We use \(w^i_t = 1\) if \(h^i_t > 0\), and zero, otherwise; \(w_t\) denotes the vector of \(w^i_t\). The set of feasible controls is given by:

\[
F_t(x_t, h_t) = \left\{ u_t \mid u^i_t \in F_t(x^i_t, h^i_t), \ i = 1, \ldots, N, \sum_{i=1}^{N} u^i_t \leq A \right\}.
\]

The utility is given by

\[
U(x_{T+1}, h_{T+1}) = \sum_{i=1}^{N} (A_i - x^i_{T+1})w^i_{T+1},
\]

and the dynamics are given analogously to Eqs. (5.1)-(5.8). We denote by

\[
f_t(x_t, h_t, u_t, q_t, v_t, \overline{h}_t) = (f_t(x^1_t, h^1_t, u^1_t, q^1_t, v^1_t, \overline{h}^1_t), \ldots, f_t(x^N_t, h^N_t, u^N_t, q^N_t, v^N_t, \overline{h}^N_t)),
\]

and likewise for \(g_t(\cdot)\). Note that with the utility function as described, the agent’s goal is to win each of the \(N\) auctions at the lowest possible price. If the agent’s goal is to win fewer than \(M < N\) items, then the same utility function is used, but the agent must constrain his bidding so that he is never the leading bidder in more than \(M\) auctions at a time.

We assume that \(P(q^i_t = j, \overline{h}^i_t = j|x_t)\) is known, in other words the bids of the population and the proxy bids depend on the listed prices of all auctions. To simplify notation, we use

\[
\sum_{q=0}^{A} = \sum_{q^1=0}^{A_1} \cdots \sum_{q^N=0}^{A_N}, \quad \sum_{v=0}^{1} = \sum_{v^1=0}^{1} \cdots \sum_{v^N=0}^{1}, \quad \sum_{h=x}^{A} = \sum_{h^1=x^1}^{A_1} \cdots \sum_{h^N=x^N}^{A_N}.
\]

Bellman’s equation is thus given by:

\[
J_{T+1}(x_{T+1}, h_{T+1}) = U(x_{T+1}, h_{T+1})
\]
\[ J_t(x_t, h_t) \]
\[ = \max_{u_t \in F_t(x_t, w_t)} \mathbb{E}_{q_t, v_t, h_t} [J_{t+1}(x_{t+1}, w_{t+1})], \quad t = 1, \ldots, T, \]
\[ = \max_{u_t \in F_t(x_t, w_t)} \sum_{q=0}^{A} \sum_{v=0}^{A} \sum_{h=0}^{A} J_{t+1}(f_t(x_t, h_t, u_t, q, v, h), g_t(h_t, u_t, q, v, h)) \]
\[ \cdot \prod_{i : w_i^t = 0} P(q_i^t = q_i, h_i^t = h_i | x_t) \prod_{i=1}^{N} P(v_i^t = v_i). \quad (5.20) \]

Note that \( h_i^t = 0 \), when \( w_i^t = 1 \), and thus we only take expectations in Eq. (5.20) over only those \( h_i^t \) for which \( w_i^t = 0 \). In practice of course, the computation from Eqs. (5.20) is barely feasible even for two auctions. Moreover, it is infeasible for three simultaneous auctions given the high dimension of Bellman's equation. For this reason, we propose in the next subsections several approximate dynamic programming methods.

5.3.1 Approximate dynamic programming method 1

The method we consider in this and the next section belongs in the class of methods of approximate dynamic programming (see Bertsekas and Tsitsiklis [12]). Under this method, abbreviated as ADP1, for each of the \( 2^N \) binary vectors \( w_t \in \{0, 1\}^N \) we approximate the cost-to-go function \( J_t(x_t, h_t) \) as follows:

\[ \hat{J}_t(x_t, h_t) = r_0(w_t, t) + \sum_{i=1}^{N} r_i(w_t, t)x_i^t, \]

where each of the coefficients \( r_i(w_t, t), i = 0, 1, \ldots, N \) are defined for each of the \( 2^N \) vectors \( w_t \).

By its nature, this approach works only for up to \( N = 5 \) auctions. We use simulation to generate feasible states \((x_t, h_t)\). The overall algorithm is as follows.

Algorithm ADP1:

1. For time period \( t = T, \ldots, 1 \) and each \( w \in \{0, 1\}^N \) select by simulation a set \( X_t(w) \) of states \((x_t(k), h_t(k))\) indexed by \( k \).
2. For each \((x_t(k), h_t(k)) \in X_t(w)\) compute

\[
\tilde{J}_t(x_t(k), h_t(k)) = \max_{u_t \in F_t(x_t(k), h_t(k))} \mathbb{E}[\tilde{J}_{t+1}(x_{t+1}, h_{t+1})],
\]

where

\[
\tilde{J}_t(x, h) = r_0(w, t) + \sum_{i=1}^{N} r_i(w, t)x^i.
\]

3. For each \(w \in \{0, 1\}^N\), find parameters \(r(w, t)\) by regression, i.e., solving the least squares problem:

\[
\sum_{(x_t(k), w) \in X_t(w)} \left( \tilde{J}_t(x_t(k), h_t(k)) - r_0(w, t) - \sum_{i=1}^{N} r_i(w, t)x^i_t(k) \right)^2.
\]

Notice that the algorithm is still exponential in \(N\) as the cost-to-go function for each time \(t\) is approximated by \(2^N\) linear functions, each corresponding to a distinct vector \(w\).

### 5.3.2 Approximate dynamic programming method 2

This method, abbreviated as \(ADP2\), is similar to the previous method, but instead of using \(2^N\) linear (in \(x_t\)) functions to approximate \(J_t(\cdot)\) it uses \(N + 1\) linear functions. In this method, the cost-to-go-function only depends on \(a = \sum_{i=1}^{N} w^i_t\), that is, the number of auctions the agent is the highest bidder at time \(t\). In this method, we only need to evaluate \(N + 1\) vectors \(r(a, t), a = 0, \ldots, N\) and \(t = 1, \ldots, T\). Although this uses a coarser approximation than method \(A\), it is capable to solve problems with a larger number of auctions.

### 5.3.3 Integer programming approximation

We consider the Lagrange-based scheme presented in Section 2.4.1 with the Lagrange multiplier set to zero. Under this method, abbreviated as \(IPA\), we let \(d_t^i(x_t^i, h_t^i, j)\) denote the expected utility of bidding \(j\) in auction \(i\) given state \((x_t^i, h_t^i)\) and optimally
bidding in this single auction thereafter. This is calculated as

\[ d_t^i(x_t^i, h_t^i, j) = E_{q_t^i, v_t^i, h_t^i}[J_{t+1}^i(f(x_t^i, h_t^i, j, q_t^i, v_t^i, h_t^i)) \cdot g(h_t^i, j, q_t^i, v_t^i, h_t^i)], \quad (5.23) \]

with

\[ J_t^i(x_t^i, h_t^i) = \max_j d_t^i(x_t^i, h_t^i, j). \quad (5.24) \]

Starting with \( J_{T+1}^i(x_{T+1}^i, h_{T+1}^i) = U(x_{T+1}^i, h_{T+1}^i) = (A_i - x_{T+1}^i)w_{T+1}^i \), we use Eqs. (5.23) and (5.24) to find \( d_t^i(x_t^i, h_t^i, j) \).

For a fixed time \( t \) we define the following decision variables \( u_t(j, t) \) as

\[ u_t(j, t) = \begin{cases} 
1, & \text{if the agent bids at least } j \text{ in auction } i \text{ at time } t, \\
0, & \text{otherwise.}
\end{cases} \]

Given the state \((x_t, h_t)\), and constants \( A_i, A_t \), the agent solves the following discrete optimization problem:

\[
\text{maximize } \sum_{i=1}^{N} \sum_{j=0}^{A_i} u_t(j, t)(d_t^i(x_t^i, h_t^i, j) - d_t^i(x_t^i, h_t^i, j - 1)) \quad (5.25)
\]

subject to

\[ u_t(j, t) \leq u_t(j - 1, t) \quad \forall i, j \quad (5.26) \]

\[ \sum_{i=1}^{N} \sum_{j=1}^{A_i} u_t(j, t) \leq A \quad (5.27) \]

\[ \sum_{j=1}^{A_i} u_t(j, t) \geq h_t^i \quad \forall i \quad (5.28) \]

\[ u_t(j, t) \in \{0, 1\} \quad \forall i, j, \]

where \( d_t^i(x_t^i, h_t^i, -1) = 0 \) \( \forall i, j \). The cost coefficients in (5.25) represent the marginal increase in utility for bidding one unit higher in a given auction. Note that if the agent bids \( j_0 \) in auction \( i \), that is \( u_t(j, t) = 1 \) for \( j \leq j_0 \), and \( u_t(j, t) = 0 \) for \( j > j_0 + 1 \), then the contribution to the objective function (5.25) is correctly \( d_t^i(x_t^i, h_t^i, j_0) \). Constraint (5.26) ensures that if we bid at least \( j \) in auction \( i \), then we had to have bid at least \( j - 1 \) in auction \( i \). Constraint (5.27) is the way auctions interact, that is through a
global budget. Constraint (5.28) ensures that if the agent is the highest bidder in auction \( i \) at time \( t \), that is \( h_t^i > 0 \), then his bid at time \( t \) should be larger than his proxy bid at time \( t - 1 \).

Note that the solution to Problem (5.25) only provides an approximate solution method as it ignores the budget constraint in future periods. It also does not take into account the possibility that the bids of the population in different auctions might be correlated. It can be seen that this is integer optimization approach presented in Section 2.4.1. In particular, we set \( \lambda = 0 \) for all future constraints and retain the immediate constraints. The heuristic calls for choosing a set of decisions that maximize the approximate cost to go of the result decoupled problem.

### 5.3.4 Pairwise integer programming approximation method

In this section, we propose a more elaborate approximation method based on integer programming. Under this method, abbreviated as \( PIPA_1 \), we optimally solve all pairs of auctions using the exact dynamic programming method, and then at each time stage, for a given state of the auctions, find the bid that maximizes the sum of the expected cost-to-go over all pairs of auctions. This algorithm is closely related to the \( M \)-order integer optimization approach of Section 3.2.6 In particular, \( M = 2 \), all Lagrange multipliers for future constraints are set to zero and the immediate constraints are retained.

Let \( M = \{ (i, k) | i, k = 1, \ldots, N, i < k \} \) be the set of all \( \binom{N}{2} \) pairs of auctions. As before we solve the two auction problem optimally by dynamic programming. This enables us to compute for all pairings \( (i, k) \) the quantity \( d_t^{(i,k)}(r, s) \), the expected cost to go after bidding \( r \) in auction \( i \) and \( s \) in auction \( k \) at time \( t \). Given the optimal cost to go function \( J_t(x_t, h_t) \) calculated from Eq. (5.20) for a two auction problem, the quantities \( d_t^{(i,k)}(r, s) \) are given by:

\[
d_t^{(i,k)}(r, s) = \mathbb{E}[J_{t+1}(f_t(x_t, h_t, (r, s), q_t, v_t, \tilde{h}_t), g_t(h_t, (r, s), q_t, v_t, \tilde{h}_t))]. \tag{5.29}
\]
We define the decision variable $u_{i,k}(r, s, t)$, which is equal to one if the agent bids at least $r$ in auction $i$ and at least $s$ in auction $k$ at time $t$, and is 0, otherwise. At time $t$, for a given state $(x_t, h_t)$ the agent solves the following discrete optimization problem:

$$\begin{align*}
\text{max} & \quad \sum_{(i,k) \in M} \sum_{r=0}^{A_i} \sum_{s=0}^{A_k} u_{i,k}(r, s, t) (d_t^{(i,k)}(r, s, t) - d_t^{(i,k)}(r-1, s, t)) \\
& \quad -d_t^{(i,k)}(r, s-1) + d_t^{(i,k)}(r-1, s-1)) \\
\text{s.t.} & \quad u_{i,k}(r, s, t) \leq u_{i,k}(r-1, s, t) \quad (5.30) \\
& \quad u_{i,k}(r, s, t) \leq u_{m}(r, s-1, t) \quad (5.31) \\
& \quad u_{i,k}(r, s, t) - u_{i,k}(r-1, s, t) - u_{i,k}(r, s-1, t) \\
& \quad +u_{i,k}(r-1, s-1, t) \geq 0 \quad \forall (i, k) \in M, \, \forall r, s \quad (5.32) \\
& \quad u_{i,k}(r, 0, t) - u_{i,k}(r, 0, t) = 0 \quad \forall i, k, l, r \quad (5.33) \\
& \quad u_{i,k}(r, 0, t) - u_{i,k}(0, r, t) = 0 \quad \forall i, k, l, r \quad (5.34) \\
& \quad u_{i,k}(0, r, t) - u_{i,k}(0, r, t) = 0 \quad \forall i, k, l, r \quad (5.35) \\
& \quad \sum_{r=1}^{A} u_{(1,2)}(r, 0, t) + \sum_{n_2=2}^{N} \sum_{r=1}^{A} u_{(1,n_2)}(0, r, t) \leq A \quad (5.36) \\
& \quad \sum_{r=1}^{A_1} u_{(1,2)}(r, 0, t) \geq h_t^1 \quad (5.37) \\
& \quad \sum_{r=1}^{A_k} u_{(1,k)}(0, r, t) \geq h_t^k \quad (5.38) \\
& \quad u_{i,k}(r, s, t) \in \{0, 1\},
\end{align*}$$

with $d_t^{(i,k)}(r, s, t) = 0$ if $r$ or $s = -1$. The optimal bidding vector is

$$\left( \sum_{r=1}^{A} u_{(1,2)}(r, 0, t), \sum_{r=1}^{A} u_{(1,3)}(0, r, t), \ldots, \sum_{r=1}^{A} u_{(1,N)}(0, r, t) \right).$$

The cost coefficients in (5.30) represent the marginal increase in utility for bidding one unit higher in both auctions of a given pair. Constraint (5.31) enforces that if
the agent bids at least $r$ in auction $i$, then he has to bid at least $r - 1$. Likewise for Constraint (5.32). Constraint (5.33) enforces that if the agent bids at least $r$ in auction $i$, at least $s - 1$ in auction $k$, and at least $r - 1$ in auction $i$ and at least $s$ in auction $k$, then he has to bid at least $r$ in auction $i$ and at least $s$ in auction $k$. Constraints (5.34)-(5.36) enforce consistent decisions in each auction pairing. Constraint (5.37) is the global budget constraint. Finally, Constraints (5.38), (5.39) ensure that if the agent is the highest bidder in auction $k$ at time $t$, that is $h^k_t > 0$, then his bid at time $t$ should be larger than his proxy bid at time $t - 1$.

5.3.5 Pairwise integer programming approximation method

The computational burden of the pairwise integer programming approximation is considerable as we need to solve $\binom{N}{2}$ pairs of auctions exactly. Alternatively, we can solve $N/2$ disjoint pairs of auctions and combine the cost to go functions in an integer programming problem. We omit the details as they are very similar to what we have already presented. We abbreviate the method as PIPA2.

5.3.6 Empirical results

We consider an agent bidding for an identical item in $N$ multiple auctions for $N = 2, 3, 6$, where the item is valued at $A$. In this case $A_i = A$. The utility received at the end of the auction is

$$\mathcal{U}(x_{T+1}, h_{T+1}) = C \sum_{i=1}^{N} (A_i \cdot x_{T+1}^{i} / w_{T+1}^{i}).$$

(5.40)

We set $A = A_i = 15$ and $C = 10$ for Palm Pilots III, and $A = A_i = 10$ and $C = 50$ for stamp collections. We use $T = 13$ and $p = 0.8$ and the competing bidding distributions are calculated as in Section 5.2.

We have implemented all the methods proposed: the exact dynamic programming method for $N = 2$ abbreviated as DP; the approximate dynamic programming
<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>41.4</td>
<td>39.3</td>
<td>102.5</td>
</tr>
<tr>
<td>ADP1</td>
<td>39.4</td>
<td>34.8</td>
<td>105.8</td>
</tr>
<tr>
<td>ADP2</td>
<td>38.4</td>
<td>34.7</td>
<td>104.8</td>
</tr>
<tr>
<td>IPA</td>
<td>42.0</td>
<td>39.0</td>
<td>103.6</td>
</tr>
</tbody>
</table>

Table 5.10: Comparison of $DP$, $ADP1$, $ADP2$ and $IPA$ for $N = 2$ auctions, $A = 15$, $C = 10$ and data from Palm Pilots III.

methods of Sections 5.3.1 and 5.3.2 abbreviated as $ADP1$ and $ADP2$ respectively; the integer programming based methods of Sections 5.3.3, 5.3.4 and 5.3.5 abbreviated as $IPA$, $PIPA1$ and $PIPA2$ respectively.

Tables 5.10-5.12 and 5.13-5.15 report simulation results averaged over 10,000 simulations of $N = 2, 3, 6$ simultaneous auctions using eBay data for Palm Pilots III, and stamp collections respectively.

In Table 5.10 we compare the performance of $DP$, $ADP1$, $ADP2$ and $IPA$ for $N = 2$ auctions with the goal of giving insight on the degree of suboptimality of the approximate methods compared to the optimal one. Notice that for $N = 3$, solving the exact dynamic programming problem is computationally infeasible. In Table 5.11 in addition to $ADP1$, $ADP2$ and $IPA$, we include $PIPA1$ in the comparison. In Table 5.12, we compare $IPA$ and $PIPA2$ for $N = 6$ auctions. The Column labeled "$\%\text{ Won}\$" is the percentage of auctions that were won, the labeled "$\%\text{ at least one win}\$" is the fraction of rounds (one round is one set of $N$ simultaneous auctions) in which at least one auction was won, and the Column "$\text{Avg. Spent per Win}\$" is the amount spent in dollars per auction won. If we set our budget $A = A_1 = \cdots = A_6$, then "$\%\text{ at least one win}\$" shows how much more often we win by competing in more auctions than just one. Tables 5.13-5.15 have the same comparisons but for stamp collections.

The results in Tables 5.10-5.12 and 5.13-5.15 suggest the following insights:

(a) The integer programming based methods ($IPA$, $PIPA1$) clearly outperform the approximate dynamic programming methods ($ADP1$, $ADP2$) (see Tables 5.10, 5.11, 5.13, 5.14).

(b) When it is computationally feasible to find the optimal policy ($N = 2$), $IPA$ is
<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADP1</td>
<td>29.3</td>
<td>17.6</td>
<td>130</td>
</tr>
<tr>
<td>ADP2</td>
<td>31.5</td>
<td>9.5</td>
<td>140</td>
</tr>
<tr>
<td>IPA</td>
<td>30.3</td>
<td>45.8</td>
<td>99.7</td>
</tr>
<tr>
<td>PIPA1</td>
<td>29.9</td>
<td>46.6</td>
<td>98.1</td>
</tr>
</tbody>
</table>

Table 5.11: Comparison of ADP1, ADP2, IPA, and PIPA1 for \( N = 3 \) auctions, \( A = 15, C = 10 \) and data from Palm Pilots III.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>% at least one win</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPA</td>
<td>15.5</td>
<td>93.1</td>
<td>54.9</td>
<td>92.1</td>
</tr>
<tr>
<td>PIPA2</td>
<td>15.9</td>
<td>94.7</td>
<td>55.4</td>
<td>91.7</td>
</tr>
</tbody>
</table>

Table 5.12: Comparison of IPA and PIPA2 for \( N = 6 \) auctions, \( A = 15, C = 10 \) and data from Palm Pilots III.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>90.5</td>
<td>626.4</td>
<td>153.9</td>
</tr>
<tr>
<td>ADP1</td>
<td>64.0</td>
<td>413.2</td>
<td>177.0</td>
</tr>
<tr>
<td>ADP2</td>
<td>57.4</td>
<td>364.0</td>
<td>183.0</td>
</tr>
<tr>
<td>IPA</td>
<td>90.3</td>
<td>624.3</td>
<td>154.5</td>
</tr>
</tbody>
</table>

Table 5.13: Comparison of DP, ADP1, ADP2 and IPA for \( N = 2 \) auctions, \( A = 10, C = 50 \) and data from stamp collections.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADP1</td>
<td>51.9</td>
<td>487.2</td>
<td>187.1</td>
</tr>
<tr>
<td>ADP2</td>
<td>47.2</td>
<td>429.2</td>
<td>196.9</td>
</tr>
<tr>
<td>IPA</td>
<td>64.0</td>
<td>677.0</td>
<td>147.3</td>
</tr>
<tr>
<td>PIPA1</td>
<td>64.3</td>
<td>686.6</td>
<td>144.3</td>
</tr>
</tbody>
</table>

Table 5.14: Comparison of ADP1, ADP2, IPA, and PIPA1 for \( N = 3 \) auctions, \( A = 10, C = 50 \) and data from stamp collections.

<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>% at least one win</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPA</td>
<td>34.2</td>
<td>99.3</td>
<td>759.6</td>
<td>129.4</td>
</tr>
<tr>
<td>PIPA2</td>
<td>34.0</td>
<td>99.7</td>
<td>762.8</td>
<td>126.4</td>
</tr>
</tbody>
</table>

Table 5.15: Comparison of IPA and PIPA2 for \( N = 6 \) auctions, \( A = 10, C = 50 \) and data from stamp collections.
<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>% Single Win</th>
<th>% Double Win</th>
<th>% Triple Win</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPA</td>
<td>54.4</td>
<td>28.1</td>
<td>66.4</td>
<td>0.8</td>
<td>71.6</td>
<td>106.1</td>
</tr>
<tr>
<td>PIPA1</td>
<td>54.4</td>
<td>28.1</td>
<td>66.4</td>
<td>0.8</td>
<td>71.6</td>
<td>106.1</td>
</tr>
</tbody>
</table>

Table 5.16: Comparison of IPA and PIPA1 for $N = 3$ auctions, $A_1 = A_2 = A_3 = A/2$, $A = 30$, $C = 10$, using Palm Pilots III data.

almost optimal (see Tables 5.10, 5.13). The exact dynamic programming policy leads to slightly higher utility.

(c) The more sophisticated PIPA1 (for $N = 3$) leads to slightly better solutions compared to IPA for Palm Pilots III data (see Table 5.11) and the same solutions for stamp collections data (see Table 5.14), but at the expense of much higher computational effort.

(d) IPA is outperformed only slightly by PIPA2 (see Tables 5.12, 5.15). For all its computational effort, PIPA2 has slightly greater average utility than IPA.

The emerging insight from all the computational results is that IPA seems an attractive method relative to the other methods. It is certainly significantly faster than all other methods, and its performance is very close to the more sophisticated PIPA1.

We next examine the robustness of this conclusion relative to the budget $A$. In Tables 5.16 and 5.17, we consider the case of bidding in $N = 3$ auctions with $A_1 = A_2 = A_3 = A/2$. For Palm Pilots III data we set $A = 30$, $C = 10$ and for stamp collections $A = 20$, $C = 50$. The columns labeled “% Single Win”, “% Double Win” and “% Triple Win” are the percentage $f_1$, $f_2$, $f_3$ of simulations in which 1 out of 3, 2 out of 3, and all 3 out of 3 auctions were won, respectively. The column labeled “% Won” is the fraction $f$ of auctions won, i.e., $f = (f_1 + 2f_2 + 3f_3)/3$. Note that the expected utility is equal to the fraction of wins $f$ times $N$ times the difference of $A/2$ and the average spent per win. The results in Tables 5.16 and 5.17 show that the performances of IPA and PIPA1 are identical. Thus, given that computationally IPA is faster and simpler, IPA is our proposed approach for the problem of multiple simultaneous auctions.

148
<table>
<thead>
<tr>
<th>Method</th>
<th>% Won</th>
<th>% Single Win</th>
<th>% Double Win</th>
<th>% Triple Win</th>
<th>Avg. Utility</th>
<th>Avg. Spent per Win</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPA</td>
<td>97.4</td>
<td>0.2</td>
<td>7.3</td>
<td>92.5</td>
<td>983.8</td>
<td>163.4</td>
</tr>
<tr>
<td>PIPA1</td>
<td>97.4</td>
<td>0.2</td>
<td>7.3</td>
<td>92.5</td>
<td>983.8</td>
<td>163.4</td>
</tr>
</tbody>
</table>

Table 5.17: Comparison of IPA and PIPA1 for $N = 3$ auctions, $A_1 = A_2 = A_3 = \frac{A}{2}$, $A = 20$, $C = 50$, using stamp collections data.

<table>
<thead>
<tr>
<th>Competitor’s Budget</th>
<th>IPA</th>
<th>Competitor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Win %</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>89.1</td>
</tr>
<tr>
<td>110</td>
<td></td>
<td>88.8</td>
</tr>
<tr>
<td>120</td>
<td></td>
<td>84.8</td>
</tr>
<tr>
<td>130</td>
<td></td>
<td>83.7</td>
</tr>
<tr>
<td>140</td>
<td></td>
<td>78.6</td>
</tr>
<tr>
<td>150</td>
<td></td>
<td>45.9</td>
</tr>
</tbody>
</table>

Table 5.18: Performance of bidding against Policy ‘Bid budget at T-1 in lowest listed price auction’ in 3 auctions, for Palm Pilot IIIIs, the agent’s budget is 150.

### 5.3.7 Bidding against a sophisticated competitor in multiple auctions

With the tremendous volume of trade occurring on eBay, it comes as no surprise that many similar goods are being auctioned off concurrently. As Zheng reports in [54], it is interesting to observe that bidders have taken advantage of this trend by employing the simple heuristic of bidding in the auction with the lowest listed price for a particular item. In this section we examine how IPA performs in a multi-bidder environment while competing in three simultaneous auctions. In addition to competing against bids from the population, we now consider a setting with an additional agent bidding in the same three auctions who has budget $A_2$ and employs the following strategy: Bid $A_2$ at time $T - 1$ in the auction with the lowest listed price. If outbid then bid $A_2$ at time $T$ in the auction with the lowest listed price, otherwise do not bid. In this three bidder environment ties between IPA and the competitor are randomly decided, while any tie with the population is won by the population.
Table 5.19: Performance of bidding against Policy ‘Bid budget of 140 at T-1 in lowest listed price auction’ in 3 auctions, for different anticipated probabilities of entrance into single auction, for Palm Pilot IIIIs, the agent’s budget is 150.

Table 5.18 shows the results of simultaneously bidding for Palm Pilots in three auctions against a population bid and a competitor. Here, ‘Win %’ is the probability of winning one out of the three auctions. The competitor’s utility is his budget minus the price paid if he won. The results indicate that as the competitors budget increases, strategy IPA causes the agent to spend more per auction on average and win less often. This is because the two bidders are often bidding in the same auction, which the agent will win since it has the greater of the two budgets. Note however that when the competitor’s budget is equal to the agent’s budget, the agent wins less often than in other scenarios, but also spends less. This is because the agent is only winning in auctions that have a low listed price and that the competitor has not bid in. These results indicate that the two strategies are similar.

Table 5.19 shows how IPA performs bidding for Palm Pilots in three auctions when IPA was constructed using a combination of the population’s bid and the competitor’s bid, as in Section 5.2.10. Note that since IPA solves auctions independently, we assume the entrance probability is the same for each auction and independent of other auctions. In simulations, the competitor is always present and bids in the auction with the lowest listed price at time \( T - 1 \). In this example, the competitor has a budget of 140. For the case when the entrance probability is less than one, IPA performs worse than if it does not know of the competitor. This occurs because IPA has trouble deciding between committing its budget to one auction in order to beat the competitor, and not bidding at all in order to keep the prices low. We notice improve-
ment in IPA's win percentage when the assumed entrance probability is one. These results show that while IPA is able to handle multiple auctions with a large enough budget, the algorithms inability to account for future bidding constraints can cause it to have difficulty in bidding against competitive agents bidding in multiple auctions. These results also demonstrate, however, that incorporating some information of the competitor's presence does increase IPA's performance by certain measures.

5.4 Multiple overlapping auctions

In this section, we extend our methods to the more general setting of a bidder interested in bidding simultaneously in multiple auctions, not all ending at the same time. The set of auctions we consider is fixed, that is we do not consider prospective auctions which are not already in process. Due to the high dimensionality required from an exact dynamic programming based approach, we focus on the integer programming approximation method IPA, as this was the method that gave the best results in the simultaneous auctions case.

Suppose there are currently \( N \) auctions currently in process. Let \( x^i, h^i, t^i \) be the listed price, proxy bid, and time remaining, respectively, in auction \( i \). Let \( A \) be the amount of the budget remaining, and \( A_i \) be the amount we are willing to spend in auction \( i \). The state space then becomes

\[
(x, h, t, A) = (x^1, \ldots, x^N, h^1, \ldots, h^N, t^1, \ldots, t^N, A).
\]

By solving a single auction problem using exact dynamic programming, we calculate the quantities \( d^i_t(x^i_t, h^i_t, A, j) \), the expected utility of bidding \( j \), in auction \( i \), with \( t^i \) time remaining and a total budget of \( A \) to spend. Let \( t \) be the current time. We use the decision variables \( u_i(j, t) \), which is equal to one if the agent bids at least \( j \) in auction \( i \) at time \( t \), and zero, otherwise.

The agent solves Problem (5.26) with a slightly modified objective function as follows:
\[
\text{maximize } \sum_{i=1}^{N} \sum_{j=0}^{A_i} u_i(j, t)(d_{x_{it}}(x_{it}^i, h_{it}^i, A, j) - d_{x_{it}}(x_{it}^i, h_{it}^i, A, j - 1)).
\]

This objective accounts for the fact that different auctions need different durations until their completions.

5.5 Summary and conclusions

We have provided an optimal dynamic programming algorithm for the problem of optimally bidding in a single on-line auction. The proposed algorithm was tested in simulation with real data from eBay, and it clearly outperforms in simulation static widely used strategies. We have also used the proposed algorithm to buy over one hundred stamp collections and a Palm Pilots III at attractive prices. The first author has applied the algorithm for a single item in over one thousand auctions for stamps and stamp collections. While it is difficult to assess scientifically the effects, the first author feels the algorithm contributed to (a) increasing the probability of winning and (b) decreasing by 20% the amount paid per win. We have also provided several approximate algorithms when bidding on multiple simultaneous auctions under a common budget. We have found that a method based on combining the value functions of single auctions found by dynamic programming using an integer programming framework produces high quality solutions fast and reliably. The method also extends to the problem of multiple auctions ending at different times.
Chapter 6

Optimal selling in online auctions

We consider the problem of selling a single item within a given period of time in an online auction. We take an empirical, as opposed to game-theoretic, approach in order to determine how a seller can optimally set the parameters he controls when selling a single item or multiple identical items in sequential online auctions. These parameters include an initial price, reserve price, Buy it Now price as well as auction length. The goal of the seller is to maximize his expected utility, possibly by holding multiple sequential auctions, within a fixed time horizon. Our philosophy in constructing the models in this paper is to only use publicly available data. We empirically test our methods using publicly available data from 17,151 auctions from eBay's website, for rare items in the $500 to $2000 range. The computational results in this paper show that (a) the Buy it Now feature adds value for the seller, and (b) it is optimal to set a reserve price in an auction. The model is extended to multi-unit auctions and the results show the benefits of multi-unit auctions over multiple single unit auctions.

Online auctions have set a precedent for e-commerce on the Internet. Their popularity as a means of both selling and buying goods has grown rapidly in the past few years. Among online auctions houses, eBay is the most popular one with over 42 million registered users and more than $9.3 billion worth of sales solely in the year 2001. An attractive feature of an eBay auction is that it is simple for both sellers and buyers to participate in. Furthermore, within its framework an eBay auction allows

the seller to have enough controls in order to affect the final price of the auction (see [3, 26, 28, 54]).

To start an eBay auction, a seller begins by submitting the relevant information on item for sale. This includes a description of the item and often pictures of the item as well. In addition, the seller sets a number of parameters specific to the auction. Two of these are the duration of the auction (3, 5, 7, or 10 days) and the initial price, which is the minimum allowable first bid. Once the auction begins, and after the first bid is submitted, bidders are informed of the listed price, which is the second highest bid to date plus a small increment. The winner is the highest bidder at the end of the auction and he receives the item for a price equal to the second highest bid plus a small increment. There are two important variations of this mechanism:

(a) In a reserve auction, the seller has the option to specify an amount below which he will not sell the item. Throughout the course of the auction, bidders know only if the highest bid is greater than the reserve price. If at the end of the auction no bid is greater than the reserve price, then the item is not sold. If the item sells, then the final price is the maximum between the second highest bid and the reserve price.

(b) The Buy it Now (BIN) feature in an auction allows bidders to immediately win the item by signaling that they are willing to buy the item for a pre-specified price set by the seller. There are two types of Buy it Now auctions, determined by the presence of a reserve price.

(i) If the auction does not have a reserve price, then the Buy it Now feature is only available before the first bid has been placed. That is, either the item is bought immediately for the Buy it Now price, or the first bid is below the Buy it Now price, in which case the auction proceeds without the Buy it Now feature.

(ii) If the auction has a reserve price, then the Buy it Now feature is present until the reserve price has been met. After the reserve price has been met, the auction proceeds as a regular eBay auction.
6.0.1 Literature review

The goal of this chapter is to examine the benefits of choosing the auction parameters (for example, the initial price, the reserve price, the duration of the auction, and whether to add the Buy it Now feature) optimally, putting aside the benefits of good advertising. In Chapter 5 we studied the problem of optimal bidding from the buyer’s perspective.

An eBay auction is based on the second price sealed bid auction designed by Vickrey [47], with some exceptions including the initial price, reserve price, information about previous bids and the hard close time of the auction. For a review of the independent, private-value auction model, see Klemperer [26], Milgrom [32], and McAfee and McMillan [30].

Researchers have recently begun to investigate the use of auctions in the setting of revenue management. The problems addressed are those of a seller wishing to sell a given number of products over a course of a fixed number of auctions. Pinker et al. [38] consider the case where the auction format is a Vickrey Auction and bidders submit their true valuation for the item. The question they address is how to dynamically choose the optimal lot size for each auction, determined before the auction is started. This model is extended to the case where the seller can use the information gained in sequential auctions to learn about the bidders’ valuations. Vulcano et al. [48] consider the allocation, payment scheme and lot size as part of their decisions. An important aspect of this chapter is that the lot size is determined after all bids are received. In using auctions in the context of revenue management, Beam et al. [5] develop a model that does not employ game theory to determine the bidding strategies. Instead, they develop a Markov chain model of bidder behavior and use it to determine the optimal lot-size per auction. Similar to the approach taken in this chapter, they do not account for strategic bidding in sequential auctions by the buyers, as discussed in Oren and Rothkopf [36] from the bidders perspective and Chakraborty et al. [16] from the sellers perspective.

Using data available from the Internet, a number of papers have reported the
effects on the final price by the controls set by a seller. Bajari and Hortacsu [3] find
that a high seller rating and initial price increase the number the number of bidders in
an auction. Lucking-Reiley et al. [28] and Houser and Wooders [25] find that a high
seller rating positively affects final price. [28] also report that initial price, reserve
price and longer auctions all lead to higher prices. Hasker et al. [24] find that longer
auctions have a small effect on final price while Zheng [54] provides evidence that
buyers bid in multiple auctions simultaneously and are paying lower prices for the
item they win. In a field experiment conducted on eBay, Lucking-Reiley and Katkar
[29] find that compared to setting an initial price, having a reserve price of the same
amount results in fewer bids, lower selling price, and a lower probability of selling the
item.

In this chapter we first consider the problem of selling a single item or multiple
items in an eBay auction, within a given time frame. As eBay has proved to be, far
away, the most popular Internet auction house, we consider how one might optimize
over the controls available for an eBay auction. Moreover, we take an empirical,
rather than a game-theoretic, approach to the bidding behavior in response to the
parameters set by the seller.

### 6.0.2 Philosophy and contributions

Our objective in this chapter is to determine the optimal settings for the auction set
by a seller wishing to sell a single item or multiple identical items over a fixed period
of time. In order to explain our modeling choices (see Section 6.1), we require that
the model we build satisfies the following requirements:

(a) It captures the essential characteristics of online auctions.

(b) It leads to a computationally feasible algorithm that is directly usable by sellers.

(c) The parameters for the model can be estimated from publicly available data.

To achieve our goals we have decided to take an empirical, as opposed to a game-
theoretic perspective. The major reason for this is the requirement of having a com-
putationally feasible algorithm that is directly based on data. Throughout the chapter
we make the assumption that bidders do not optimize their bidding strategies over sequential auctions.

We feel that this research makes the following contributions:

1. We propose a model for online auctions that satisfies requirements (a)-(c), mentioned above. The model determines how a seller should optimally set parameters based on a dynamic programming approach.

2. We construct our model using data from 17151 completed auctions for rare items, and use it to show the benefits of a reserve price and a Buy it Now price.

3. We extend our model to multi-unit auctions and provide evidence that multi-unit auctions are beneficial to a seller.

6.0.3 Structure of the chapter

The chapter is structured as follows. In Section 6.1, we present a formulation and a dynamic programming algorithm for selling a single item within a pre-specified number of sequential auctions. In Section 6.2 we consider the effects of choosing the length of the auction. Finally, in Section 6.3, we extend the model to multi-unit auctions. In Section 6.4, we present our conclusions.

6.1 Selling in sequential auctions

6.1.1 The model

The seller is considering up to $T$ auctions for selling a single item. The seller’s goal is to maximize his expected utility by holding sequential auctions and varying the initial price, reserve price and BIN price. If the item is sold at the end of an auction the seller receives his reward. Otherwise, he can re-list the item. If the item is not sold by the end of the auction $T$ then no utility is received. The auctions are indexed by $t = 1, \ldots, T$. 
State

We define the state to be $w_t$, where

$$w_t = \begin{cases} 
1 & \text{if the item has already been sold before auction } t, \\
0 & \text{otherwise.}
\end{cases}$$

Control

If the item has been sold before auction $t$ ($w_t = 1$), then there are no decisions to be made. Otherwise, he may re-list the item and he has available to him the following controls:

$$p_t = \text{initial price},$$
$$r_t = \text{reserve price},$$
$$B_t = \text{Buy it Now price}.$$

We also need the following notation:

$$\rho_t = 1, \text{ if auction } t \text{ has a reserve, and } 0, \text{ otherwise},$$
$$\beta_t = 1, \text{ if auction } t \text{ has a BIN price, and } 0 \text{ otherwise.}$$

The following constraints pertain to the selection of the controls:

$$\rho_t = 0 \Rightarrow r_t \text{ does not exist},$$  \hspace{1cm} (6.1)
$$\rho_t = 1 \Rightarrow r_t \geq p_t + 1,$$  \hspace{1cm} (6.2)
$$\beta_t = 0 \Rightarrow B_t \text{ does not exist},$$  \hspace{1cm} (6.3)
$$\beta_t = 1, \rho_t = 0 \Rightarrow B_t \geq p_t,$$  \hspace{1cm} (6.4)
$$\beta_t = 1, \rho_t = 1 \Rightarrow B_t \geq r_t.$$  \hspace{1cm} (6.5)

Constraint (6.1) states that if the auction is not a reserve auction, then no reserve price exists. The importance of having this constraint is that $\rho_t = 1$ is not equivalent to setting $r_t = 0$, since if there is a reserve price, bidders only know $\rho_t = 0$, and not the value of $r_t$. Constraint (6.2) says that if the auction has a reserve price, then the
reserve price must be at least one unit above the initial price. Constraint (6.3) states that if the auction does not have the BIN feature, then $B_i$ does not exist. Constraint (6.4) states that if there is a BIN price, and there is no reserve price, then the BIN price cannot be less than the initial price. Constraint (6.5) states that if there is a BIN price and a reserve price, then the BIN price cannot be less than the reserve price.

6.1.2 Randomness

The seller is clearly interested in maximizing the final price and the probability the item is sold. The primitive random variables are the number of bidders who bid, and the highest bid of each bidder. We model these as being dependent on the initial price of the auction, the absence or presence of a reserve price, and the BIN price, if it exists. This model avoids imposing major assumptions about bidder behavior, and we believe is useful to a seller interested in the effects of the initial price, reserve price and the BIN feature, at an aggregate level.

The four elements of randomness in the model are:

(a) The number of bidders to place bids for the item, $n_t$. As we will only keep track of the highest bid by a bidder, the number of bidders equals the number of bids. We assume $P(n_t = i | p_t, \rho_t)$ is known and estimated from available data, as described in Section 2.2.

(b) The highest bid, $b_i^t$, placed by each bidder $i = 1, \ldots, n_t$. We assume $P(b_i^t = b | p_t, \rho_t)$ is known and estimated from available data, as described in Section 2.2. We will denote the vector of bids $(b_1^t, \ldots, b_n^t)$ as $b_t$.

(c) The indicator $v_t = 1$ is the event that the auction ended with BIN. If the auction does not have the BIN feature, or if the auction did not end with BIN, then $v_t = 0$.

(d) The final price at the end of the auction, $f_t$. For auctions without the BIN feature, the final price is determined by the following dynamics. For the remainder
of the chapter, we define $\max_2 b_i^t$ to be the second highest among bids $b_t$.

\begin{align*}
  n_t = 0 & \Rightarrow f_t = 0, \quad (6.6) \\
  \rho_t = 0, \quad n_t = 1 & \Rightarrow f_t = p_t, \quad (6.7) \\
  \rho_t = 0, \quad n_t \geq 2 & \Rightarrow f_t = \max_2 b_i^t, \quad (6.8) \\
  \rho_t = 1, \quad n_t \geq 1, \quad \max_i b_i^t < r_t & \Rightarrow f_t = \max_2 b_i^t, \quad (6.9) \\
  \rho_t = 1, \quad n_t \geq 1, \quad \max_i b_i^t \geq r_t & \Rightarrow f_t = \max\{r_t, \max_2 b_i^t\}. \quad (6.10)
\end{align*}

Eq. (6.6) states that if no bidders arrive, then the final price is zero. Eq. (6.7) states that if one bid is made for an auction with no reserve price, then the final price is the initial price of the auction. Eq. (6.8) corresponds to the case where more than two bids are made for an auction without a reserve. In such a case the final price is equal to the second highest bid. Eq. (6.9) corresponds to the case where the highest bid is less than the reserve price in a reserve auction. In such a case the final price is the second highest bid at the end of the auction, but the item is not sold. Eq. (6.10) states that if the highest bid is greater than the reserve price, then the final price is the maximum of the second highest bid and the reserve price. If there is only one bid, and it exceeds the reserve price, then $f_t = r_t$.

For auctions which have the BIN feature, $f_t$ is equal to $B_t$ if $v_t = 1$. Otherwise, if $v_t = 0$ then the final price is determined as if there was no BIN feature, as above.

\subsection{6.1.3 Dynamics}

If $w_t = 1$, then the item has been sold and there are no dynamics to consider. For convenience we will consider that $w_t = 1$ implies that $f_t = 0$. Otherwise, if $w_t = 0$
then the transitions are

\begin{align*}
\rho_t &= 0, \quad \beta_t = 0, \quad n_t = 0 \quad \Rightarrow \quad w_{t+1} = 0, \quad (6.11) \\
\rho_t &= 0, \quad \beta_t = 0, \quad n_t \geq 1 \quad \Rightarrow \quad w_{t+1} = 1, \quad (6.12) \\
\rho_t &= 1, \quad \beta_t = 0, \quad f_t < r_t \quad \Rightarrow \quad w_{t+1} = 0, \quad (6.13) \\
\rho_t &= 1, \quad \beta_t = 0, \quad f_t \geq r_t \quad \Rightarrow \quad w_{t+1} = 1, \quad (6.14) \\
\rho_t &= 0, \quad \beta_t = 1, \quad v_t = 0, \quad n_t = 0 \quad \Rightarrow \quad w_{t+1} = 0, \quad (6.15) \\
\rho_t &= 0, \quad \beta_t = 1, \quad v_t = 0, \quad n_t \geq 1 \quad \Rightarrow \quad w_{t+1} = 1, \quad (6.16) \\
\rho_t &= 0, \quad \beta_t = 1, \quad v_t = 1, \quad n_t = 1 \quad \Rightarrow \quad w_{t+1} = 1, \quad (6.17) \\
\rho_t &= 1, \quad \beta_t = 1, \quad v_t = 0, \quad f_t < r_t \quad \Rightarrow \quad w_{t+1} = 0, \quad (6.18) \\
\rho_t &= 1, \quad \beta_t = 1, \quad v_t = 0, \quad f_t \geq r_t \quad \Rightarrow \quad w_{t+1} = 1, \quad (6.19) \\
\rho_t &= 1, \quad \beta_t = 1, \quad v_t = 1, \quad n_t \geq 1 \quad \Rightarrow \quad w_{t+1} = 1. \quad (6.20)
\end{align*}

Eqs. (6.11) - (6.12) are for auctions with no reserve price and no BIN feature. In such auctions, if the number of bidders is zero then the item does not sell, as in Eq. (6.11). Otherwise, the item sells, as in Eq. (6.12). Equations (6.13) - (6.14) are for auctions with a reserve price but no BIN feature. If the final price is less than the reserve price, as in Eq. (6.13), then the item does not sell. Otherwise, the item sells, as in Eq. (6.14).

Eqs. (6.15)-(6.17) are the dynamics for auctions with the BIN feature, but no reserve price. Eq. (6.15) is for the case where the number of bids is zero. Eq. (6.16) is for the case where the first bid is not for the BIN price. Eq. (6.17) is the case where the first bid is for the BIN price. Note that when \( \rho_t = 0, \beta_t = 1, v_t = 1 \), it is not possible to have \( n_t \neq 1 \). Eqs. (6.18)-(6.20) are the dynamics for auctions with the BIN feature and a reserve price. Eq. (6.18) addresses the case when the BIN feature was not selected and the reserve price was not met. Eq. (6.19) is for the case where the reserve price was met and the BIN feature was never selected. In Eq. (6.20) the BIN feature was selected. Note that it is not possible to have both \( \rho_t = 1, \beta_t = 1, v_t = 1 \) and \( n_t = 0 \).
6.1.4 Objective

The seller seeks to maximize his expected utility

\[
\text{maximize } \mathbb{E}[\sum_{t=1}^{T} U(f_t, w_{t+1})].
\]

In particular, in this research we consider the following utility function

\[
U(f_t, w_{t+1}) = (f_t - A)^+ w_{t+1}.
\]

Quantity \( A \geq 0 \) represents the seller's lower bound of his valuation of the item. Note that at a final price \( f_t = A \), the seller is indifferent between selling the item or not (if he does not plan to hold another auction).

6.1.5 Bellman equation

The problem of maximizing the expected utility in a single item auction can be solved using the Bellman equation:

If \( w_t = 1 \), then \( J_t(w_t) = 0 \) for \( t = 1, \ldots, T \).

If \( w_t = 0 \), then

\[
J_T(w_T) = \max_{p_T, r_T, B_T} \mathbb{E}_{n_T, b_T} [(f_T - A)^+ w_{T+1}]
\]

\[
J_t(w_t) = \max_{p_t, r_t, b_t} \mathbb{E}_{n_t, b_t} [(f_t - A)^+ w_{t+1} + (1 - w_{t+1}) J_{t+1}(w_{t+1})]
\]

\[
= \max_{p_t, r_t, B_t} \left\{ (B_t - A)^+ P(v_t = 1|p_t, \rho_t, B_t) + P(v_t = 0|p_t, \rho_t, B_t) \cdot \left( \sum_{n \geq 1} \sum_{\{b_t|f_t \geq \rho_t, \beta_t\}} (f_t - A)^+ P(b_t^1|p_t, \rho_t) \cdots P(b_t^n|p_t, \rho_t) \right) P(n_t = n|p_t, \rho_t, \beta_t) \right. \\
+ \left. J_{t+1}(0) P(b_t^1|p_t, \rho_t) \cdots P(b_t^n|p_t, \rho_t) \right) \right\}.
\]
Recall that \( f_t \) is a function of \( b_1^t, \ldots, b_n^t, p_t \) and \( r_t \). In the case where there is no reserve, the set \( \{ b_t | f_t < p_t r_t \} \) is empty and thus contributes nothing to the overall calculation of \( J_t(w_t) \), as desired. When there is no BIN feature, \( \beta_t = 0 \) and \( P(v_t = 0 | p_t, \rho_t) = 1 \).

### 6.1.6 Estimation of parameters

One important aspect of our model is that it relies solely on publicly available data from eBay. Using the available data, we have calculated the empirical bidding distribution for rare items whose final selling price was between $500 and $2000. By rare items we mean stamps, antiques and other such items which cannot be readily bought elsewhere. All auctions were single item eBay auctions with a hard closing time. eBay reports the amount of all received bids, except for the highest bid for which the smallest required bid to win is reported. We used this amount. We collected data from 17,151 auctions from May 2002 to January 2003. Of these, 8,804 received bids for a total of 53,931 bids per unique user. 6,596 auctions had a reserve price, 934 ended with the Buy It Now feature, and 4,088 were Buy It Now auctions that received zero bids. The average final price of the auctions was $821.

eBay publishes the history of auctions completed in the past two weeks, thus the distribution of \( n_t \) dependent on \( p_t \) and \( \rho_t \) are readily available. For a given combination of \( (p_t, \rho_t) \) we calculate the empirical distribution of \( n_t \) and bids \( b_t \). To reduce the size of the estimation problem, and to eliminate having to deal with extremely sparse distribution matrices, we round up bids \( b_t^i \) to \( \lceil b_t^i / 100 \rceil \). For example, an observed bid of $610 is counted as \( b = 7 \). One caveat is that eBay publishes the amount of all received bids, except for the highest bid for which the smallest required bid to win is reported. We used the reported bid amount and not what the bid might have been.

For a given initial price \( p \), reserve price indicator \( \rho \) and BIN price \( B \), from the 17,151 auctions in our data set let

\[
N_B = \text{the number of auctions that ended with BIN,}
\]
\[
N_Z = \text{the number of BIN auctions that ended with zero bids.}
\]
Both $N_B$ and $N_Z$ are available from the data, for if an auction had the BIN feature, then it reports if the auction ended with BIN, or it reports if zero bids arrived. However, the data available do not report the number of auctions which began as BIN auctions, but ended as regular auctions, $N_R$. If $N_R$ were available, then it would follow that

$$N_R + N_B + N_Z = \text{the number of auctions that are BIN auctions} = N_{BIN}. \quad (6.21)$$

As we have no recording of $N_R$, we have chosen to approximate it with $aN_R$, where $a \geq 0$. In the model we consider in this chapter, we have set $a = 3$, so that $P(v_t = 1 | p_t, \rho_t, B_t)$ is equal to

$$P(v_t = 1 | p_t, \rho_t, B_t) = \frac{N_B}{(a + 1)N_B + N_Z} = \frac{N_B}{4N_B + N_Z}.$$

### 6.1.7 Empirical results

We consider a seller interested in selling an item with $T = 4$ periods for the cases $A = 0$ and $A = 1000$. The utility function we used is $U(f, w) = (100f - A)^+ w$. Note that $f$ is a measure of hundreds of dollars, so that $f = 10$ represents a final price of $901 - 1000$. For both values of $A$ we find the optimal policy when the BIN feature is and is not available. Tables 1-4 show these results. Column ‘Stage’ contains the variable $t$ for values 1 through 4, ‘Expected Utility’ is a column with the cost to go, $J_t(1)$, ‘Prob. of Immediate Sale’ contains the probability the item is sold in that auction and column ‘Expected Price’ is the expected final price of the auction given that the item is sold in that auction.

Tables 6.1 and 6.2 show the results for selling an item over four auctions with and without the BIN feature, respectively, for $A = 0$. Tables 6.3 and 6.4 show the results for selling an item over four auctions with and without the BIN feature, respectively, for $A = 1000$. In Tables 6.1 and 6.3, we see it is optimal to keep the initial price
### Table 6.1: Final sale price using reserve and initial price.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>1400</td>
<td>1071</td>
<td>0.21</td>
<td>409</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>1200</td>
<td>982</td>
<td>0.32</td>
<td>1224</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>1200</td>
<td>866</td>
<td>0.32</td>
<td>224</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>500</td>
<td>695</td>
<td>0.97</td>
<td>720</td>
</tr>
</tbody>
</table>

### Table 6.2: Final sale price using reserve, initial price and BIN.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Buy Now Price</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>1800</td>
<td>1800</td>
<td>1515</td>
<td>0.30</td>
<td>1800</td>
</tr>
<tr>
<td>2</td>
<td>400</td>
<td>1500</td>
<td>1800</td>
<td>1392</td>
<td>0.36</td>
<td>1698</td>
</tr>
<tr>
<td>3</td>
<td>400</td>
<td>1400</td>
<td>1800</td>
<td>1218</td>
<td>0.39</td>
<td>1651</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>500</td>
<td>1800</td>
<td>938</td>
<td>0.96</td>
<td>977</td>
</tr>
</tbody>
</table>

### Table 6.3: Final sale price using reserve and initial price.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1500</td>
<td>260</td>
<td>0.16</td>
<td>1505</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1500</td>
<td>212</td>
<td>0.16</td>
<td>1505</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1500</td>
<td>155</td>
<td>0.16</td>
<td>1505</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>1400</td>
<td>86</td>
<td>0.21</td>
<td>1409</td>
</tr>
</tbody>
</table>

### Table 6.4: Final sale price using reserve, initial price and BIN.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Buy Now Price</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>400</td>
<td>1800</td>
<td>1800</td>
<td>614</td>
<td>0.30</td>
<td>1800</td>
</tr>
<tr>
<td>2</td>
<td>400</td>
<td>1800</td>
<td>1800</td>
<td>536</td>
<td>0.30</td>
<td>1800</td>
</tr>
<tr>
<td>3</td>
<td>400</td>
<td>1800</td>
<td>1800</td>
<td>423</td>
<td>0.30</td>
<td>1800</td>
</tr>
<tr>
<td>4</td>
<td>400</td>
<td>1400</td>
<td>1800</td>
<td>259</td>
<td>0.39</td>
<td>1660</td>
</tr>
</tbody>
</table>
constantly low, either $0 or $100. Similarly for Tables 6.2 and 6.4, the initial price is kept constant at $400. The other interesting observation is the sharp increase in the reserve price between stages 3 and 4 for both Tables 6.1 and 6.2. The reason for this is the already high expect final price when the initial price is set to $100.

The results of these tables suggest the following:

(1) The BIN feature adds value. Moreover, it is optimal to set it to a level well above the expected sale price.

(2) It is optimal for the seller to increase the reserve price as the number of auctions remaining increases.

(3) It is optimal to keep the initial constant for each auction.

The final point can be restated as: It is optimal to find an initial price that leads to a with a high variance and expected sale price, and adjust the reserve price alone.

These results also demonstrate, as one would expect, that there are significant advantages to holding multiple auctions. Consider, for example, the expected utility in Stages 1 and 4 from Table 6.1 in which the expected utilities are 1075 and 695, respectively. There are also some subtle strategies that the algorithm suggests. For example, in Table 6.2 we see that the initial price decreases to zero from $100 when there are more two or more auctions remaining. The reason for this is that probability of having high bids was greater for $p = 0$ than for $p = 100$, thus making the high reserve price for $p = 0$ beneficial. We also see, as expected, that as the number of auctions remaining increases, the seller should adjust the controls to increase the expected sale price, given a sale, and thereby decrease the probability of selling the item in that auction.
6.2 Selling in a fixed time horizon

6.2.1 The model

A more common problem a seller faces is to sell an item over a given period of time, rather than a given number of auctions. Given up to \( D \) days for selling an item, the question for the seller, in addition to optimally setting parameters \( p_t, r_t \) and \( B_t \), is how to optimally set the duration of each auction, \( d_t \). Auctions are indexed by \( t = 1, \ldots, T \), so that \( t \), as before, counts the auctions to date. Note that an upper bound on the value of \( T \) is \([D/3]\).

State

The state is \((s_t, w_t)\), for \( t = 1, \ldots, T \), where

\[
\begin{align*}
s_t &= \text{days remaining to sell the item at the start of auction } t, \\
w_t &= 1 \text{ if the item has been sold by auction } t, \text{ 0 otherwise.}
\end{align*}
\]

Control

The question for the seller is, given \( s_t \) days remaining to sell the item, how to optimally choose, in addition to parameters \( p_t, r_t \) and \( B_t \), the length \( d_t \) of the auction. In addition to the usual constraints on the controls, we set \( d_t \leq s_t \), where \( d_t \in \{3, 5, 7, 10\} \) days.

Randomness

By selecting \( p_t, r_t, B_t \), and auction length \( d_t \) with \( s_t \) days remaining in the time horizon in auction \( t \), the seller is interested in affecting the final price \( f_t \) and whether or not the item is sold, \( w_{t+1} \). As before, the primitive random variables we are concerned with are the number of bidders and what each bidder bids. The key difference is that the random variables \( n_t \) and \( b_t \) are now dependent on the length of the auction as well as the previous parameters. The elements \( f_t \) and \( w_{t+1} \) functions of the following
random variables:

\[ n_t = \text{the number of bidders that arrive in an auction of length } d_t, \]
\[ b_i^t = \text{the bid placed by the bidder } i \text{ during auction of length } d_t, \ i = 1, \ldots, n_t. \]

We need to determine \( P(n_t = n|p_t, \rho_t, \beta_t, d_t) \) and \( P(b_i^t = b|p_t, \rho_t, d_t) \). In addition, if the auction has the BIN feature, then we are also interested in \( P(v_t = 1|p_t, \rho_t, B_t) \). Unfortunately, for auctions that end with the Buy It Now, we are unable to determine the length of the auction set by the seller. Thus, we have considered a distribution of \( v_t \) that is independent of the auction length. We calculate the distributions of \( n_t, b_i \) and \( v_t \) as we describe in Section 2.2. In this case, we also add the dependence on the length of the auction for \( n_t \) and \( b_i \).

Ultimately, the seller is interested in the distribution of the final price and whether or not the item was sold. The dynamics of \( f_t \) are the same as in Section 6.1.

**Dynamics**

As the seller can control an auction only at the beginning, the dynamics occur over a length \( d_t \). The equations governing the dynamics of \( w_t \) are as described in Section 6.1. The dynamics for state variable \( s_t \) are

\[ s_{t+1} = \max\{s_t - d_t, 0\}, \quad (6.22) \]

where the maximization ensures we do not spend more days attempting to sell the item than what we have available.

**Objective**

As before, the objective of the seller is to maximize the quantity

\[ \max \mathbb{E}\left[\sum_{t=1}^{T} U(f_t, w_{t+1})\right]. \]
and we will focus on the utility function

\[ U(f_t, w_{t+1}) = (f_t - A)^+ w_{t+1}. \]

**Bellman equation**

The problem of maximizing the expected utility in a single item auction can be solved using the Bellman equation:

If \( w_t = 1 \) or if \( s_t = 0 \) then \( J_t(s_t, w_t) = 0 \).

If \( w_t = 0 \) then

\[
J_T(s_T, w_T) = \max_{p_T, r_T, B_T, d_T} E_{w_T, B_T, d_T} [(f_T - A)^+ w_{T+1}]
\]

\[
J_t(s_t, w_t) = \max_{p_t, r_t, B_t, d_t} E_{w_t, B_t, d_t} [(f_t - A)^+ w_{t+1} + (1 - w_{t+1}) J_{t+1}(s_{t+1}, w_{t+1})]
\]

\[
= \max_{p_t, r_t, B_t, d_t} \left\{ (B_t - A)^+ P(v_t = 1|p_t, \rho_t, B_t) + P(v_t = 0|p_t, \rho_t, B_t) \cdot \left( \sum_{n \geq 1} \left( \sum_{(b_t|f_t \geq \rho_t)} (f_t - A)^+ P(b_t^1|p_t, \rho_t, d_t) \cdots P(b_t^n|p_t, \rho_t, d_t) \right) \right) \right. \\
+ \sum_{(b_t|f_t < \rho_t)} J_{t+1}(\max\{s_t - d_t, 0\}, 0) P(b_t^1|p_t, \rho_t, d_t) \cdots P(b_t^n|p_t, \rho_t, d_t) \right) \right. \\
\left. \cdot P(n_t = n|p_t, \rho_t, \beta_t, d_t) \right) \left. + J_{t+1}(\max\{s_t - d_t, 0\}, 0) P(n_t = 0|p_t, \rho_t, \beta_t, d_t) \right) \}.
\]

The selling period begins with \( t = 1 \), \( s_1 = D \) and \( w_1 = 0 \).

**6.2.2 Estimation of parameters**

We used the same data set as in Section 6.1. The difference is that we consider a distribution for \( n_t \) and \( b_t^i \) that depends on \( p_t, \rho_t \) and \( d_t \). One important issue to point out regarding the data is that it is not possible to find the intended length of the auction, set by the seller, if the auction ended with BIN. The reason for this is that the length of the auction is normally calculated by subtracting the starting date from


<table>
<thead>
<tr>
<th>Days To Go</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Length (days)</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>1319</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>13</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>1306</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>12</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>1250</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>11</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>1250</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1120</td>
<td>0.58</td>
<td>1200</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1120</td>
<td>0.58</td>
<td>1200</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>1300</td>
<td>3</td>
<td>1099</td>
<td>0.49</td>
<td>1474</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1008</td>
<td>0.58</td>
<td>1203</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1008</td>
<td>0.58</td>
<td>1203</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>739</td>
<td>0.64</td>
<td>1154</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>739</td>
<td>0.64</td>
<td>1154</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>739</td>
<td>0.64</td>
<td>1154</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.5: The optimal parameters and selling price for selling an item over a 14 day period.

the ending date of the auction. When an auction ends with BIN, the ending date becomes the date the item was sold, and hence is no longer the intended number of days after the start date. For this reason we have made the assumption that the probability of an auction ending because of BIN, is independent of the auction length.

6.2.3 Empirical results

We first consider the case of a seller interested in selling a single item within $D = 14$ days, by only adjusting the initial and reserve price. We then consider the addition of the BIN feature. The results are shown in Tables 5-8. Column ‘Days Remaining’ is the state $s_t$ for some $t$. All other columns are described in Section 6.1.7.

Tables 5 and 7 show the results without using the BIN feature for $A = 0$, 1000 respectively. For both instances, it is optimal to (i) always set the initial price equal to $1000; (ii)$ decrease the reserve price in consecutive auctions; (iii) hold short auctions of 3 or 5 days. Tables 6 and 8 show when allowing for the use of the BIN feature for $A = 0$, 1000 respectively. These results suggest that (i) it is optimal to keep the
### A = 0, With Buy It Now

<table>
<thead>
<tr>
<th>Days To Go</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Buy It Now Price</th>
<th>Length (days)</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>500</td>
<td>1700</td>
<td>1700</td>
<td>3</td>
<td>1631</td>
<td>0.28</td>
<td>1700</td>
</tr>
<tr>
<td>13</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1604</td>
<td>0.44</td>
<td>1700</td>
</tr>
<tr>
<td>12</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1604</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>11</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1556</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1556</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>9</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1556</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>8</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>1556</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>1200</td>
<td>1500</td>
<td>5</td>
<td>1469</td>
<td>0.95</td>
<td>1483</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>1200</td>
<td>1500</td>
<td>5</td>
<td>1469</td>
<td>0.95</td>
<td>1483</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>1200</td>
<td>1500</td>
<td>5</td>
<td>1469</td>
<td>0.95</td>
<td>1483</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>1500</td>
<td>3</td>
<td>1239</td>
<td>0.97</td>
<td>1277</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>1500</td>
<td>3</td>
<td>1239</td>
<td>0.97</td>
<td>1277</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.6: The optimal parameters and selling price for selling an item over a 14 day period.

### A = 1000, Without Buy It Now

<table>
<thead>
<tr>
<th>Days To Go</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Length (days)</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>451</td>
<td>0.28</td>
<td>1800</td>
</tr>
<tr>
<td>13</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>451</td>
<td>0.28</td>
<td>1800</td>
</tr>
<tr>
<td>12</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>399</td>
<td>0.28</td>
<td>1800</td>
</tr>
<tr>
<td>11</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>399</td>
<td>0.28</td>
<td>1800</td>
</tr>
<tr>
<td>10</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>399</td>
<td>0.28</td>
<td>1800</td>
</tr>
<tr>
<td>9</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>317</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>317</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>245</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>245</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>245</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>116</td>
<td>0.58</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>116</td>
<td>0.58</td>
<td>1200</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.7: The optimal parameters and selling price for selling an item over a 14 day period.
<table>
<thead>
<tr>
<th>Days To Go</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Buy It Now Price</th>
<th>Length (days)</th>
<th>Expected Utility</th>
<th>Prob. of Immediate Sale</th>
<th>Expected Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>500</td>
<td>1700</td>
<td>1700</td>
<td>3</td>
<td>636</td>
<td>0.28</td>
<td>1700</td>
</tr>
<tr>
<td>13</td>
<td>500</td>
<td>1700</td>
<td>1700</td>
<td>3</td>
<td>610</td>
<td>0.44</td>
<td>1700</td>
</tr>
<tr>
<td>12</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>610</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>11</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>610</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>567</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>9</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>567</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>8</td>
<td>500</td>
<td>1600</td>
<td>1700</td>
<td>3</td>
<td>567</td>
<td>0.44</td>
<td>1664</td>
</tr>
<tr>
<td>7</td>
<td>1000</td>
<td>1500</td>
<td>1500</td>
<td>5</td>
<td>490</td>
<td>0.88</td>
<td>1557</td>
</tr>
<tr>
<td>6</td>
<td>1000</td>
<td>1500</td>
<td>1500</td>
<td>5</td>
<td>490</td>
<td>0.88</td>
<td>1557</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>1500</td>
<td>1500</td>
<td>5</td>
<td>490</td>
<td>0.88</td>
<td>1557</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>1600</td>
<td>1600</td>
<td>3</td>
<td>311</td>
<td>0.52</td>
<td>1600</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>1600</td>
<td>1600</td>
<td>3</td>
<td>311</td>
<td>0.52</td>
<td>1600</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.8: The optimal parameters and selling price for selling an item over a 14 day period.

initial price set at around $1000 (the decrease to $900 is to increase the variance in the final price); (ii) decrease the reserve price in consecutive auctions; (iii) hold short auctions of 3 or 5 days. One interesting observation is of the strategy used when there are 3 and 4 days remaining to sell the item when $A = 1000$ in Table 6.8. Here the algorithm suggests starting with an initial price of $500$, and setting the the reserve price equal to the BIN price. The results of these tables lead us to make the following observations:

(1) The BIN feature adds value to the auction.

(2) In general, it is better to have short auctions as opposed to long ones.

(3) With more days remaining it is optimal to be aggressive with the the reserve price, the decreasing the probability of selling the item, but increasing the expected price given it is sold.

Qualitatively the algorithm suggests similar findings for the reserve price and BIN price as in Section 6.1.7, however the optimal initial prices are strikingly different. The differences can best be explained by the effects of auction length on the final
price; the benefits of having multiple short auctions with high reserves and high initial prices outweigh the exposure and large bidder population provided by a few long auctions with low initial prices. However, when the auction length is ignored, as in Section 6.1.7, this advantage no longer exists.

6.3 Multiple identical items in a fixed time interval

6.3.1 The model

Another option a seller has in an eBay auction is to sell multiple identical items in a given auction. There is evidence to suggest that bidders strategically bid in the auction of lowest listed price when the same good is being sold in simultaneous auctions, Zheng [54]. As in previous sections we will consider only sequential auctions. Given $D$ days during which the seller would like to sell $K$ identical items in sequential, non-overlapping auctions in which each bidder places a bid for a single item. Such auction formats are available for sellers to use on eBay, and the controls for such auctions that are common to single unit auctions are the initial price, the reserve price and the length of the auction. As there is no BIN feature for multi-unit auctions we cannot include the BIN feature in our model. The question for the seller, in addition to how to set $\rho_t, r_t$ and $d_t$, is the number of items, $k_t$, to make available for sale in each auction. Again, $t$ indicates the auction number.

State

The state is $(s_t, K_t)$, for $t = 1, ..., T$, where

\[
\begin{align*}
    s_t &= \text{days remaining to sell the item at the start of auction } t, \\
    K_t &= \text{the number items left to sell by auction } t.
\end{align*}
\]

We set $K_1 = K$, the initial number of items to sell.

173
Control

The controls are $p_t, r_t, d_t$ and $k_t$, the number of items the seller is willing to sell. The constraints on the controls are the same as in Section 2, with the addition constraint that $k_t \leq K_t$.

Randomness

As before, the primitive random variables are $n_t$ and $b_t$. We are now interested in determining the final price $f_t$ and the number of items sold in the auction $t$, $m_t$.

The transitions are variations of the following simple case introduced by Vickrey [47]: With $k$ items for sale, and $n > k$ bids made, the final price is equal to the $(k+1)$ highest bid and $k$ items are sold for this price. The following equations describe the dynamics which are dependent on $n_t$ and $r_t$. We denote the $i + 1 \leq n$ highest bid by $b_t^{(i+1)}$, and we let $G_t(r_t)$ denote the number of bids greater than the reserve $r_t$.

\begin{align}
\rho_t = 0, \quad n_t = 0 \quad &\Rightarrow f_t = 0, \quad m_t = 0, \quad (6.23) \\
\rho_t = 0, \quad 1 \leq n_t \leq k_t \quad &\Rightarrow f_t = p_t, \quad m_t = n_t, \quad (6.24) \\
\rho_t = 0, \quad n_t > k_t \quad &\Rightarrow f_t = b_t^{(k_t+1)}, \quad m_t = k_t, \quad (6.25) \\
\rho_t = 1, \quad G_t(r_t) = 0 \quad &\Rightarrow f_t = 0, \quad m_t = 0, \quad (6.26) \\
\rho_t = 1, \quad 1 \leq G_t(r_t) \leq k_t \quad &\Rightarrow f_t = r_t, \quad m_t = G_t(r_t), \quad (6.27) \\
\rho_t = 1, \quad k_t + 1 \leq G_t(r_t) \leq n_t \quad &\Rightarrow f_t = \max\{b_t^{(k_t+1)}, r_t\}, \quad m_t = k_t. \quad (6.28)
\end{align}

Eqs. (6.23)-(6.25) are the dynamics if there is no reserve price. Eq. (6.23) states that if no bids arrive then the final price is zero and no items are sold. Eq. (6.24) applies to the case where the number of bids is not greater than the items up for auction. In this case the final price is equal to the initial price and the number of items sold is equal to the number of bids. For the case that the number of bids is greater than the items for sale, Eq. (6.25) states that the final price is equal to the $k_t+1$ highest bid, and all $k_t$ items are sold for this price.
Eqs. (6.26)-(6.28) apply to the case when there is a reserve price. Eq. (6.26) states that if all bids, if there are any, are less than the reserve, then the final price is equal to the $k_t + 1$ highest bid, but no items are sold. Eq. (6.27) states that if there are not more than $k_t$ bids greater than or equal to the reserve, then the final price is equal to the reserve price and $G_t(r_t)$ units are sold. For the case that there are more than $k_t$ bids not less than the reserve, Eq. (6.28) states that $k_t$ bidders pay the final price, which is the maximum of the reserve and the $(k_t + 1)$ highest bid.

**Dynamics**

There is not a BIN feature for multi-unit auctions on eBay, and the controls for the seller are here limited to $p_t, r_t, d_t$ and $k_t$. The dynamics for $s_t$ are as in Section 6.2. For $K_t$ the dynamics are as follows:

\begin{align*}
\rho_t = 0, \quad n_t = 0 & \Rightarrow \quad K_{t+1} = K_t, \\
\rho_t = 0, \quad n_t \geq 1 & \Rightarrow \quad K_{t+1} = K_t - m_t, \\
\rho_t = 1, \quad f_t < r_t & \Rightarrow \quad K_{t+1} = K_t, \\
\rho_t = 1, \quad f_t \geq r_t & \Rightarrow \quad K_{t+1} = K_t - m_t.
\end{align*}

**Objective**

The objective of the seller is to maximize the expected utility

\[
\text{maximize } E[\sum_{t=1}^{T} U(f_t, m_t)].
\]

As before, we will focus on the utility function

\[
U(f_t, m_t) = (f_t - A)^+ m_t.
\]
Bellman equation

The problem of maximizing the expected utility in a series of sequential single item auctions can be solved using the Bellman equation. As shorthand we will let \( G_t(r) = n_t \) when \( r_t = 0 \).

If \( K_t = 0 \), or \( s_t = 0 \) then \( J(s_t, K_t) = 0 \). Otherwise,

\[
J_T(s_T, K_T) = \max_{r_T, r_T, d_T, k_T} E_{n_T, b_T} [(f_T - A)^+ m_T],
\]

\[
J_t(s_t, K_t) = \max_{r_t, r_t, d_t, k_t} E_{n_t, b_t} [(f_t - A)^+ m_T + J_{t+1}(s_{t+1}, K_{t+1})]
\]

\[
= \max_{r_t, r_t, d_t, k_t} \left\{ \sum_{n \geq 0} \left( \sum_{b_t} \left( (f_t - A)^+ m_t + J_{t+1}(\max\{s_t - d_t\}, K_t - m_t) \right) \right) \right\}.
\]

Here \( f_t \) and \( m_t \) are functions of \( n_t, b_t, x_t \) and \( k_t \).

6.3.2 Estimation of parameters

The data set is the same as in Section 6.2.2. Note that we assume the bids \( b_t \), and the number of bids \( n_t \) to be independent of the number of items for sale, \( k_t \).

6.3.3 Empirical results

We consider the case of a seller interested in selling multiple units of an identical item within a 21 day period and \( A = 0 \). The seller is allowed to adjust the length of the auction, initial price, the reserve price and the number of units for sale. Table 9 shows the optimal parameters as derived from our model. From this table we draw the following conclusions:

1. It is optimal to put most, if not all, of the items up for auction, regardless of the time remaining. A consequence of this is that it is optimal to hold sequential
multi-unit auctions rather than sequential single unit auctions to sell identical goods.

(2) The optimal initial price is constant at $1000 and the reserve price monotonically increases with the number of days remaining.

(3) For the same number of days remaining, the more goods to be sold the lower the reserve price should be.

(4) With more days remaining to sell a given number of units, the optimal policy results in a decrease in the expected number of sold items, but an increase in the expected final price given an item is sold.

6.4 Summary and conclusions

In this chapter, we have taken a dynamic programming approach in order to assist sellers to increase their revenue when selling an item over a fixed time interval on eBay. The dynamic programming approach we considered utilized real data from eBay. It allowed us to produce strong evidence for the merits of setting an appropriate reserve price, using the BIN feature, and setting the length of the auction to the seller's advantage. We have also demonstrated the gains to be had by re-listing an item. Furthermore, we have considered the problem of selling multiple identical items over a fixed time interval, and provided evidence for the benefits of properly selecting the lot size and the length of the auction.
<table>
<thead>
<tr>
<th>Days to Go</th>
<th>Items to Sell</th>
<th>Initial Price</th>
<th>Reserve Price</th>
<th>Length (days)</th>
<th>Items for Sale</th>
<th>Exp. Utility</th>
<th>Exp. Number Sold</th>
<th>Exp. Sale Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>1</td>
<td>1000</td>
<td>1800</td>
<td>5</td>
<td>1</td>
<td>1513</td>
<td>0.27</td>
<td>1800</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>2</td>
<td>2762</td>
<td>0.56</td>
<td>1609</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>3</td>
<td>3847</td>
<td>0.61</td>
<td>1601</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1300</td>
<td>5</td>
<td>3</td>
<td>4820</td>
<td>1.29</td>
<td>1309</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>1</td>
<td>1319</td>
<td>0.38</td>
<td>1643</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1600</td>
<td>5</td>
<td>2</td>
<td>2438</td>
<td>0.56</td>
<td>1609</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1300</td>
<td>5</td>
<td>3</td>
<td>3372</td>
<td>1.29</td>
<td>1309</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1300</td>
<td>5</td>
<td>4</td>
<td>4238</td>
<td>1.46</td>
<td>1300</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1</td>
<td>1120</td>
<td>0.58</td>
<td>1200</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>2</td>
<td>2114</td>
<td>0.91</td>
<td>1200</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>3</td>
<td>2908</td>
<td>1</td>
<td>1200</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>4</td>
<td>3540</td>
<td>1.33</td>
<td>1100</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1000</td>
<td>1200</td>
<td>3</td>
<td>1</td>
<td>1008</td>
<td>0.58</td>
<td>1203</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>2</td>
<td>1828</td>
<td>1.13</td>
<td>1114</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>3</td>
<td>2398</td>
<td>1.33</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>4</td>
<td>2730</td>
<td>1.33</td>
<td>1100</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>1</td>
<td>739</td>
<td>0.64</td>
<td>1154</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1200</td>
<td>5</td>
<td>2</td>
<td>1307</td>
<td>0.99</td>
<td>1317</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>5</td>
<td>3</td>
<td>1804</td>
<td>1.49</td>
<td>1205</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>5</td>
<td>4</td>
<td>2222</td>
<td>1.96</td>
<td>1131</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>1</td>
<td>739</td>
<td>0.64</td>
<td>1154</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>2</td>
<td>1266</td>
<td>1.13</td>
<td>1114</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>3</td>
<td>1466</td>
<td>1.33</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1000</td>
<td>1100</td>
<td>3</td>
<td>4</td>
<td>1466</td>
<td>1.33</td>
<td>1100</td>
</tr>
</tbody>
</table>

Table 6.9: The optimal parameters and selling price for selling 1,2,4 and 8 items over a 21 day period.
Chapter 7

Conclusions

In this dissertation we have studied the application of Lagrange multipliers to weakly coupled dynamic optimization problems. The main contributions of this dissertation are:

(a) We have presented a Lagrangian based approach to decoupling weakly coupled dynamic optimization problems for both finite and infinite horizon problems. We have shown how to use the decoupled problems to obtain bounds on the true value of the problem, and presented two methods for computing the bounds. In the applications considered, we were able to obtain bounds on the true value of intractable problems.

(b) We have presented three Lagrange-based policies for weakly coupled dynamic optimization problems. For the multiarmed bandit problem, one of these policies was optimal. For the remaining applications the policies performed very well.

(c) We study the implications of our policies in multiarmed bandits, restless bandits, supply chain inventory problems and capacitated queueing networks.

(d) We extend our results to non-decomposable problems and investigate our Lagrangian techniques in constrained quadratic-cost linear systems. In this case we were able to achieve quality bounds on good policies for otherwise intractable
problems.

(e) We have proposed two models of on-line auctions, one from a bidder’s perspective and the other from a seller’s perspective, that allow for dynamic optimization by the user. These models are tractable, directly applicable by the user, and the parameters of the models can be constructed from publicly available data.

(f) For the bidder’s problem, we have provided several algorithms, based on approximate dynamic programming and integer programming algorithms, to find optimal bidding strategies in multiple simultaneous on-line auctions. The strongest of these methods is based on our Lagrangian-based decoupling heuristic. We provide computational evidence that these methods produce high quality solutions fast and reliably.

(g) For the seller’s problem in on-line auctions, we use our model of the dynamics of the auction in the context of dynamic optimization. We investigate the effectiveness of the reserve price, Buy it Now price and auction length. Our algorithm also provides a seller the means for determining the optimal number of items to put up for sale in sequential multi-unit auctions.

In this dissertation we have taken a simple approach to decoupling constrained dynamic optimization problems and investigated its usefulness in a number of applications. From this dissertation it should be clear that there is an exciting opportunity for further research in the area of Lagrangian methods in dynamic optimization problems.
Appendix A

Counterexample of monotonicity of $D(\lambda)$

Consider the Markovian reward process in illustrated in Fig. A-1. From Fig. A-1 it can be seen that

\[
L(1; \lambda) = \max\{1 + \beta L(3; \lambda), 8 - \lambda + \beta L(2; \lambda)\}
\]

\[
L(2; \lambda) = \max\{\frac{5 - \lambda}{1 - \beta}, 0\}
\]

\[
L(3; \lambda) = \max\{\frac{6 - \lambda}{1 - \beta}, 0\}.
\]

For $\beta = 3/4$ the optimal decision when in state 1, given $\lambda$ is summarized in Table A.

Table A shows that additional stopping criteria will be needed to ensure some sort

![Diagram](image)

Figure A-1: All transition probabilities given a particle decision are one. Solid lines correspond to active transitions, and dashed lines correspond to passive transitions. Rewards appear next to transitions.
of convergence for restless bandits. In addition, we now have no guarantee that at least $K$ bandits will have $D_t(\lambda) = 1$, thus a method for choosing which bandits to play is needed, which we call RANK.
Appendix B

Calculation of $J^D(x)$

$J^D_0(x_0)$ is the solution to the deterministic minimization problem

$$J^D_0(x_0) = \min_{u_0,\ldots,u_{T-1}} x_T'Q_Tx_T + \sum_{t=0}^{T-1} (x_t'Q_t x_t + u_t'R_t u_t),$$

where

$$x_{t+1} = A_t x_t + B_t u_t.$$ 

The approach will be to solve $u_0,\ldots,u_{T-1}$ simultaneously. Expressing $x_t$ as a function of $x_{t-1}$ and $u_{t-1}$ we have

$$x_{t+1} = A_t A_{t-1} x_{t-1} + A_t B_{t-1} u_{t-1} + B_t u_t.$$ 

Define

$$D_{t,t} = A_t \cdots A_i, \quad 0 \leq i < t$$

$$D_{t,t+1} = I.$$ 

183
We can then express $x_{t+1}$ in terms of all prior conditions and the starting point $x_0$ by

$$x_{t+1} = \sum_{n=0}^{t} D_{t,n+1} B_n u_t + D_{t,0} x_0.$$  

It follows that

$$J_0^D(x_0) = \min_{u_0, \ldots, u_T} x_0' Q_0 x_0 + \sum_{t=1}^{T} \left( \sum_{n=0}^{t-1} D_{t-1,n+1} B_n u_n + D_{t-1,0} x_0 \right)' Q_t$$

$$\cdot \left( \sum_{m=0}^{t-1} D_{t-1,m+1} B_m u_m + D_{t-1,0} x_0 \right)$$

s.t. $F_t u_t \leq g_t$, $t = 0, \ldots, T - 1$.

Consequently, it can be seen that $J_0^D(x_0)$ is the solution to a quadratic minimization problem with linear constraints.
Appendix C

Proof of Theorem 13

The proof of Theorem 13 is below.

**Proof.** The case for \( t = T - 1 \) has been proved. We will assume the lemma true for \( t + 1 \) and show it is true for stage \( t \).

\[
L_t(x_t; \lambda) = \min_{u_t} \mathbb{E}\left[ x_t'Q_t x_t + u_t' R_t u_t + L_{t+1}(A_t x_t + B_t u_t + w_t, \lambda) + 2 \lambda'(F_t u_t - g_t) \right]
\]

\[
= \min_{u_t} \left\{ \mathbb{E}\left[ x_t'Q_t x_t + u_t' R_t u_t + (A_t x_t + B_t u_t + w_t)' K_{t+1}(A_t x_t + B_t u_t + w_t) \right. \right.
\]

\[
+ 2 \lambda' \left( S_{t+1}(A_t x_t + B_t u_t + w_t) - \sum_{s=t+1}^{T-1} g_s \right) \left. \right] - \lambda' V_{t+1} \lambda
\]

\[
+ 2 \lambda'(F_t u_t - g_t) + \sum_{s=t+1}^{T-1} \mathbb{E}[w_s' K_{s+1} w_s] \}
\]

\[
= x_t' Q_t x_t + x_t' A' K_{t+1} A_t x_t + \sum_{s=t}^{T-1} \mathbb{E}[w_s' K_{s+1} w_s]
\]

\[
+ 2 \lambda'(S_{t+1} A_t x_t - \sum_{s=t}^{T-1} g_s) - \lambda' V_{t+1} \lambda
\]

\[
+ \min_{u_t} \{ 2 \lambda'(F_t + S_{t+1} B_t) u_t + u_t' (R_t + B_t' K_{t+1} B_t) u_t + 2 x_t' A_t' K_{t+1} B_t u_t \}.
\]
By differentiating with respect to \( u_t \) and by setting the derivative equal to zero, we obtain

\[
(R_t + B_t'K_{t+1}B_t)u_t = -(B_t'K_{t+1}A_t x_t + (F_t + S_{t+1}B_t)')\lambda.
\]

The matrix multiplying \( u_t \) on the left is positive definite since \( R_t \) is positive definite and \( B_t'K_{t+1}B_t \) is positive semidefinite. It follows that the minimizing control of this Lagrangian problem is

\[
u_t^* = -(R_t + B_t'K_{t+1}B_t)^{-1}(B_t'K_{t+1}A_t x_t + (F_t + S_{t+1}B_t)')\lambda.
\]

By substitution into the expression for \( L_t \), we have

\[
L_t(x_t; \lambda)
= x_t'Q_t x_t + x_t'A_t'K_{t+1}A_t x_t + \sum_{s=t}^{T-1} \mathbb{E}[w_s' K_{s+1} w_s] + 2\lambda'(S_{t+1}A_t x_t - \sum_{s=t}^{T-1} g_s) - (B_t'K_{t+1}A_t x_t + (F_t + S_{t+1}B_t)')\lambda
\]

\[
-\lambda'(F_t + S_{t+1}B_t)(R_t + B_t'K_{t+1}B_t)^{-1}(F_t + S_{t+1}B_t)')\lambda
+ 2\lambda' \left( (S_{t-1} - (F_t + S_{t+1}B_t)(R_t + B_t'K_{t+1}B_t)^{-1}B_t'K_{t+1}A_t) x_t \right) + x_t'Q_t x_t + x_t'A_t'K_{t+1}A_t - A_t'K_{t+1}B_t (R_t + B_t'K_{t+1}B_t)^{-1}B_t'K_{t+1}A_t
\]

\[
= \sum_{s=t}^{T-1} \mathbb{E}[w_s' K_{s+1} w_s] - \lambda'(V_{t+1} + (F_t + S_{t+1}B_t)(R_t + B_t'K_{t+1}B_t)^{-1}(F_t + S_{t+1}B_t))'\lambda
+ 2\lambda' \left( (S_{t-1} - (F_t + S_{t+1}B_t)(R_t + B_t'K_{t+1}B_t)^{-1}B_t'K_{t+1}) A_t x_t - \sum_{s=t}^{T-1} g_s \right) + x_t'Q_t x_t + x_t'A_t' (K_{t+1} - K_{t+1}B_t (R_t + B_t'K_{t+1}B_t)^{-1}B_t'K_{t+1}) A_t x_t
\]

\[
= \sum_{s=t}^{T-1} \mathbb{E}[w_s' K_{s+1} w_s] + x_t'K_t x_t - \lambda' V_t x_t + 2(S_t x_t - \sum_{s=t}^{T-1} g_s)'\lambda.
\]

The semipositive-definiteness and symmetry of \( V_t \) and \( K_t \) follow by induction.
Bibliography


189


