A Robust Optimization Approach to Online Problems

by

Nikita Korolko

B.S., Novosibirsk State University (2010)
M.S., Novosibirsk State University (2011)

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 2017

Author .....
Sloan School of Management
May 12, 2017

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

Accepted by .....
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center
Thesis Supervisor

A Robust Optimization Approach to Online Problems

by

Nikita Korolko

Submitted to the Sloan School of Management
on May 12, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

In this thesis, we consider online optimization problems that are characterized by incrementally revealed input data and sequential irrevocable decisions that must be made without complete knowledge of the future. We employ a combination of mixed integer optimization (MIO) and robust optimization (RO) methodologies in order to design new efficient online algorithms that outperform state-of-the-art methods for many important practical applications. We empirically demonstrate that RO-based algorithms are computationally tractable for instances of practical size, generate more cost-effective decisions and can simultaneously model a large class of similar online problems due to exceptional modeling power of MIO.

In Part I, we consider the well-known K-server problem from the perspective of robust adaptive optimization. We propose a new tractable mixed integer linear formulation of the K-server problem that incorporates both information from the past and uncertainty about the future. By combining ideas from classical online algorithms developed in the computer science literature and robust and adaptive optimization developed in the operations research literature we propose a new method that (a) is computationally tractable, (b) almost always outperforms all other methods in numerical experiments, and (c) is stable with respect to potential errors in the assumptions about the future.

In Part II, we consider several extensions of the asset-based weapon-to-target assignment problem whose objective is to protect ships in a fleet from incoming threats. We demonstrate that the new highly nonlinear MIO formulation (a) can be combined with lazy constraints techniques allowing the system designer to find optimal solutions in real time, (b) can be extended to the multiperiod setting, and (c) admits a decentralized solution with limited loss of optimality.

In Part III, we present a novel covariate-adaptive optimization algorithm for online allocation in clinical trials. The new approach leveraging MIO and RO techniques (a) guarantees a better between-group covariate balance in comparison with state-of-the-art methods, (b) yields statistical power at least as high as, and sometimes

significantly higher than, randomization-based algorithms, and (c) is well protected against selection, investigator and accidental bias.

Thesis Supervisor: Patrick Jaillet
Title: Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

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

Acknowledgments

First, my deepest gratitude goes to my academic advisors Dimitris and Patrick for teaching the most important lessons during my years at MIT, both mathematical and personal. I will be forever thankful for their invitation to the ORC family, for believing in me and their inspiring and sincere guidance. It was a great honor for me to work with them.

I was very fortunate to have Juan Pablo Vielma in my thesis committee. His kindness, insightful questions and recommendations, as well as perceptive discussions made an exceptionally positive impact on the quality of the dissertation.

My learning curve at MIT was much smoother thanks to my collaborator Alexander Weinstein. After innumerous days of brainstorming new ideas, polishing formulations and celebrating the new results together, I am certain, that the total collaboration effect significantly exceeds the sum of individual efforts.

I am very grateful to Zafer and Gozde Sahinoglu for their priceless support, heartfelt warmth and mutual friendship.

My first real-world optimization project was exceptional and unforgettable thanks to the invitation of Tor Schoenmeyr to work with him at Tesla over the summer. I think it is impossible to do a better job than Tor did as a boss. We not only did great research together, but also enjoyed fantastic extracurricular activities.

I gratefully acknowledge the wisdom, inspiration, help and kindness of my friends Nataly Youssef and Miles Lubin. They definitely made my MIT journey more exciting and made the ORC feel like an even friendlier environment to work in.

There are also some individuals in the Eastern Hemisphere without whom I would not be able to attend MIT and have this incredible life changing experience. Sergey K. Vodopyanov is an extraordinary advisor, mentor and friend, who did much more for me than I could even hope for during my years at Novosibirsk State University. Nataliya Aulchenko not only helped me to diligently design a long-term plan for the application process, but also to successfully realize it. It was a great pleasure to have Sergey S. Goncharov as the dean during my Bachelor and Master programs. His

commitment and assistance with the application process was invaluable.

Finally, my sincere gratitude goes to my small, but precious family - Valentina, Elena, Evgeny and Anya - for their unconditional and continuous love. Beyond all doubts, everything I have now I owe to them.

Nikita Korolko

April 2017

Cambridge, MA

To my beloved family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Every computational algorithm that solves a particular numerical problem performs a sequence of operations with input data in order to generate a final output decision. There are two large classes of algorithms that differ from each other based on availability of all problem data ahead of time. An *offline* algorithm is given all input data from the very beginning, while an *online* algorithm, on contrary, receives problem data sequentially piece by piece and is obliged to immediately respond without knowing the future data with certainty.

The primary focus of this thesis is the class of online problems that has numerous important applications. First, some remarkable optimization problems are intrinsically online, for instance, in the decision-making domains where waiting for all relevant information is costly or impossible (multi-armed bandit, secretary, paging problems). Second, there are some fundamental optimization problems that can be formulated and considered as deterministic programs (bin packing, load balancing, scheduling and matching). However, their online counterparts may have much higher relevance and practical value. The reason is that the offline formulations typically exploit *estimations* of future parameters (but the quality of the best available forecast can be poor) and they do not address potential unexpected events that may dramatically affect the system. The online approach may significantly alleviate both these concerns.

In this introductory chapter, we briefly discuss existing frameworks for model-

ing and solving online problems (Section 1.1). We also present a generic modeling approach based on a combination of Mixed Integer Optimization (MIO) and Robust Optimization (RO) that is the core of all online algorithms designed in this thesis. The proposed optimization-based framework has a large number of modeling, qualitative and numerical advantages that we discuss in Section 1.2. Finally, in Section 1.3, we introduce three optimization problems from different decision-making domains that serve as practical examples that the proposed MIO approach may significantly outperform existing online algorithms in terms of efficiency, robustness and computational tractability.

1.1 Online Computation and Optimization

The generic concept of an optimization problem P includes several standard ingredients: input data D, decision variables X that belong to a feasible set $F(D)$ as well as the system performance measure or a cost function $C(D, X)$. In offline computation, the problem data D are assumed to be known exactly and to be fully available for the decision-maker in the very beginning of the calculation process. In an online setting, however, a sequence of actions produced by an algorithm must be made based on the past and current fragments of the input data, while information about the future remains uncertain. More precisely, the sequential model of the online problem assumes that there is a discrete sequence of time periods $t = 1, \ldots, N$, and input data $D = (D_1, \ldots, D_N)$ are revealed incrementally, that is, at any given time-step $\tau \in \{1, \ldots, N\}$ a decision X_τ (as a fragment of a series of actions $X = (X_1, \ldots, X_N)$) must be made solely based on $D_1, \ldots, D_\tau$.

Let ξ denote the vector of variables representing uncertainty of the future input data that affects the optimization problem P. In this case, online optimization of $C(D, X; \xi)$ under uncertainty ξ admits a large number of modeling approaches that vary in assumptions about uncertain parameters, computational complexity and level of conservatism. Below we briefly outline a few popular methodologies for online optimization including stochastic optimization, competitive analysis and RO approach,

which is a primary tool used in this thesis.

Stochastic Optimization

One of the frameworks to studying online algorithms is stochastic optimization, whereby one may model uncertain parameters ξ of the problem P as random variables with some additional assumptions about their probabilistic distribution. This approach implies that the objective function $C(D, X; \xi)$ and constraints describing the feasible set $F(D; \xi)$ can be random. The stochasticity in linear problems was first considered and modeled by Dantzig in his seminal work [1955], while Charnes and Cooper [1959] introduced a concept of chance constrained and percentile optimization.

In the setting of partially random input data, one may study the system performance in terms of expected value of the cost function

$$\min_{X \in F(D;\xi)} \mathbb{E}_{\xi \sim \Xi} \Big[C(D, X; \xi) \Big], \tag{1.1}$$

where mathematical expectation is calculated with respect to uncertainty parameters ξ that follow some prespecified distribution Ξ.

Stochastic optimization has been applied widely to different online problems, for instance, scheduling [Rothkopf, 1966, Möhring et al., 1984], routing [Bent and Van Hentenryck, 2003, Oyola et al., 2016], matching problems [Karp et al., 1990], as well as combinatorial auctions [Hoos and Boutilier, 2000] and ad allocation problems [Mehta et al., 2007]. However, this approach has several major drawbacks. First, it is not always possible to identify or even approximate properly underlying probability distribution Ξ due to a lack of relevant historical data, especially if this distribution is not stable and varies over time. Second, even if we are able to determine a good distribution fit to the available data, results of optimization (1.1) can be very sensitive to assumptions about probability distribution Ξ. Finally, in a multiperiod dynamic scenario, computational complexity of problem (1.1) grows exponentially in a length of the optimization horizon. While recent computational advances permit stochastic linear formulations to be solved efficiently, generic nonlinear models or models with integrality constraints still impose major tractability challenges for practitioners.

Competitive Analysis

An alternative approach to solving problems with incrementally revealed data, that does not require any assumptions about neither probability distribution nor the structure of the uncertainty, is competitive analysis. The key idea of this framework is to measure the performance of an online algorithm $\mathbb{A}$ in terms of its *competitive ratio*, that for cost minimization problems can be defined as follows:

$$\sup_{I \in \mathcal{I}} \frac{C_{\mathbb{A}}(I)}{C_{OPT}(I)}, \tag{1.2}$$

where supremum is taken with respect to all possible instances of input data $\mathcal{I}$ and OPT denotes the optimal offline method that has full knowledge of the future.

Probably the first description of ideas behind competitive analysis applied to allocation of jobs among identical computer processors is due to Graham [1969], while Yao considered the performance of heuristic online bin packing algorithms in [1980]. The first methodical analysis of online algorithms was published by Sleator and Tarjan [1985] and Karlin et al. [1986]. We refer the reader to [Borodin and El-Yaniv, 2005, Albers, 1996] as well as [Jaillet and Wagner, 2010] for a comprehensive survey of a wide selection of applications and techniques leveraging competitive analysis.

At the same time, this online computation framework has a number of practical limitations. First, given that the competitive ratio defined by (1.2) is a worst-case metric with respect to all possible realizations of input data, the resulting online algorithms may generate overly conservative decisions. Furthermore, there are no straightforward parameters that a decision maker can tune in order to control the level of conservatism. Second, online algorithms are typically heuristic (therefore, potentially suboptimal) and *ad hoc*, that is each online problem usually requires its own new approach for designing a method that would solve it efficiently. Finally, competitive analysis framework does not incorporate any assumptions about the uncertain future (despite the fact that some valuable information may be available) and relies only on the historical data observations.

Robust Optimization

The primary goal of the RO framework is to protect the system against all possible realizations of uncertainty that reside in prespecified deterministic *uncertainty set*, rather than assume probability distributions describing the system stochasticity. This immunization can be modeled via the following formulation:

$$\min_{X \in F(D;\xi)} \max_{\xi \in U(\Gamma)} C(D, X; \xi),$$

where uncertainty set $U(\Gamma)$ is parametrized by Γ, which controls the size of the set and the level of the model conservatism.

Robust optimization methodology was first introduced by Ben-Tal and Nemirovski [1998, 1999], who considered the linear programming formulations with ellipsoidal uncertainty sets that turned out to be equivalent to computationally tractable second order cone problems. The pioneering work on robust solutions of quadratic and semidefinite programs is due to El Ghaoui and Lebret [1997], El Ghaoui et al. [1998]. Bertsimas and Sim [2003, 2004], Bertsimas et al. [2004] discuss risk measures and trade-offs between system efficiency and its robustness with respect to input data uncertainty. The authors demonstrate that linear programs with polyhedral uncertainty sets admit robust counterparts in a form of tractable linear optimization problem, and that the price of injected robustness is low. We refer the reader to [Ben-Tal et al., 2009, Bertsimas et al., 2011, Gorissen et al., 2015] for both theoretical and practical details on RO.

In this thesis, we leverage robust optimization methodology in order to model system uncertainty. The major practical benefits of this approach that motivate its implementation are:

1. Numerical tractability, i.e., the ability to solve instances of practical size within reasonable time for a giving application.

2. Avoiding the difficulty of fitting a probability distribution to available data.

3. Efficient system performance with low price of robustness.

4. Ability to regulate the level of model conservatism.

5. Probabilistic guarantees of problem feasibility [Bertsimas and Sim, 2004].

6. Practical data-driven guidelines for constructing uncertainty sets based on historical data [Bertsimas and Brown, 2009, Bertsimas et al., 2017].

1.2 Proposed Framework

Mixed Integer Optimization is recognized to be a powerful and universal tool to modeling and solving a wide spectrum of both offline and online problems [Jünger et al., 2009]. At the same time, it is well known that even in the simpler offline setting solving a nominal optimization problem that exploits estimated parameters can be practically useless due to severe sub-optimality, sensitivity or even infeasibility of a generated solution. Needless to say, that in the online framework, the problem of uncertain input data becomes even more relevant.

In this thesis, we employ a combination of MIO and RO methodologies to process online optimization problems under uncertainty. The benefit of implementing this combination is threefold. First, it turns out that many important online problems from different decision-making domains can be modeled in the form of mixed integer multiperiod optimization problem due to exceptional modeling power of MIO. Second, there is a natural split of input parameters into two groups: certain (that have already been realized and are known to an algorithm) and uncertain (that represent unknown future fragments of input data). Finally, RO methodology allows the system designer to incorporate useful and relevant information about future parameters by means of uncertainty set rather than just to use only historical observations. The generic schema that we leverage in this thesis in order to produce efficient RO algorithms for specific complex online problems can be described as follows:

1. Formulate an online problem as a mixed integer (potentially nonlinear) discrete-time multiperiod optimization problem.

2. Employ robust optimization techniques and solve the problem. Extract current time-step decisions from the optimal solution.

3. Address specific practical needs of the online problem: reduction of solving time, incorporation of communication, injection of randomization, etc.

4. Run computational experiments demonstrating the edge of RO-based algorithms in comparison with existing heuristics.

This schema is not new and proven to be practically advantageous in many fields including supply chain [Thiele, 2004], finance [Bandi and Bertsimas, 2014], energy [Bertsimas et al., 2013], etc. We build upon recent advances in mathematical optimization and amplify the power of MIO and RO approach by considering new important applications. We also demonstrate the following advantages of the optimization framework that augment the list in the end of Section 1.1:

- Universality. MIO formulations are often flexible and, therefore, model *a class* of similar online problems.

- Fast computation. MIO formulations can be combined with lazy constraints techniques in order to expedite solving of large-scale and nonlinear problems.

- Decentralization. When combined with RO techniques, MIO problems modeling the centralized multi-agent setting may admit extensions to the distributed setting via introduction of auxiliary communication decision variables.

1.3 Thesis Outline and Main Contributions

The primary objective of this thesis is to demonstrate that the aforementioned advantages of robust and mixed-integer optimization often allow a practitioner to design brand new efficient algorithms for solving online problems that outperform state-of-the-art methods. We consider three applied online optimization problems under uncertainty originating from different decision-making domains: scheduling and dynamic resource allocation (Chapter 2), multi-agent cooperation (Chapter 3) and sequential clinical trials (Chapter 4). We empirically prove that in all three considered settings

robust optimization-based algorithms are computationally tractable for instances of practical size, and they generate decisions with better final objective values.

Chapter 2. The K-Server Problem.

In Chapter 2, we reconsider the well-known K-server problem from the perspective of mixed integer, robust and adaptive optimization. Existing online methods typically exploit only information from the past in order to make the next decision. We propose a new tractable mixed integer linear formulation of the K-server problem that includes both information from the past and available assumptions about the future.

Combining ideas behind an existing online method called the Work Function Algorithm and adjustable robust optimization, we design a new method that inherits positive properties of both approaches:

(a) It is computationally tractable.

(b) It gradually outperforms pure online algorithms and classical adaptive optimization methods when increasing information about the future becomes available.

(c) It is stable with respect to potential errors in the assumptions about the future.

Research in Chapter 2 is a joint work with Prof. Dimitris Bertsimas and Prof. Patrick Jaillet. The paper titled "The K-Server Problem via a Modern Optimization Lens" is submitted to *INFORMS Journal on Computing*.

Chapter 3. The Fleet Defense Problem.

In Chapter 3, we consider several extensions of the asset-based weapon-to-target assignment problem whose objective is to protect assets in a fleet from incoming threats.

(a) We prove that this highly nonlinear mixed integer optimization problem can be efficiently solved with lazy constraints techniques, and therefore optimal solutions can be obtained online for instances of practical size.

(b) We design a new MIO formulation for multiperiod scenarios, when the fleet has to plan the defense strategy for several consecutive attacks.

(c) We develop communication and coordination protocols for the decentralized version of the problem, in which captains of the assets have to make local decisions based on their own objectives and some limited communication with other ships. The suggested protocol uses robust optimization principles and generates weapon assignments that significantly improve upon a no-communication decentralized solution.

Research in Chapter 3 is a joint work with Prof. Dimitris Bertsimas and Prof. Patrick Jaillet. The paper titled "Multiperiod Optimization for Fleet Defense: Centralized and Distributed Approaches" is submitted to *European Journal of Operational Research*.

Chapter 4. Sequential Clinical Trials.

Pharmaceutical companies spend tens of billions of dollars each year to operate multi-year clinical trials needed for the approval of new drugs. In Chapter 4, we present a novel covariate-adaptive optimization algorithm for online allocation in clinical trials that leverages robust mixed integer optimization and has the following practical benefits:

(a) In all tested scenarios, the proposed method guarantees a better between-groups covariate balance in comparison with state-of-the-art covariate-adaptive randomization approaches.

(b) A new algorithm also yields statistical power at least as high as, and sometimes significantly higher than, randomization-based methods. We present a setting in which our algorithm achieves a desired level of power at a sample size 25-50% smaller than that required with state-of-the-art approaches.

(c) We empirically demonstrate that the proposed algorithm is well protected against selection, investigator and accidental bias.

(d) We prove that a complex robust optimization formulation with second order constraints and ellipsoidal uncertainty set admits a closed-form solution, what

makes suggested algorithm to be applicable not just for clinical trials, but also for settings requiring real-time decisions.

Research in Chapter 4 is a joint work with Prof. Dimitris Bertsimas and Alexander M. Weinstein. The paper titled "Covariate-adaptive Optimization in Online Clinical Trials" is submitted to *Biometrics*.

Chapter 2

The K-Server Problem

2.1 Introduction

The K-server problem, introduced by Manasse et al. [1990], is one of the most fundamental problems considered from the perspective of online algorithms and competitive analysis [Borodin and El-Yaniv, 2005]. Given a metric space S, K mobile servers are initially located at some predefined points in that space. Over time, service requests appear successively at different locations in the space. Upon knowing the location of the request, one has to immediately dispatch one of K servers to this location. The cost associated with the assignment is the distance between the location of the server and the location of the request. The objective is to minimize the total cost of serving requests, that is the sum of distances traveled by all servers over a given time horizon.

The typical setting for the K-server problem requires that assignment decisions be made in online fashion, that is, by considering only the current and the past requests. The offline version of the K-server is to find an optimal strategy of serving a finite sequence of known requests.

In the online context, one does not have information about precise locations of future requests, this is why online algorithms may incur significantly worse costs than an optimal offline method. One of the common ways to assess the quality of an online algorithm is via its *competitive ratio*, that is the worst-case ratio between the performance of the online algorithm and the optimal offline clairvoyant over all

instances of the problem [Sleator and Tarjan, 1985, Borodin and El-Yaniv, 2005].

The K-server problem together with its variants have many practical applications. First of all, the K-server problem is a generalization of important caching/paging problems that arise in disciplines such as computer science [Sleator and Tarjan, 1985], statistics [Raghavan, 1992], mathematics [Du and Hwang, 1993] and many others. The so-called Cable News Network (CNN) problem where servers move along the lines [Koutsoupias and Taylor, 2000], as well as its simpler versions, such as the bridge and the cow path problem, are also examples that possess the structure of the K-server problem. Additionally, this problem has applications in graph theory, certain scheduling problems [Burley, 1996] and in the theory of metrical task systems [Borodin et al., 1992].

Many online algorithms for solving the K-server problem have been suggested in the literature [Rudec et al., 2013, Floratos and Boppana, 1997]. They differ in their competitive ratios, tractability and amount of historical data they use in order to make the online assignment decisions. In general, the more information is used, the better the performance of the online method is. The Work Function Algorithm (WFA) [Borodin and El-Yaniv, 2005] is one of the most important online algorithms for the K-server problem from both theoretical perspective of competitiveness and practical performance [Rudec et al., 2013]. It considers the full amount of historical data to make its online decisions, as opposed to other simpler online methods.

In practice, it is often the case that information about past observations can lead to some reasonable, data-driven assumptions about the future. In this chapter, the online version of the K-server problem is considered from the perspective of mixed integer optimization (MIO), robust optimization (RO) and adaptive optimization. We combine several optimization techniques and present a new holisitic adaptive optimization method that simultaneously incorporates both information from the past and assumptions about the future, giving a significant edge in the performance of the algorithm.

More specifically, the major contributions of this chapter can be summarized as follows:

1. We propose a new MIO formulation of the offline version of the K-server problem. This formulation is computationally tractable and can model many practical generalizations of the classical problem.

2. We introduce robust and affine adaptive counterparts of the online K-server problem by modeling the uncertainty of the positions of future requests as an uncertainty set in the context of the new MIO approach. To the best of our knowledge, this is the first application of robust optimization to online algorithms.

3. We design a new algorithm called the Holistic Adaptive Robust Optimization (HARO) method by combining ideas of the WFA and affine adaptive robust optimization (AARO) techniques. To the best of our knowledge, this is the first algorithm that combines ideas from the computer science and operations research tradition in the context of online algorithms.

4. We empirically demonstrate that the HARO method is tractable and almost always outperforms all other methods considered including WFA and AARO. HARO remains the best method for various settings: finite and continuous metric space S, uniform and non-uniform distribution of the requests and different number of servers and locations. Moreover, HARO is stable with respect to potential errors in the uncertainty set describing the future time stages.

The rest of the chapter is organized as follows. In Section 2.2, we briefly discuss existing online algorithms for solving the K-server problem. We represent the offline variant of the K-server problem as a mixed binary optimization problem. We also give a MIO formulation of the WFA procedure. In Section 2.3, we extend the offline formulation from Section 2.2 to the online setting of the K-server problem and design its RO and AARO counterparts. In Section 2.4, we introduce the new HARO method combining the formulations of WFA and AARO algorithms. In Section 2.5, we consider generalizations of the HARO approach that can be adapted to different settings. We consider various types of the underlying metric space S (multidimensional and/or

finite) with different distance functions, as well as weighted version of the HARO algorithm. In Section 2.6, we give examples of possible uncertainty sets and describe a way to measure varying amount of information about future requests. In Section 2.7, we present our numerical results and empirically prove the computational tractability, efficiency and stability of the proposed HARO method.

2.2 Problem Formulation and Existing Methods

In this section, we first describe in detail the specific variants of the K-server problem that we consider. We also present a brief overview of the most frequently used online algorithms for solving this class of problems. Finally, we introduce a new MIO formulation of the offline K-server problem and a MIO formulation generating the decisions made by the WFA. For the sake of clarity, the formulations in this section are designed for problems with many simplifying assumptions, most of which will be relaxed in Section 2.5.

2.2.1 Basic One-dimensional Version of the K-Server Problem

We consider a one-dimensional continuous compact metric space $S = [0, 1]$ with standard Euclidean metric $d(x, y) = |x - y|$. We assume that there are K mobile servers within S that should respond to the requests that appear sequentially in S. The starting locations of the servers, $\hat{\mathbf{x}}^0 = \{\hat{x}_k^0 \in S \,|\, k = 1, \ldots, K\}$, are assumed to be known and form the initial configuration of the system. In what follows, we put a "hat" symbol on variables with known values to emphasize the difference between input data and uncertain parameters of the problem.

The sequence of request locations $\{\sigma_t \in S \,|\, t \geq 1, t \in \mathbb{Z}\}$ appears in an online fashion. At any time-step τ, the locations of only the first τ requests $\{\hat{\sigma}_1, \ldots, \hat{\sigma}_\tau\}$ are known to the online decision maker. At time-step τ the new revealed request at location $\hat{\sigma}_\tau$ should be immediately served by a server $k \in \{1, \ldots, K\}$, where k is chosen according to a proposed algorithm $\mathbb{A}$, and before the next request is revealed at time-step $\tau + 1$. In this case, the configuration of the system, described by the

vector of the locations of all K servers, $\mathbf{x}^\tau$, changes according to the rule

$$\mathbf{x}^{\tau-1} = \{x_1^{\tau-1}, \ldots, x_{k-1}^{\tau-1}, x_k^{\tau-1}, x_{k+1}^{\tau-1}, \ldots, x_K^{\tau-1}\} \mapsto$$
$$\mathbf{x}^\tau = \{x_1^{\tau-1}, \ldots, x_{k-1}^{\tau-1}, \hat{\sigma}_\tau, x_{k+1}^{\tau-1}, \ldots, x_K^{\tau-1}\}.$$

When the server k is assigned to the new request at location $\hat{\sigma}_\tau$ it incurs the cost $d(x_k^{\tau-1}, \hat{\sigma}_\tau) \geq 0$, where $x_k^{\tau-1} \in S$ denotes the position of server k at the end of time-step $\tau - 1$. The objective of an online algorithm $\mathbb{A}$ is to minimize the overall total distance traveled by all K servers. The basic version of the K-server problem assumes that a request cannot be canceled or postponed, does not incur service time while being served, and that the time it takes for a server to move between locations is negligible. As a result, when a new request $\hat{\sigma}_\tau$ appears, all K servers are available to be assigned.

2.2.2 Offline and Online Algorithms

In this subsection, we briefly discuss various well-known algorithms (both offline and online) for solving the K-server problem [Rudec et al., 2013, Floratos and Boppana, 1997].

First of all, if the sequence of requests $\boldsymbol{\sigma}$ is finite of known length N, and all components of the vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ are known ahead of time, then the corresponding offline linear optimization problem can be modeled as a standard network flow optimization problem [Chrobak et al., 1991, Bazaraa et al., 2011] that can be solved efficiently. The objective value of such an offline optimal solution represents a benchmark for evaluating the performance of online algorithms.

One way of measuring the quality of an online algorithm $\mathbb{A}$ is its competitive ratio as introduced in Sleator and Tarjan [1985]. Let us assume that $C_{OPT}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma})$ and $C_{\mathbb{A}}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma})$ are the total costs incurred by an offline algorithm OPT and an online algorithm $\mathbb{A}$, respectively, for the K-server problem with initial locations of the servers $\hat{\mathbf{x}}^0$ and a sequence of requests $\boldsymbol{\sigma}$. If there are constants $c_1 > 0$ and c_2 such that for

any input data of the problem $\hat{\mathbf{x}}^0$ and $\boldsymbol{\sigma}$ the following inequality holds

$$C_{\mathbb{A}}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma}) \leq c_1 \cdot C_{OPT}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma}) + c_2, \tag{2.1}$$

then the algorithm $\mathbb{A}$ is said to be c_1-competitive. In that case, the competitive ratio of $\mathbb{A}$ is defined as the infimum of all c_1 such that $\mathbb{A}$ is c_1-competitive. When no such finite constants exist, then the online algorithm $\mathbb{A}$ is sometimes called non-competitive.

Probably the most straightforward deterministic online algorithm for solving the K-server problem is GREEDY. This method assigns the closest server to the request. Despite the fact that it is not competitive for any fixed value of c_1, the empirical experiments presented in [Rudec et al., 2013] demonstrate that in many different settings GREEDY performs reasonably well. BALANCE is another deterministic non-competitive algorithm that tries to keep the total traveled distance by all servers as equal as possible. It assigns a server whose cumulative distance traveled so far plus the distance to the new request location is smallest [Chrobak et al., 1991].

In contrast to deterministic algorithms (like GREEDY and BALANCE), *randomized* algorithms contain random steps as part of their design. In this case, the notion of a competitive ratio can be generalized from the perspective of expected costs produced by the algorithms [Borodin and El-Yaniv, 2005]. The competitiveness of randomized online algorithms depends on the adversary model under consideration (oblivious, adaptive online or adaptive offline). We will assume here an oblivious adversary that knows the probability distributions underlying the proposed randomized algorithm $\mathbb{A}$, but not their realizations. In this case, one modifies the definition (2.1) as

$$\mathbb{E}\, C_{\mathbb{A}}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma}) \leq c_1 \cdot C_{OPT}(\hat{\mathbf{x}}^0, \boldsymbol{\sigma}) + c_2, \tag{2.2}$$

where $\mathbb{E}[\cdot]$ is the mathematical expectation operator taken with respect to the random choices made by $\mathbb{A}$.

A natural randomized online algorithm, called RAND, chooses a server for each request randomly with equal probability. However, it ignores historical data and dis-

tances from the servers to the current request location. The HARMONIC algorithm, introduced by Raghavan and Snir [1989], randomizes between servers with probabilities inversely proportional to distances between their locations and the position of the request. This method is also memoryless and $O(2^K log(K))$-competitive against an adaptive online adversary [Bartal and Grove, 2000], and no better result is known for HARMONIC, even against the weaker oblivious adversary.

One of the most important online algorithms for the K-server problem is WFA [Rudec et al., 2013, Koutsoupias and Papadimitriou, 1995]. At time-step τ this deterministic method selects a server k^* as the solution of the following optimization problem:

$$k^* = \arg \min_{k=1,\dots,K} \Big\{ C_{OPT}\big(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \dots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)\big) + d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau) \Big\}, \tag{2.3}$$

where the vector $\hat{\mathbf{x}}^\tau(k) = \big(\hat{x}_1^\tau(k), \dots, \hat{x}_K^\tau(k)\big)$ is defined as follows:

$$\hat{x}_j^\tau(k) = \begin{cases} \hat{x}_j^{\tau-1}, & \text{if } j \neq k \\ \hat{\sigma}_\tau, & \text{if } j = k. \end{cases}$$

The first term in (2.3), $C_{OPT}\big(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \dots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)\big)$, is called the *work function*. It represents the optimal value of the offline problem with the known sequence of requests $\hat{\sigma}_1, \dots, \hat{\sigma}_\tau$ and fixed initial and final configurations $\hat{\mathbf{x}}^0$ and $\hat{\mathbf{x}}^\tau(k)$, respectively. Intuitively, WFA tries to find a good balance between a strategic solution (represented by the work function term) and a greedy solution (second term of (2.3)). There are some important generalizations of the WFA method [Rudec et al., 2013], including a *truncated* version w-WFA, when only the most recent w observations are included into the optimization of the work function

$$C_{OPT}\big(\hat{\mathbf{x}}^{\tau-w}, \hat{\sigma}_{\tau-w}, \dots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)\big), \tag{2.4}$$

and a *weighted* counterpart WFA_λ proposed for some applications [Burley, 1996], in

which there is a positive constant weight λ in front of the second term

$$C_{OPT}\big(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \ldots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)\big) + \lambda \cdot d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau).$$

For many online problems, WFA gives the best possible competitive ratio or it is conjectured to be best possible Borodin and El-Yaniv [2005]. For example, for the classical K-server problem it has the best known competitive ratio of $2K-1$ for any metric space, and a competitive ratio of K for some special cases of the problem [Chrobak et al., 1991, Koutsoupias and Papadimitriou, 1995, Koutsoupias, 1999]. Moreover, in many practical settings, WFA demonstrates superior empirical perfomance in terms of total cost incurred compared to other online algorithms [Rudec et al., 2013, Bartal and Grove, 2000]. The empirical ratio between the cost incurred by WFA and OPT in practice is significantly smaller than the theoretical upper bound of $2K-1$.

Taking into account the theoretical and practical value of WFA, we use this algorithm as a starting point for constructing a more efficient method for solving the K-server problem from a robust and adaptive optimization point of view.

2.2.3 MIO Formulation of the Offline K-Server Problem

Let us formulate the offline version of the K-server described in Section 2.2.1 as a multiperiod mixed binary linear problem. This model will be easily extended to more realistic settings later by virtue of the strong modeling power of MIO.

Following the basic notation introduced in Section 2.2.1, we first define the decision variables used in our formulation:

$x_k^t \in [0,1]$ represents the position of server $k \in \{1,\ldots,K\}$ at the end of time-step $t \in \{1,\ldots,N\}$;

$y_k^t \in \{0,1\}$ is a binary assignment indicator equal to 1 if server k is assigned to request $\hat{\sigma}_t$.

The objective function minimizing the cumulative distance traveled by all servers

can then be expressed as

$$\min \sum_{k=1}^{K} \sum_{t=1}^{N} |x_k^t - x_k^{t-1}|,$$

where we have

$$x_k^t = x_k^{t-1} + y_k^t(\hat{\sigma}_t - x_k^{t-1}), \tag{2.5}$$

that is, the location of server k at time-step t is either the same as it was at the previous time-step $t-1$, if $y_k^t = 0$ (i.e., if server k is not dispatch to request $\hat{\sigma}_t$), or is equal to $\hat{\sigma}_t$, if $y_k^t = 1$.

Therefore, the offline K-server problem can be formulated as the following multi-period optimization problem:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{y}} \quad & \sum_{k=1}^{K} \sum_{t=1}^{N} |x_k^t - x_k^{t-1}| \\
\text{s.t.} \quad & x_k^t = x_k^{t-1} + y_k^t(\hat{\sigma}_t - x_k^{t-1}), \quad t = 1, \ldots, N, k = 1, \ldots, K \\
& x_k^0 = \hat{x}_k^0, \quad k = 1, \ldots, K \\
& \sum_{k=1}^{K} y_k^t = 1, \quad t = 1, \ldots, N \\
& 0 \le x_k^t \le 1, \; y_k^t \in \{0, 1\}, \quad t = 1, \ldots, N, k = 1, \ldots, K.
\end{aligned} \tag{2.6}$$

Formulation (4.2) is nonlinear due to products of the form $y_k^t x_k^{t-1}$. These products can easily be linearized by introducing $z_k^t = y_k^t x_k^{t-1}$ and the following additional constraints:

$$x_k^t = x_k^{t-1} + y_k^t \hat{\sigma}_t - y_k^t x_k^{t-1} = x_k^{t-1} + y_k^t \hat{\sigma}_t - z_k^t,$$

where

$$z_k^t \le y_k^t, \quad z_k^t \le x_k^{t-1}, \quad z_k^t \ge y_k^t + x_k^{t-1} - 1.$$

As a result, the offline K-server problem can be formulated as a mixed binary linear optimization problem:

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}} \quad \sum_{k=1}^{K} \sum_{t=1}^{N} v_k^t$$

$$
\begin{aligned}
\text{s.t.} \qquad & \text{For } k = 1, \ldots, K;\ t = 1, \ldots, N: \\
& v_k^t \geq x_k^t - x_k^{t-1} \\
& v_k^t \geq -(x_k^t - x_k^{t-1}) \\
& x_k^t = x_k^{t-1} + y_k^t \hat{\sigma}_t - z_k^t \\
& z_k^t \leq y_k^t \\
& z_k^t \leq x_k^{t-1} \\
& z_k^t \geq y_k^t + x_k^{t-1} - 1 \\
& 0 \leq x_k^t,\ z_k^t \leq 1,\ y_k^t \in \{0, 1\} \\
& x_k^0 = \hat{x}_k^0, \quad k = 1, \ldots, K \\
& \sum_{k=1}^{K} y_k^t = 1, \quad t = 1, \ldots, N.
\end{aligned}
\tag{2.7}
$$

2.2.4 MIO Formulation of the WFA

WFA (2.3) makes successive assignment decisions for the K-server problem. At each time-step τ (for $1 \leq \tau \leq N$), one may obtain this assignment decision as part of an optimal solution of a MIO formulation that we present in this section.

According to the definition of WFA for a fixed time-step τ, in order to make a locally optimal assignment decision $y_{k^*}^\tau$, we need to know the initial locations of the servers $\{\hat{x}_k^0 \,|\, k = 1, \ldots, K\}$, the sequence of requests $\{\hat{\sigma}_t \,|\, t = 1, \ldots, \tau\}$ and the locations of the servers at the end of time-step $\tau - 1$, that is,

$$
\hat{\mathbf{x}}^{\tau-1} = \big(\hat{x}_1^{\tau-1}, \ldots, \hat{x}_K^{\tau-1}\big).
$$

Similarly to (2.7), instead of solving K optimization problems of the type (2.3), we find an optimal assignment k^* by solving the following consolidated optimization problem:

$$
\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{v}} \qquad \sum_{k=1}^{K} \sum_{t=1}^{\tau} v_k^t + \sum_{k=1}^{K} y_k^\tau \cdot d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau)
$$

$$
\begin{aligned}
\text{s.t.} \quad & \text{For } k = 1, \ldots, K: \\
& v_k^t \geq x_k^t - x_k^{t-1}, \quad t = 1, \ldots, \tau \\
& v_k^t \geq -(x_k^t - x_k^{t-1}), \quad t = 1, \ldots, \tau \\
& x_k^t = x_k^{t-1} + y_k^t \hat{\sigma}_t - z_k^t, \quad t = 1, \ldots, \tau - 1 \\
& x_k^\tau = \hat{x}_k^{\tau-1} + y_k^\tau (\hat{\sigma}_\tau - \hat{x}_k^{\tau-1}) \\
& z_k^t \leq y_k^t, \quad t = 1, \ldots, \tau - 1 \\
& z_k^t \leq x_k^{t-1}, \quad t = 1, \ldots, \tau - 1 \\
& z_k^t \geq y_k^t + x_k^{t-1} - 1, \quad t = 1, \ldots, \tau - 1 \\
& x_k^0 = \hat{x}_k^0 \\
& 0 \leq x_k^t, \, z_k^t \leq 1, \; y_k^t \in \{0, 1\}, \quad t = 1, \ldots, \tau \\
& \sum_{k=1}^{K} y_k^t = 1, \quad t = 1, \ldots, \tau.
\end{aligned}
\tag{2.8}
$$

The only difference with the offline formulation (2.7) is that the locations of the servers at the last time-step τ are constrained by the given configuration $\hat{\mathbf{x}}^{\tau-1}$ and the objective function is augmented with the second "greedy" summand as in definition (2.3).

The formulations of more complex problem settings with multidimensional and finite metric spaces are discussed in Section 2.5.

2.3 The K-Server Problem from a RO Perspective

In the previous section, both formulations (2.7) and (2.8) assume that all request locations explicitly used in the optimization programs are certain. More specifically, vector $(\hat{\sigma}_1, \ldots, \hat{\sigma}_N)$ is a part of the input data for the offline formulation (2.7), and vector $(\hat{\sigma}_1, \ldots, \hat{\sigma}_\tau)$ is certain in the WFA formulation (2.8) at time-step τ.

In our RO-based online algorithm framework described in Section 2.3.1, at each time-step τ (for $1 \leq \tau \leq N$) the allocation decisions $\{y_k^\tau\}_{k=1}^K$ are made by solving one instance of a robust MIO formulation under uncertainty (2.14). More precisely, we

build upon the deterministic formulation (2.7), assuming that at time-step τ requests $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_1, \ldots, \hat{\sigma}_\tau)$ are exactly known, while the future requests $\boldsymbol{\sigma} = (\sigma_{\tau+1}, \ldots, \sigma_N)$ are only known up to some uncertainty. We model this uncertainty using a robust optimization approach [Ben-Tal and Nemirovski, 2008, Bertsimas et al., 2011]. As opposed to stochastic optimization, it does not require the knowledge of the probability distribution for the parameters $\boldsymbol{\sigma}$, but rather assumes that the uncertain data reside in a so-called uncertainty set U represented by a predefined bounded polyhedral set

$$U = \{\mathbf{A}\boldsymbol{\sigma} \leq \mathbf{b}, \boldsymbol{\sigma} \geq \mathbf{0}\}, \tag{2.9}$$

where $\mathbf{A}$ is a given $m \times (N - \tau)$ matrix and $\mathbf{b}$ is a given $m \times 1$ vector. This set of inequalities (2.9) describes *hard* constraints, that is, constraint violation is impossible for any realization of parameters $\boldsymbol{\sigma}$ in the uncertainty set. The robust optimization approach developed in Section 2.3.1 leads to computationally tractable online algorithm for the K-server problem. Moreover, it will allow us to design (in Section 2.3.2) more advanced *adaptive* robust optimization methods that outperform the basic robust counterpart.

2.3.1 Multiperiod RO Formulation of the K-Server Problem

In order to extend the deterministic formulation (2.7) to the online setting with partially unknown values of request locations (namely, $\sigma_{\tau+1}, \ldots, \sigma_N$ are uncertain at time-step τ), we need to modify two types of equalities.

First, given that any server assignment $y_k^t = 1$ incurs a non-negative traveling cost, we substitute the requirement that exactly one of K servers should be assigned to each of the requests with the following equivalent inequality

$$\sum_{k=1}^{K} y_k^t \geq 1, \quad t = \tau, \ldots, N.$$

Second, we eliminate the following identities (used to keep track of the locations of

the servers)

$$x_k^t = x_k^{t-1} + y_k^t(\hat{\sigma}_t - x_k^{t-1}), \tag{2.10}$$

as well as the variables $\mathbf{x}$ from formulation (2.7). In order to do it, we notice that the location x_k^t of server k at the end of time-step t (for $\tau \leq t \leq N$) can be expressed as a function of the initial location $\hat{x}_k^{\tau-1}$ (part of the input data at time-step τ), the sequence of requests $\boldsymbol{\sigma}$ and the assignment decisions $\mathbf{y} = \{y_k^t \,|\, k = 1, \ldots, K,\, t = \tau, \ldots, N\}$. Namely,

$$x_k^t = u[k, t, \tau - 1]\, \hat{x}_k^{\tau-1} + u[k, t, \tau]\hat{\sigma}_\tau + \sum_{j=\tau+1}^{t} u[k, t, j]\sigma_j, \tag{2.11}$$

where $\mathbf{u} = \big\{u[k, t, j]\big\}$ are new binary variables uniquely determining locations of the servers defined as

$$u[k, t, j] = y_k^j \cdot \prod_{i=j+1}^{t} (1 - y_k^i), \quad k = 1, \ldots, K,\, t = \tau - 1, \ldots, N,\, j = \tau - 1, \ldots, t. \tag{2.12}$$

For any time-step t, the server k can be in one of the following $t - \tau + 2$ locations: $\hat{x}_k^{\tau-1}$, $\hat{\sigma}_\tau$ or σ_j (for $j = \tau + 1, \ldots t$). Hence, the binary indicators $\mathbf{u}$ must satisfy the following constraint:

$$\sum_{j=\tau-1}^{t} u[k, t, j] = 1.$$

To see this let us consider all possible options. If the server k was not assigned to any of the requests $\sigma_\tau, \ldots, \sigma_t$, then it is at its initial location $\hat{x}_k^{\tau-1}$,corresponding in (2.11) to the case $u[k, t, \tau - 1] = 1$. If the last time the server k was assigned to a request was j (for some $j \in \{\tau, \ldots, t\}$), and was not assigned to any of the subsequent requests $\sigma_{j+1}, \ldots, \sigma_t$, then at time-step t it is at the location of request σ_j, corresponding to the case $u[k, t, j] = 1$. We add dummy variables $y_k^{\tau-1} := 1$ (for $k = 1, \ldots, K$) to make definition (2.12) correct when $j = \tau - 1$.

We also substitute the change of location of the server k at time-step $t \in \{\tau, \ldots, N\}$ (previously expressed as $x_k^t - x_k^{t-1}$ in (2.7) and (2.8)) by a new expression d_k^t defined

as follows:

$$d_k^t(\mathbf{u},\boldsymbol{\sigma}) = \big(u[k,t,\tau-1]-u[k,t-1,\tau-1]\big)\,\hat{x}_k^{\tau-1} + \big(u[k,t,\tau]-u[k,t-1,\tau]\big)\,\hat{\sigma}_\tau + \\ + \sum_{j=\tau+1}^{t-1}\big(u[k,t,j]-u[k,t-1,j]\big)\sigma_j + u[k,t,t]\sigma_t. \tag{2.13}$$

Expression (2.13) depends on allocation decisions $\mathbf{u}$ and uncertain parameters $\boldsymbol{\sigma}$, and is derived from the change of variables (2.11). Hence, in the RO-based algorithm framework, we solve the following formulation (2.14) at each time-step τ (for $1 \le \tau \le N$):

$$\min_{\mathbf{y},\mathbf{v},\mathbf{u}} \quad \sum_{k=1}^{K}\sum_{t=\tau}^{N} v_k^t \tag{2.14a}$$

s.t. For $k = 1,\dots,K;\ t = \tau,\dots,N$:

$$v_k^t \ge d_k^t(\mathbf{u},\boldsymbol{\sigma}), \quad \forall \boldsymbol{\sigma} \in U \tag{2.14b}$$

$$v_k^t \ge -d_k^t(\mathbf{u},\boldsymbol{\sigma}), \quad \forall \boldsymbol{\sigma} \in U \tag{2.14c}$$

$$u[k,t,j] \le y_k^j, \qquad j = \tau-1,\dots,t \tag{2.14d}$$

$$u[k,t,j] \le 1 - y_k^i, \qquad j = \tau-1,\dots,t,\ i = j+1,\dots,t \tag{2.14e}$$

$$u[k,t,j] \ge y_k^j + \sum_{i=j+1}^{t}(1-y_k^i) - (1+(t-j)-1), \\ j = \tau-1,\dots,t \tag{2.14f}$$

$$\sum_{j=\tau-1}^{t} u[k,t,j] = 1 \tag{2.14g}$$

$$0 \le\ u[k,t,j] \le 1, \qquad j = \tau-1,\dots,t$$

$$y_k^t \in \{0,1\}$$

$$y_k^{\tau-1} = 1; \quad u[k,\tau-1,\tau-1] = 1, \qquad k = 1,\dots,K \tag{2.14h}$$

$$\sum_{k=1}^{K} y_k^t \ge 1, \quad t = \tau,\dots,N. \tag{2.14i}$$

The objective function (2.14a) is to minimize the total distance traveled as defined

by constraints (2.14b) - (2.14c) and expressions (2.13). Inequalities (2.14d) - (2.14f) characterize the linear representation of variables $\mathbf{u}$ (2.12). Identity (2.14g) states that server k at time-step t must be in exactly one of possible locations indexed by $j = \tau-1, \ldots, t$. Constraints (2.14h) define initial values of variables, while inequalities (2.14i) guarantee that we assign at least one server for each request.

The formulation (2.14) includes the uncertain parameters $\boldsymbol{\sigma}$ only in the first two groups of inequalities describing the feasible set. As such, it is a regular robust mixed integer linear optimization problem and has been shown to be computationally tractable by virtue of standard duality techniques [Bertsimas et al., 2011]. We solve the RO problem (2.14), observe an optimal solution $\mathbf{y}^*$ and assign the server k at time-step τ to the request $\hat{\sigma}_\tau$, if $(y_k^\tau)^* = 1$.

Similarly to the idea of the truncated w-WFA method defined in (2.4), we introduce a new parameter T for the RO formulation, which controls (and fixes) the number of subsequent time periods taken into consideration. That is, in order to make a decision at time-step τ, we consider the uncertainty set U associated to the next T requests $(\sigma_{\tau+1}, \ldots, \sigma_{\tau+T})$ only, as opposed to all remaining $N - \tau$ requests $(\sigma_{\tau+1}, \ldots, \sigma_N)$. The truncated version of the RO formulation with parameter T (that we denote as RO(T)) has an identical form (2.14) with the only difference in the limits for time index t. For RO(T) algorithm, we substitute $t = \tau, \ldots, N$ with $t = \tau, \ldots, \min(\tau + T, N)$, where the new upper bound on time index guarantees that t is always less or equal than the problem time horizon N.

2.3.2 Affinely Adjustable Robust Formulation

In this subsection, we develop an online algorithm for the K-server problem based on an affinely adjustable robust optimization (AARO) approach, which is a natural extension to the robust formulation (2.14). AARO is an efficient methodology for solving multiperiod optimization problems that was first suggested by Ben-Tal et al. [2004] and proven to yield less conservative solutions compared to the robust optimization approach in a computationally tractable way.

The motivation for designing an AARO extension to our case is twofold. First,

the K-server problem is by definition a multiperiod optimization problem, justifying a natural split of variables into two groups: non-adjustable *here-and-now* decisions that should be implemented before uncertain parameters $\boldsymbol{\sigma}$ are realized, and adjustable *wait-and-see* variables that can be determined after realization of the uncertain locations of future requests. Second, when the uncertainty set U is not very restrictive and does not provide a lot of information about positions of future requests, consideration of the worst-case scenario (which is intrinsic to robust optimization perspective) may lead to risk-averse and therefore potentially suboptimal solutions. The AARO approach addresses both of these issues.

Following the idea behind AARO [Ben-Tal et al., 2004], we increase the adaptability of the model by assuming that variables y_k^t are affine functions of the uncertainty. More precisely, at time-step $\tau \in \{1, \ldots, N\}$ we set

$$y_k^t = \beta[k,t,\tau] + \sum_{p=\tau+1}^{t} \beta[k,t,p]\sigma_p, \quad k = 1,\ldots,K,\ t = \tau,\ldots N, \tag{2.15}$$

where $\boldsymbol{\beta} = \left\{\beta[k,t,p]\right\}$ become new decision variables replacing variables $\mathbf{y}$. We do keep here-and-now decisions (previously modeled by y_k^τ, $k = 1,\ldots,K$) as binary variables by imposing

$$\beta[k,\tau,\tau] \in \{0,1\}, \quad k = 1,\ldots,K,$$

while wait-and-see decisions (modeled by y_k^t, for $k = 1,\ldots,K, t = \tau+1,\ldots,N$) are *relaxed* to be continuous variables

$$0 \le y_k^t \le 1, \qquad t = \tau+1,\ldots,N,$$

and interpreted as *probabilities* that at time-step t the server k will be assigned to request σ_t. The auxiliary variables $\mathbf{u}, \mathbf{v}$ from formulation (2.14) remain unchanged.

In the online AARO algorithm framework, we solve one instance of the problem (2.16) at each time-step τ (for $1 \le \tau \le N$). It is a standard mixed integer linear robust

optimization problem obtained from formulation (2.14) by substituting variables y_k^t according to formula (2.15) as follows:

$$\min_{\boldsymbol{\beta},\mathbf{u},\mathbf{v}} \quad \sum_{k=1}^{K}\sum_{t=\tau}^{N} v_k^t$$

s.t. For $k = 1,\dots,K;\ t = \tau,\dots,N$:

$$v_k^t \geq d_k^t(\mathbf{u},\boldsymbol{\sigma}), \quad \forall \boldsymbol{\sigma} \in U$$

$$v_k^t \geq -d_k^t(\mathbf{u},\boldsymbol{\sigma}), \quad \forall \boldsymbol{\sigma} \in U$$

$$u[k,t,j] \leq \beta[k,j,\tau] + \sum_{p=\tau+1}^{j} \beta[k,j,p]\sigma_p, \quad j = \tau-1,\dots,t,\ \forall \boldsymbol{\sigma} \in U$$

$$u[k,t,j] \leq 1 - \Big(\beta[k,i,\tau] + \sum_{p=\tau+1}^{i} \beta[k,i,p]\sigma_p\Big),$$

$$j = \tau-1,\dots,t,\ i = j+1,\dots,t,\ \forall \boldsymbol{\sigma} \in U$$

$$u[k,t,j] \geq \Big(\beta[k,j,\tau] + \sum_{p=\tau+1}^{j} \beta[k,j,p]\sigma_p\Big) + \tag{2.16}$$

$$+ \sum_{i=j+1}^{t} \Big(1 - (\beta[k,i,\tau] + \sum_{p=\tau+1}^{i} \beta[k,i,p]\sigma_p)\Big) - (1 + (t-j) - 1),$$

$$j = \tau-1,\dots,t,\ \forall \boldsymbol{\sigma} \in U$$

$$0 \leq u[k,t,j] \leq 1, \qquad j = \tau-1,\dots,t$$

$$\sum_{j=\tau-1}^{t} u[k,t,j] = 1$$

$$\sum_{k=1}^{K}\Big(\beta[k,t,\tau] + \sum_{p=\tau+1}^{t} \beta[k,t,p]\sigma_p\Big) \geq 1, \quad t = \tau,\dots,N,\ \forall \boldsymbol{\sigma} \in U$$

$$0 \leq \beta[k,t,\tau] + \sum_{p=\tau+1}^{t} \beta[k,t,p]\sigma_p \leq 1, \quad t = \tau+1,\dots,N, k = 1,\dots,K,\ \forall \boldsymbol{\sigma} \in U$$

$$\beta[k,\tau,\tau] \in \{0,1\}, \qquad k = 1,\dots,K$$

$$u[k,\tau-1,\tau-1] = 1; \quad \beta[k,\tau-1,\tau] = 1, \qquad k = 1,\dots,K.$$

The resulting optimization problem (2.16) is a relaxation of the robust counterpart

(2.14), since products of binary variables $\mathbf{y}$ defining $u[k,t,j]$ in (2.12) are now approximated by the corresponding affine expressions in terms of continuous variables $\boldsymbol{\beta}$, (see (2.15)), as opposed to the binary variables $\mathbf{y}$. The last group of constraints in (2.16) replaces the convention $y_k^{\tau-1} = 1$ used in formulation (2.14).

The optimization problem (2.16) yields an optimal solution $\boldsymbol{\beta}^*$ that uniquely identifies the optimal server assignment at the current time-step τ. Namely, if $\beta^*[k,\tau,\tau] = 1$ for some $k \in \{1,\ldots,K\}$, then the server k is assigned to the request σ_τ.

The truncated problem formulation AARO(T) is defined similarly to RO(T) as described in Section 2.3.1, i.e., at time-step τ we restrict the number of time periods in the optimization problem (2.16) from $t = \tau,\ldots,N$ to $t = \tau,\ldots,\min(\tau+T,N)$.

Let us denote the feasible set in formulation (2.16) with up to T future time periods ($t = \tau,\ldots,\min(\tau+T,N)$) by $\mathcal{F}(\tau,T,\hat{\mathbf{x}}^{\tau-1},K,N,U)$, where paramaters $\tau,T,\hat{\mathbf{x}}^{\tau-1},K,N,U$ uniquely determine the optimization problem (2.16). In this case, a short representation of the AARO(T) problem (2.16) that we will use in subsequent sections is as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta},\mathbf{u},\mathbf{v}} \quad & \sum_{k=1}^{K}\sum_{t=\tau}^{N} v_k^t \\ \text{s.t.} \quad & \boldsymbol{\beta},\mathbf{u},\mathbf{v} \in \mathcal{F}(\tau,T,\hat{\mathbf{x}}^{\tau-1},K,N,U). \end{aligned} \tag{2.17}$$

2.4 Holistic Adaptive Robust Optimization Approach

Both the WFA and AARO approaches for the K-server problem have their own merits. The main idea behind the design of the WFA is to reconsider past decisions in order to help find a high-quality assignment for the current one. On the other hand, the AARO method exploits information about future requests using uncertainty set in order to improve the current decision. The main purpose of this section is to introduce a new *holistic* method for solving multiperiod optimization problems which simultaneously incorporates information from the past and assumptions about the future. This holistic adaptive robust optimization (HARO) approach is a combination of the WFA method (involving past observations) and the AARO method (modeling

future uncertainty in adaptive manner by means of robust optimization).

The HARO approach can be seen as a natural extension of WFA in the following way. The basic definition of WFA contains a "greedy" term:

$d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau)$ in definition (2.3), or equivalently $\sum_{k=1}^{K} y_k^\tau \cdot d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau)$ in formulation (2.8),

which contains only information about the location of the current request $\hat{\sigma}_\tau$ and provides the distance that server will travel at the current time-step τ. Within the HARO framework, the uncertainty set U now contains additional information about future requests $\boldsymbol{\sigma}$ (not available before), and it is now possible to augment this "greedy" term. Instead of describing only one immediate time period, the "greedy" term is replaced by an expression giving an uncertain cumulative distance traveled by all servers during the next $N - \tau$ stages indexed by $t = \tau + 1, \ldots, N$.

At each time-step τ (for $1 \leq \tau \leq N$), an optimal assignment k^* generated by the HARO algorithm is now defined as follows:

$$k^* = \arg \min_{k=1,\ldots,K} \Big\{ C_{OPT}(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \ldots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)) + d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau) + \\ + C_{AARO}\big((\sigma_{\tau+1}, \ldots, \sigma_N) \in U\big) \Big\}, \quad (2.18)$$

where the first two terms are identical to the definition in WFA (2.3). The last term

$$C_{AARO}\big((\sigma_{\tau+1}, ..., \sigma_N) \in U\big) = \sum_{k=1}^{K} \sum_{t=\tau+1}^{N} (v_k^t)^*$$

represents the total distance that servers will travel over the next $N - \tau$ time-steps ($t = \tau + 1, \ldots, N$) according to the optimal solution $\mathbf{v}^*$ of the AARO method from Section 2.3.2. The last time-step of WFA (indexed by τ in (2.3)) and the first time-step of AARO (indexed by τ in (2.16) as well) now both simultaneously represent the current time-step of the multiperiod optimization problem (2.18). This is the only time-step where time periods of WFA and AARO formulations overlap.

Let us define the HARO(w, T) method as a merger of the w-WFA and AARO(T) methods. More specifically, we formulate HARO(w, T) as a mixed binary optimization

model with $w + T + 1$ time-steps, where the WFA block has w periods, the AARO block has T periods, and the current time-step has index τ. In order to solve one stage of the HARO(w, T) problem at time-step τ we need the following input data:

1. The previous request locations: $\hat{\sigma}_{\max(\tau-w,1)}, \ldots, \hat{\sigma}_\tau$, where $\hat{\sigma}_\tau$ is the position of the current request to be served. The definition of the first time index $t = \max(\tau - w, 1)$ guarantees that it is always greater or equal than 1.

2. The current configuration of the system, that is, vector $\hat{\mathbf{x}}^{\tau-1} = (\hat{x}_1^{\tau-1}, \ldots, \hat{x}_K^{\tau-1})$ of server locations at the end of previous time-step $t = \tau - 1$.

3. The configuration of the system at the end of time-step $t = \max(\tau - w - 1, 0)$, that is, vector $\hat{\mathbf{x}}^{\max(\tau-w-1,0)}$ representing the starting configuration of the WFA block.

4. The description of the uncertainty set U of the form (2.9) for up to T future request locations $(\sigma_{\tau+1}, \ldots, \sigma_{\min(\tau+T,N)})$.

In the HARO(w, T) online framework, at each time-step τ (for $1 \leq \tau \leq N$) we solve the formulation (2.19), which is by construction a combination of the w-WFA and AARO(T) formulations:

$$\min_{\substack{\mathbf{x},\mathbf{y},\mathbf{z}\\ \boldsymbol{\beta},\mathbf{u},\mathbf{v}}} \quad \sum_{k=1}^{K} \sum_{t=\max(\tau-w,1)}^{\tau} v_k^t + \sum_{k=1}^{K} y_k^\tau \cdot d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau) + \sum_{k=1}^{K} \sum_{t=\tau+1}^{\min(\tau+T,N)} v_k^t$$

s.t. For $k = 1, \ldots, K$:

$$v_k^t \geq x_k^t - x_k^{t-1}, \quad t = \max(\tau - w, 1), \ldots, \tau$$

$$v_k^t \geq -(x_k^t - x_k^{t-1}), \quad t = \max(\tau - w, 1), \ldots, \tau$$

$$x_k^t = x_k^{t-1} + y_k^t\, \hat{\sigma}_t - z_k^t, \quad t = \max(\tau - w, 1), \ldots, \tau - 1$$

$$x_k^{\max(\tau-w,1)} = \hat{x}_k^{\max(\tau-w-1,0)} + y_k^{\max(\tau-w,1)} \left(\hat{\sigma}_{\max(\tau-w,1)} - \hat{x}_k^{\max(\tau-w-1,0)}\right) \tag{2.19a}$$

$$x_k^\tau = \hat{x}_k^{\tau-1} + y_k^\tau \left(\hat{\sigma}_\tau - \hat{x}_k^{\tau-1}\right) \tag{2.19b}$$

$$z_k^t \leq y_k^t, \quad t = \max(\tau - w, 1), \ldots, \tau - 1$$

$$
\begin{aligned}
& z_k^t \le x_k^{t-1}, \quad t = \max(\tau - w, 1), \ldots, \tau - 1 \\
& z_k^t \ge y_k^t + x_k^{t-1} - 1, \quad t = \max(\tau - w, 1), \ldots, \tau - 1 \\
& y_k^\tau = \beta[k, \tau, \tau] \qquad (2.19c) \\
& 0 \le x_k^t,\ z_k^t \le 1,\ y_k^t \in \{0, 1\}, \quad t = \max(\tau - w, 1), \ldots, \tau \\
& \sum_{k=1}^{K} y_k^t = 1, \quad t = \max(\tau - w, 1), \ldots, \tau - 1 \\
& \boldsymbol{\beta}, \mathbf{u}, \mathbf{v} \in \mathcal{F}(\tau, T, \hat{\mathbf{x}}^{\tau-1}, K, N, U). \qquad (2.19d)
\end{aligned}
$$

The formulation (2.19) has a similar structure as the formulation (2.8) which represented the WFA method. Equations (2.19a) and (2.19b) fix configurations of the system at time-steps $t = \max(\tau - w - 1, 0)$ and $t = \tau - 1$, respectively, as required by the w-WFA definition. The most important binary assignment variables for the current time-step $t = \tau$ are modeled by y_k^τ and $\beta[k, \tau, \tau]$, $k = 1, \ldots, K$ in the WFA and AARO blocks, respectively. These decision variables are consistent with each other due to equations (2.19c). The last group of constraints (2.19d) duplicates the AARO(T) block (2.17) complementing the MIO formulation of the HARO(w, T) algorithm.

The HARO approach is a computationally tractable method that is stable with respect to errors in the uncertainty set. As we will see in Section 2.7 where we present numerical experiments, the HARO framework demonstrates superior performance in terms of objective value as opposed to the existing online algorithms and the adaptive algorithms under uncertainty (RO, AARO).

2.5 Generalizations of the K-Server Problem

In the previous sections, we have considered the simplest version of the K-server problem on a one-dimensional continuous metric space in order to introduce our proposed mixed integer, robust and adaptive optimization approaches. In this section, we discuss extensions of the basic formulation of the K-server problem to more general settings.

Multidimensional continuous metric space

If the metric space is the unit cube of dimension $D \geq 2$, $S = [0,1]^D$, then all requests $\boldsymbol{\sigma}$ and server locations $\mathbf{x}$ become vectors of dimension D and we can adjust the dynamics equation (2.5) as follows:

$$x_{k,d}^t = x_{k,d}^{t-1} + y_k^t(\sigma_{t,d} - x_{k,d}^{t-1}), \quad d = 1, \ldots, D.$$

If the distance between points in the metric space S is induced by the first norm $\|\cdot\|_1$ or infinity norm $\|\cdot\|_\infty$, then all previous formulations (OPT, WFA, RO, AARO, HARO) remain linear and have almost identical form as before. The only difference is that now it is necessary to introduce variables

$$v_{k,d}^t = |x_{k,d}^t - x_{k,d}^{t-1}|, \quad d = 1, \ldots, D,$$

instead of just the one-dimensional counterparts v_k^t used when $D = 1$.

Moreover, for the case of the infinity norm $\|\cdot\|_\infty$, the distance traveled by all servers at time-step $t = 1, \ldots, N$, v_t, can be expressed using inequalities

$$v_t \geq v_{k,d}^t, \quad k = 1, \ldots, K, d = 1, \ldots, D,$$

instead of a sum $v_t = \sum_{k=1}^{K} \sum_{d=1}^{D} v_{k,d}^t$, as is the case for the first norm $\|\cdot\|_1$.

Finite metric space

We consider the setting when the metric space S consists of a finite number of fixed points $\{\alpha_l \,|\, l = 1, \ldots, L\} \subset [0,1]^D$, where D is the dimension of the embedding Euclidean space. In this setting, we replace the previously considered uncertain parameters $\boldsymbol{\sigma} = (\sigma_{\tau+1}, \ldots, \sigma_N)$ (representing the positions of requests) with new uncertain

parameters $\boldsymbol{\nu} = \{\, \nu_l^t \mid t = \tau+1, \ldots, N,\, l = 1, \ldots, L\}$ defined as follows:

$$\nu_l^t = \begin{cases} 1, & \text{if request } \sigma_t \text{ is at location } \alpha_l, \\ 0, & \text{otherwise.} \end{cases} \tag{2.20}$$

The connection between the old and new uncertain parameters is given by a linear equation

$$\sigma_{t,d} = \sum_{l=1}^{L} \alpha_{l,d} \cdot \nu_l^t, \quad t = \tau+1, \ldots, N,\, d = 1, \ldots, D. \tag{2.21}$$

Hence, in case of a finite metric space equipped with distance $d(x,y) = \|x-y\|_1$ or $d(x,y) = \|x-y\|_\infty$ the HARO algorithm for the K-server problem can be modeled similarly to (2.19) using the linear change of variables (2.21). The only difference is that now a polyhedral uncertainty set should be expressed in terms of the parameters $\boldsymbol{\nu}$, rather than $\boldsymbol{\sigma}$, in the following way:

$$U_\nu = \{\mathbf{A}_\nu \boldsymbol{\nu} \leq \mathbf{b}_\nu,\, \boldsymbol{\nu} \geq \mathbf{0}\}, \tag{2.22}$$

where the new parameters $\boldsymbol{\nu}$ are supposed to be binary variables. We relax this integrality assumption and consider $0 \leq \nu_l^t \leq 1$ in order to use the same duality methodology as in (2.14), (2.16) and (2.19). In this setting, we treat the uncertain parameters ν_l^t as probabilities $\Pr(\sigma_t = \alpha_l)$ that request σ_t is located at location α_l.

Finite metric space with predefined distances

The network formulation of OPT and WFA for the K-server with given pairwise distances $d_{ll'} = d(\alpha_l, \alpha_{l'})$ in the form of minimum cost maximum flow problem is due to Chrobak et al. [1991] and Rudec et al. [2013], respectively. One may construct the robust counterparts of the nominal problems, for instance, introducing binary indicators

$$\eta[t_1, l; t_2, l'] := \nu_l^{t_1} \nu_{l'}^{t_2}.$$

However, the dimension of the uncertainty vector $\boldsymbol{\eta}$ in RO methods grows quadratically fast as a function of the number of locations L in the space S, and it makes these methods significantly slower than the formulations designed in Sections 2.3 and 2.4. This is why the case of a finite space S with predefined distances is beyond the scope of this chapter.

Weighted version of HARO

One of the natural extensions of the WFA online assignment policy is its weighted version WFA_λ defined as

$$k^* = \arg\min_{k=1,\dots,K} \Big\{ C_{OPT}(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \dots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)) + \lambda \cdot d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau) \Big\}, \tag{2.23}$$

where the positive parameter λ gives the relative priority of the "greedy" term over the work function term. It is known that in some settings, the incorporation of a nontrivial weight $0 < \lambda < 1$ can lead to a better performance for the online algorithm [Burley, 1996, Sitters, 2014]. In our case, we can introduce a weighted version of HARO_λ as

$$k^* = \arg\min_{k=1,\dots,K} \Big\{ C_{OPT}(\hat{\mathbf{x}}^0, \hat{\sigma}_1, \dots, \hat{\sigma}_\tau, \hat{\mathbf{x}}^\tau(k)) + d(\hat{x}_k^{\tau-1}, \hat{\sigma}_\tau) + \\ + \lambda \cdot C_{AARO}\big((\sigma_{\tau+1}, \dots, \sigma_\tau) \in U\big) \Big\}, \tag{2.24}$$

which may improve the performance of the HARO approach, for instance, in case of large uncertainty sets that are not very restrictive and the weight of the AARO term, λ, is chosen to be less than 1.

2.6 Construction of Uncertainty Sets

In this section, we give examples of uncertainty sets that can be used in practice and introduce one possible way to measure different levels of information about future requests. We will use this "varying information level" scale in order to empirically

demonstrate in Section 2.7 that the performance of all robust optimization methods (RO, AARO and HARO) improves as more information becomes available.

In order to quantify the amount of available information about a future request $\bar{\sigma}$, we simply use a metric-based notion of *neighborhood* $\mathcal{N}(\bar{\sigma}, r)$, defined as the ball centered around the true location of the request and of radius r, that is,

$$\mathcal{N}(\bar{\sigma}, r) = \{x \in S \,|\, d(x, \bar{\sigma}) \leq r\},$$

where $d(\cdot, \cdot)$ is a chosen distance of the metric space. For the case of continuous metric space $S = [0, 1]^D$ with the uniform norm $\|\cdot\|_\infty$ and radius $r \geq 0$, we have

$$\mathcal{N}(\bar{\sigma}, r) = \{x \in S \,|\, \|x - \bar{\sigma}\|_\infty \leq r\}.$$

In the context of a finite metric space, the neighborhood $\mathcal{N}(\bar{\sigma}, r)$ with integral radius $r \in \{1, \ldots, |S|\}$ consists of r closest neighbors to the point $\bar{\sigma}$ including itself.

Having specified the notion of the neighborhood, one may now consider the following family of uncertainty sets at any given time-step τ:

$$U_\tau(T, r) = \{(\sigma_{\tau+1}, \ldots, \sigma_{\tau+T}) \,|\, \sigma_t \in \mathcal{N}(\hat{\sigma}_t, r),\, t = \tau + 1, \ldots, \tau + T\}, \tag{2.25}$$

where T denotes time horizon and $\{\hat{\sigma}_t \,|\, t = \tau + 1, \ldots, \tau + T\}$ are true locations of future requests. Generally speaking, we do not know the exact locations of T future requests $\{\hat{\sigma}_t \,|\, t = \tau+1, \ldots, \tau+T\}$, but only their surrounding neighborhoods of radius r. In this case, the parameter r measures how precise the information is. Moreover, for both cases of continuous and finite metric spaces, the uncertainty set $U_\tau(T, r)$ can be represented as a bounded polyhedron of the form (2.9) or (2.22), expressed in terms of $\boldsymbol{\sigma}$ or $\boldsymbol{\nu}$, respectively.

In the next section, we use uncertainty sets of type (2.25) in order to show the increasing advantage of the HARO method as the amount of available information grows. At the same time, other types of polyhedral uncertainty sets describing future requests are also possible. One possible example is a Central Limit Theorem-

motivated set defined as

$$U = \Big\{(\sigma_{\tau+1}, \ldots, \sigma_{\tau+T}) \,|\, \Big|\frac{\sum_{t=\tau+1}^{\tau+T} \sigma_t - \bar{\mu}}{\bar{s}\sqrt{T}}\Big| \leq \Gamma\Big\},$$

where $\bar{\mu}$ and $\bar{s}$ denote empirical mean and standard deviation of requests $\boldsymbol{\sigma}$, respectively, and Γ is a predefined level of robustness [Bertsimas et al., 2011]. In case of a finite metric space, parameters $\boldsymbol{\nu}$ defined in (2.20) allow a practitioner to incorporate information about the distribution of future requests. For instance, inequalities

$$\underline{\rho} \leq \sum_{l:\, \alpha_l \in S_0} \nu_l^t \leq \bar{\rho}$$

state that at time-step t probability that request σ_t will emerge somewhere in a subset $S_0 \subset S$ is bounded by pre-specified parameters $\underline{\rho}$ and $\bar{\rho}$.

2.7 Numerical Experiments

In this section, we provide empirical evidence that the HARO method (2.19) designed in Section 2.4 for the K-server problem is computationally tractable, and almost uniformly outperforms existing online methods, as well as adaptive optimization methods. We show that the HARO algorithm gradually improves as more information about the future requests becomes available, and that it is stable with respect to possible mistakes in the assumptions about uncertain request locations.

The K-server problem as well as the proposed HARO method are characterized by many different parameters. The most significant ones are:

- The underlying metric space S (continuous or finite) with distance metric d;
- The number of servers K and their initial locations $\mathbf{x}_0$ in S;
- The distribution of requests (uniform or non-uniform) in space S;
- The number of locations L in case of finite metric space S;

- The length of the WFA optimization window w and the AARO optimization window T in the definition of the HARO(w, T) method;
- The uncertainty set U describing future requests;
- The computational time required to find high-quality solution.

The benchmark method in all our experiments is the optimal offline algorithm OPT. We implemented OPT and WFA as network flow problems according to Rudec et al. [2013]. Of all heuristic online algorithms presented in Section 2.2.1, only results for GREEDY and WFA are reported, since the rest of the methods (RAND, HARMONIC, BALANCE) resulted in all of our experiments in worse objective values. We coded all algorithms in Julia/JuMP [Lubin and Dunning, 2015, Dunning et al., 2017] and solved them with Gurobi 6.5 on a computer with 2.5Hz Intel Core i7 processor and 16GB of memory.

In all experiments, the overall sequence of requests corresponds to 1000 time-steps, while the length of the windows in the HARO(w, T) method varies from 10 to 30 for w, and from 5 to 15 for T. Each of the six methods that we test (OPT, GREEDY, WFA, RO(T), AARO(T), HARO(w, T)) generates a sequence of server assignments $\mathbf{y}^* = (y_1^*, \ldots, y_{1000}^*)$. Having obtained a sequence $\mathbf{y}^*$, one can calculate the total distance traveled by all servers on the 1000 requests. In all experiments, we report the empirical competitive ratios, that is the ratio between the cost incurred by a given algorithm and the optimal offline cost produced by the offline clairvoyant OPT. The length of the request sequence $N = 1000$ was empirically proved to be sufficient to statistically differentiate the behavior of the six methods considered in our experiments. The typical convergence rate of the empirical competitive ratios with respect to the length N is presented in Figure 2-1.

The metric space S is either $[0, 1]^2$ for the continuous case, or a set of L randomly generated points in $[0, 1]^2$ for the finite case. The distance between locations is calculated using the metric $d(x, y) = \|x - y\|_1$. The vector of initial locations for the servers $\mathbf{x}_0$ is generated uniformly at random. By default, the uncertainty sets $U_\tau(T, r)$ (2.25) do not include possible errors and the points $\hat{\sigma}_t$, $t = \tau + 1, \ldots, \tau + T$ represent

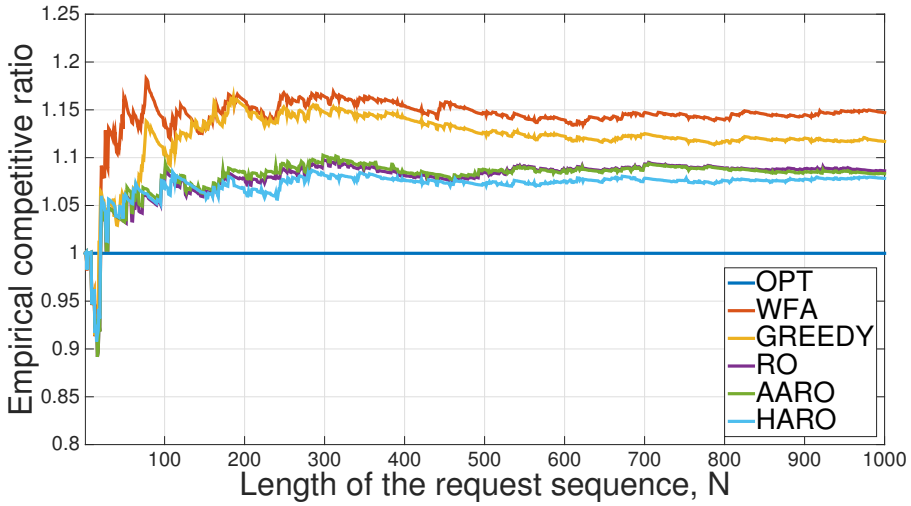


Figure 2-1: Convergence of the empirical competitive ratios with respect to N

the true locations of the future requests.

In case of the continuous metric space S, we test the performance of each RO-based algorithm with the same sequence of requests $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_1, \ldots, \hat{\sigma}_{1000})$, but different levels of information that is available about the next T requests. We model this varying level of information by the uncertainty sets $U_\tau(T, \frac{1}{n})$, for $n = 1, \ldots, 10$. For any fixed value of n, the decision maker knows that the next T uncertain requests σ_t belong to the corresponding intervals of type

$$\hat{\sigma}_t - \frac{1}{n} \leq \sigma_t \leq \hat{\sigma}_t + \frac{1}{n}. \tag{2.26}$$

In the context of finite metric space S, we run all online algorithms 12 times with identical sequence of requests $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_1, \ldots, \hat{\sigma}_{1000})$, but with gradually increasing amount of information represented by uncertainty sets

$$U_\tau\Big(T, \lceil L\big(1 - \frac{n-1}{12}\big)\rceil\Big).$$

Generally speaking, for any given $t = \tau+1, \ldots, \tau+T$, parameter $n = 1, \ldots, 12$ controls the number of neighboring locations that contain a true (but uncertain) location $\hat{\sigma}_t$. In both scenarios, the larger values of n correspond to more precise information about

the future requests.

In our experiments, we consider two types of request distributions over space S. The *uniform* distribution models scenario when requests appear in different locations of S with equal probability. The *non-uniform* distribution is defined as follows. When $S = [0,1]^2$, we consider a right top quadrant of the square $\bar{S} = \{(x,y) \,|\, 0.5 \leq x, y \leq 1\} \subset S$ and generate 50% of all requests in $\bar{S}$ (permuted randomly), and the remaining 50% of requests are sampled outside this subset $\bar{S}$. In case of finite metric space S, we first randomly select 25% of locations L and then again assign 50% of all requests to the specified subset of points $\bar{S}$.

In the next subsections, we conduct numerical experiments to empirically answer the following questions:

1. Does the HARO method generate more efficient solutions than other server assignment algorithms?

2. How stable the HARO method is with respect to potential noise in the description of the future?

3. How does the performance of the HARO(w,T) approach change with respect to parameters w and T?

4. Does the HARO method preserve its strong performance across various scenarios defined by the number of servers and locations and the distribution of requests?

2.7.1 Performance of the Methods

In this experiment, we demonstrate that the HARO(w,T) method outperforms the best existing online algorithms (GREEDY, WFA) and adaptive methods (RO(T), AARO(T)) in terms of the objective value for all tested levels of information about the future. We consider four different scenarios: when the metric space S is continuous or finite, and when distribution of requests is uniform or not, as defined in the beginning of this section.

Other parameters of the experiment are: $T = 5$, $w = 20$, $K = 7$, and $L = 25$ (when space S is finite). Given that $N = 1000$ is large enough, we report the performance results based on one realization of the sequence $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_1, \ldots, \hat{\sigma}_{1000})$ that yields an accurate representation of the behaviors of the six methods in expectation. The simulation results presented in Figure 2-2 infer the following conclusions:

1. All three proposed methods (RO(T), AARO(T), HARO(w, T)) improve as the amount of available information measured by parameter n grows. The performance of online heuristic algorithms GREEDY and WFA by construction does not depend on n what can be seen in Figure 2-2.

2. In all four scenarios, the asymptotic empirical competitive ratio of the RO-based methods (RO(T), AARO(T), HARO(w, T)) is approximately the same, and it is between 1.08 and 1.13 depending on the particular experiment. This level is strictly greater than 1, because even for large values of information precision n, our model incorporates information only about the very next $T = 5$ requests.

3. For small values of n, RO(T) method yields solutions that are too conservative with worse objective values. It illustrates the situation when there is almost no information about the next T requests, and RO(T) tries to find server assignments against the worst-case scenario. The adjustable counterpart of RO algorithm (AARO(T)) has a slightly better performance than RO(T) for small values of n, but after some point RO(T) and AARO(T) results are almost indistinguishable.

4. Given that the weighted version of HARO (2.24) with $\lambda = 0$ is the WFA and with λ large enough is the AARO algorithm, the HARO approach with small value of parameter $\lambda \approx 0$ outperforms a regular HARO approach with $\lambda = 1$ for small values of n.

5. AARO(T) quickly starts to perform at least as well as existing heuristics (GREEDY, WFA) at the information level $n \approx 3$. For both continuous and finite

space S this corresponds to eliminating approximately a quarter of irrelevant locations for the next $T = 5$ requests.

6. The HARO(w, T) algorithm, as a combination of stable WFA and adaptive AARO(T) methods, does not have high empirical competitive ratios for small values of n and at the same time improves with increasing of the available information. In most of the cases, even for large values of n, HARO(w, T) method outperforms AARO(T) algorithm.

7. In case of the finite metric space S, AARO(T) requires more information (compared to the case of continuous S) in order to approach the quality of existing methods GREEDY and WFA. This is why the benefit of the HARO(w, T) algorithm over AARO(T) becomes more pronounced for finite spaces.

8. The average computational time needed to find an optimal assignment produced by the HARO(w, T) algorithm at any given time-step is $2.35\,s$ when S is continuous and $4.11\,s$ when S is finite.

9. Overall, the empirical performance of HARO(w, T) algorithm is the best out of all considered methods (GREEDY, WFA, RO(T), AARO(T)) regardless of the information level n, characteristics of the metric space and the distribution of requests.

2.7.2 Sensitivity of the HARO Approach with Respect to Noise

The second experiment demonstrates the stability of HARO(w, T) algorithm with respect to noise in the description of the uncertainty sets $U_\tau(T, r)$. More precisely, we introduce a new parameter μ which is equal to the probability that the true location of future request $\hat{\sigma}_t$ is mistakenly replaced with some random point $\tilde{\sigma}_t(\mu)$ in S, for $t = \tau + 1, \ldots, \tau + T$. That is, we define the perturbed version of the uncertainty set as follows:

$$\tilde{U}^{\mu}_{\tau}(T, r) = \big\{(\sigma_{\tau+1}, \ldots, \sigma_{\tau+T}) \,|\, \sigma_t \in \mathcal{N}\big(\tilde{\sigma}_t(\mu), r\big),\ t = \tau + 1, \ldots, \tau + T\big\},$$

where for each $t = \tau + 1, \ldots, \tau + T$ the perturbed request $\tilde{\sigma}_t(\mu)$ is defined by

$$\tilde{\sigma}_t(\mu) = \begin{cases} \hat{\sigma}_t, & \text{with probability } 1 - \mu, \\ \alpha_l, & \text{with probability } \mu, \text{ for some randomly chosen } l \in \{1, \ldots, L\}. \end{cases} \tag{2.27}$$

For this experiment we select the finite space S with $L = 30$ locations and a uniform distribution of requests $\hat{\boldsymbol{\sigma}}$. We fix the level of available information $n = 4$, and set parameters $K = 10$, $N = 7$, $w = 15$. From Figure 2-3 we infer that when the distortion probability $\mu = 0$ and, therefore, there are no errors in provided information, then the HARO(w, T) method performs significantly better than GREEDY and WFA. Apparently, the mistakes in uncertainty set $\tilde{U}^{\mu}_{\tau}(T, r)$ aggravate the efficiency of the HARO(w, T) algorithm, but deterioration of the method performance is slow. The HARO(w, T) method starts to perform at the level of the best existing online algorithm when the probability of distortion μ is greater than 40% for each of the time-steps $t = \tau + 1, \ldots, \tau + T$.

We also tested HARO(w, T) algorithm with assymetric uncertainty sets $U_{\tau}(T, r)$, when the true location of request $\hat{\sigma}_t$ was not necessarily in the middle of the uncertainty interval as in (2.26), but rather just a random point of an interval of length $\frac{2}{n}$:

$$\hat{\sigma}_t - (1 - \theta_t)\frac{2}{n} \leq \sigma_t \leq \hat{\sigma}_t + \theta_t \frac{2}{n},$$

where parameters θ_t were generated uniformly at random between 0 and 1. This pivot in the definition of the uncertainty sets did not observably change the performance of the HARO(w, T) method presented in Figure 2-2.

2.7.3 Sensitivity of the HARO Approach with Respect to the Number of Time-steps

In this experiment, we demonstrate the influence of parameters w and T on the performance of the HARO(w, T) method, which are equal to the number of time-steps in AARO and WFA blocks, respectively. On average, the more information

about the future and the past is incorporated to the formulation (provided by larger values of w and T), the better decisions the HARO(w, T) method generates. However, we also discuss a trade-off between the efficiency of the HARO(w, T) algorithm and the computational time needed to find optimal assignments.

We consider continuous metric space S with $K = 5$ servers with a uniform distribution of requests and standard uncertainty sets (2.25). Computational results presented in Figure 2-4 imply that for any fixed value of T increasing the window size w improves the performance of HARO(w, T) algorithm, particularly for small values of information level n. Furthermore, there is a substantial improvement in the efficiency of the HARO(w, T) algorithm when length of an adaptive window grows from $T = 5$ to $T = 10$. First of all, when $T = 10$ the method on average has a steeper slope for $1 \leq n \leq 4$, and it converges to lower empirical competitive ratio 1.06, rather than 1.07 for HARO(w, T) with $T = 5$. At the same time, there is no observable change when parameter T is increased from 10 to 15. Probably, for $K = 5$ servers knowing the approximate locations of the next $T = 10$ requests is already sufficient.

It is worth mentioning, that the vast majority of variables in formulation (2.19) are auxiliary, and only vector $\mathbf{y}^*(\tau) = \{(y_k^\tau)^* \mid k = 1, \ldots, K\}$ determines which server $k = 1, \ldots, K$ should be assigned at the current time-step τ. A typical behavior of the solution progress is that the optimal assignment values $\mathbf{y}^*(\tau)$ are obtained by Gurobi in seconds and remain unchanged after this, while it takes from minutes to tens of minutes (depending on the instance size) to optimize over auxiliary variables and prove the optimality of vector $\mathbf{y}^*(\tau)$. Table 2.1 displays average computational times needed to find the optimal values of binary assignment variables $\mathbf{y}^*(\tau)$ for each of the variants of HARO(w, T) approach. As a result, we infer that the HARO(w, T) algorithm provides the decision maker with a high-quality recommendation within seconds for instances of practical size.

Table 2.1: Average time needed to find an optimal assignment for one time-step of the HARO(w, T) method

	T=5	T=10	T=15
w=10	1.21 s	6.75 s	10.52 s
w=20	2.2 s	17.96 s	28.4 s
w=30	5.17 s	27.33 s	43.97 s

2.7.4 Advantage of the HARO Approach

In this experiment, we demonstrate a substantial advantage of the HARO(w, T) algorithm over existing online methods (GREEDY, WFA) in various scenarios that differ by a number of servers K, locations L and distribution of requests in space S. In some sense, we extend the first experiment from Section 2.7.1 for finite metric space S to a large number of possible combinations of K and L, but present results in a more concise form. Instead of reporting a vector of length 12 describing the performance of the HARO(w, T) method for various $n = 1, \ldots, 12$, we present in Tables 2.2 and 2.3 a scalar which is equal to the average empirical competitive ratio of HARO approach:

$$\text{HARO (mean)} = \frac{1}{12} \sum_{n=1}^{12} \frac{\text{Cost of HARO}(w, T) \text{ with information level } n}{\text{Cost of OPT}}. \tag{2.28}$$

Having fixed optimization window sizes $w = 15$ and $T = 5$, we infer from Monte Carlo simulations that on average empirical competitive ratio of all methods (including HARO(w, T)) increases when number of servers K grows or number of locations L decreases. The empirical competitive ratios of the HARO(w, T) algorithm are higher when the distribution of requests $\hat{\boldsymbol{\sigma}}$ is non-uniform (Table 2.3) for large enough values of L, when compared to the corresponding experiments with uniformly distributed requests.

In all tested scenarios the HARO(w, T) algorithm has a similar behavior to one presented in Section 2.7.1 and it almost uniformly outperforms existing online methods. The mean advantage of the HARO(w, T) method defined in (2.28) varies from 2% ($K = 3, L = 25$, Table 2.2) to 26% ($K = 10$, $L = 15$, Table 2.3).

Table 2.2: Performance of the methods (uniform request distribution)

Number of locations, L	15			25			40		
Number of servers, K	3	5	10	3	5	10	3	5	10
GREEDY	1.23	1.41	1.58	1.17	1.37	1.55	1.15	1.23	1.33
WFA	1.24	1.43	1.51	1.24	1.45	1.48	1.21	1.27	1.39
HARO (mean)	1.17	1.35	1.40	1.15	1.29	1.31	1.11	1.21	1.30

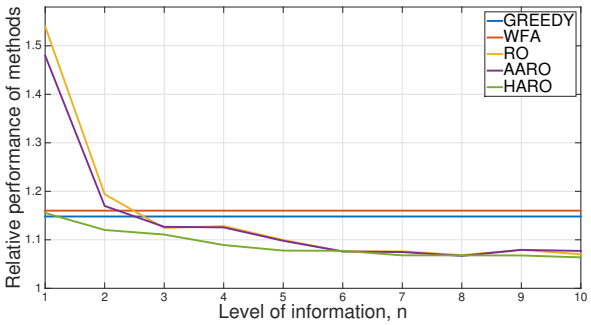
Table 2.3: Performance of the methods (non-uniform request distribution)

Number of locations, L	15			25			40		
Number of servers, K	3	5	10	3	5	10	3	5	10
GREEDY	1.18	1.36	1.56	1.32	1.46	1.69	1.26	1.41	1.43
WFA	1.19	1.34	1.39	1.22	1.47	1.67	1.23	1.39	1.44
HARO (mean)	1.15	1.33	1.30	1.16	1.39	1.53	1.16	1.28	1.36

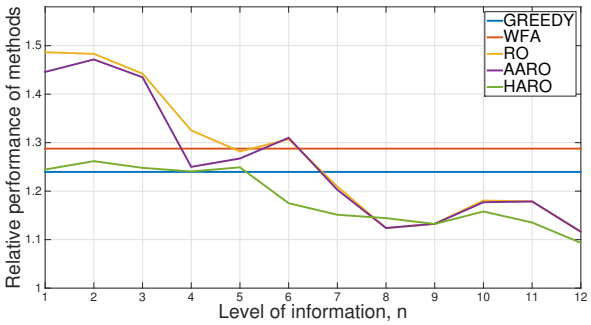
2.8 Conclusion

In this chapter, we have considered the K-server problem from the perspective of mixed integer, robust and adaptive optimization. We have designed new tractable MIO formulations for the offline version of the problem as well as for the online WFA. We have also introduced new algorithms that employ robust optimization methodology and include into consideration assumptions about future locations of requests.

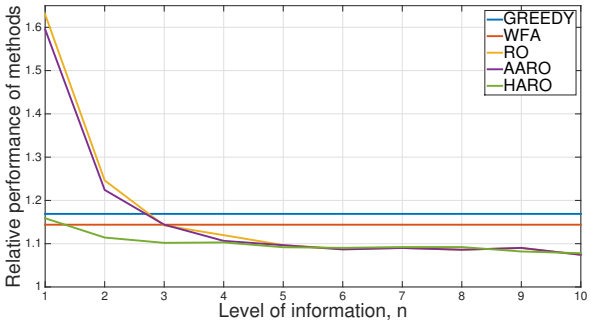
We have merged ideas behind an efficient WFA method and adaptive optimization techniques (AARO) and designed a new holistic algorithm (HARO) that simultaneously incorporates data describing the past and information about the future time periods. We have empirically demonstrated that HARO method is computationally tractable and almost uniformly yields lower objective cost than existing online heuristic algorithms and standard robust optimization approaches. Regardless of the level of available information about the future time periods, of the structure of the metric space, of the distribution of requests and number of servers, the HARO algorithm has led to lower empirical competitive ratio than other methods we have considered. Finally, we have shown that the HARO method is stable with respect to potential errors in the modeling of the uncertainty set associated with future requests.



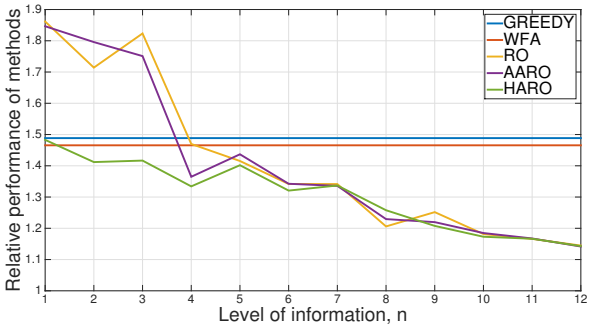
(a) S is continuous; uniform distribution of $\boldsymbol{\sigma}$



(b) S is finite; uniform distribution of $\boldsymbol{\sigma}$



(c) S is continuous; non-uniform distribution of $\boldsymbol{\sigma}$



(d) S is finite; non-uniform distribution of $\boldsymbol{\sigma}$

Figure 2-2: Empirical performance of online algorithms

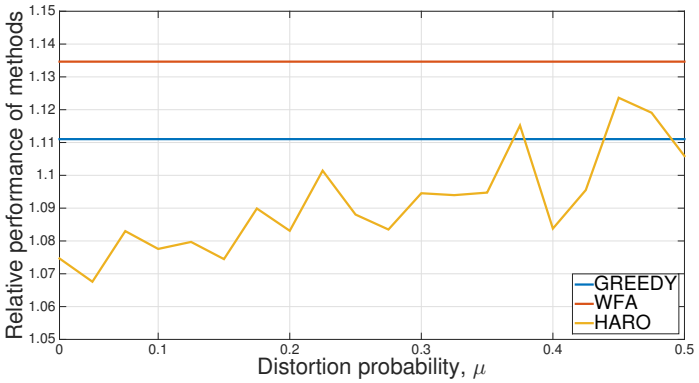
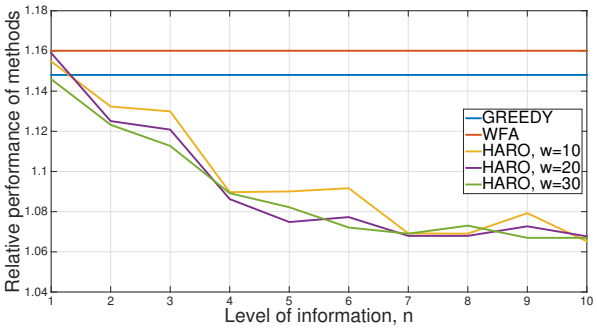
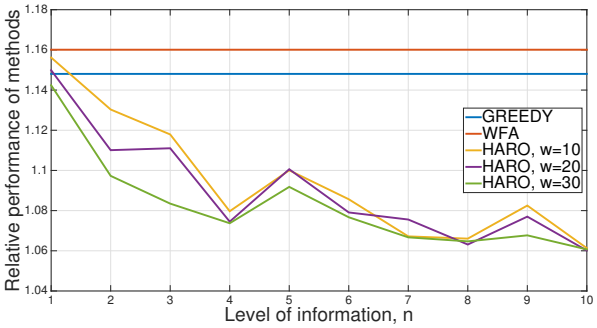


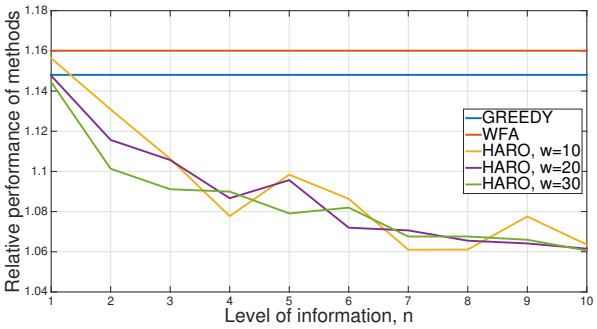
Figure 2-3: Performance of HARO with perturbed sets $\tilde{U}_\mu$



(a) Performance of HARO with $N = 5$ and $w \in \{10, 20, 30\}$



(b) Performance of HARO with $N = 10$ and $w \in \{10, 20, 30\}$



(c) Performance of HARO with $N = 15$ and $w \in \{10, 20, 30\}$

Figure 2-4: Sensitivity of the HARO(w, T) approach with respect to w and T

Chapter 3

The Fleet Defense Problem

3.1 Introduction

In this chapter, we consider generalizations of the asset-based weapon-to-target assignment (WTA) problem, first introduced by Manne [1958], which is one of the most fundamental defense-related applications of operations research. We model the situation when a set of ships that have access to various weapons need to decide how to deploy their weapons optimally in order to protect themselves against a variety of incoming threats. Each ship has a limited number of kinetic (hard-kill) and non-kinetic (soft-kill) weapons. The hard-kill weapons aims at physically destroying an incoming missile, while soft-kill weapons use electronics to deflect threats from ships, for instance, by jamming or creating an imaginary target. The use of soft-kill weapons provides flexibility but adds complexity to the allocation of weapons to targets since such weapon not only can simultaneously affect multiple threats, but can also adversely deflect a missile from one ship towards another one.

A fleet defense optimization problem has two main objectives. The first one is to maximize the assets survival probabilities, and the second one is to minimize the resource used. The second objective is important when there are potential attacks over several periods, and there is a need to conserve weapons to address future raids of incoming threats.

3.1.1 Previous research

The single period weapon resource coordination problem has been studied before. In most of the cases researchers addressed the problem of coordinating only hard-kill weapons, which is known as the classical WTA problem [Manne, 1958]. It assumes that each assignment of weapons to target has its own efficacy, and the decision maker tries to find an optimal allocation of available weapons against incoming threats. Existing formulations of the WTA problem include defense of a single ship as well as of multiple assets.

The formulation of the single ship defense problem can be written as an integer optimization problem with convex logarithmic objective [Ahuja et al., 2007]. This problem is nonlinear since several weapons may target the same threat. In this case, the probability of survival of the threat decreases proportionally to the *product* of the efficacies of the assigned weapons. The major drawback of this formulation is that it is not computationally tractable for even small instances with 20 weapons and 20 threats. This is why Ahuja et al. [2007] proposed network flow based heuristics. An alternative heuristic to the single ship defense problem involving enumeration of all possible assignments of multiple weapons against a particular threat is presented in [Bogdanowicz and Coleman, 2008b].

Another popular approach to address the single ship WTA problem is to assume that each threat can only be engaged by at most one weapon system and therefore eliminate non-linearity in the formulation. Under this simplifying assumption, various algorithms have been proposed based on auction theory [Cheung and Chung, 2010, Bertsekas, 1990] or elaborate greedy methods [Bogdanowicz and Coleman, 2008a].

The multi-asset WTA problem with hard-kill weapons is even more complex and nonlinear than the single-asset one since one threat may simultaneously target more than one ship. Hosein and Athans [1990] have suggested a heuristic method assuming that all weapons have the same efficacy, while other approximating algorithms without this assumption based on simulated annealing techniques have been presented in [Malhotra and Jain, 2001, Bisht, 2004]. There are also methods based on

network flow and auction theory that address the multi-asset WTA problem under the assumption that one threat can be engaged by at most one available hard-kill weapon [Bogdanowicz and Coleman, 2008a, 2007]. Murphey [2000] and Cai et al. [2006] provide comprehensive surveys of different variants of the WTA problem.

Becker et al. [2013] recently obtained some important results on the weapon coordination problem with multiple ships in a fleet. First, they modeled the exact centralized WTA problem with both hard-kill and soft-kill weapons in a form of a binary second-order cone problem (SOCP) [Boyd and Vandenberghe, 2004]. Optimization problems of this type can be efficiently solved for offline purposes by commercial solvers for instances of typical size [Gurobi Optimization Inc., 2016]. The authors also developed a heuristic based on linearization of second-order cone formulation that is fast enough for real-time applications. In this case, the optimal solution of the SOCP serves as a benchmark for the online heuristic. Finally, they considered a decentralized setting, for which they adapted their heuristic method and introduced a message passing algorithm. Their decomposition approach allows each ship to solve its local defense subproblem with updated data from other assets. However, the total amount of data that needs to be broadcast among ships is relatively large for such an online decision making, and therefore this protocol cannot be applied in many real world situations.

3.1.2 Contributions

In what follows, we denote by the fleet defense problem (FDP) any extension of the classical WTA problem that includes at least one additional feature: resource minimization objective, both hard-kill and soft-kill weapons, multiple consecutive raids or communication between agents in the decentralized setting. This chapter generalizes and improves some of the results on the FDP introduced in [Becker et al., 2013]. Our main contributions are as follows.

1. In Section 3.2, we develop an efficient algorithm solving the exact SOCP formulation of multi-asset FDP with hard-kill and soft-kill weapons in real time

for instances of practical size. Our approach uses lazy constraints techniques applied to the highly nonlinear mixed-binary fleet defense optimization problem.

2. We propose in Section 3.3 a new method to address the multiperiod FDP introducing binary adaptive survival indicators and leveraging modeling power of MIO. The new multiperiod formulation generates more efficient weapon assignments in case of a sequence of attacks and remains computationally tractable.

3. We extend in Section 3.4 the FDP to a decentralized setting under uncertainty using a Robust Optimization (RO) approach. In this setting, each ship optimizes by itself with some limited communication with other ships and limited information on the severity of the threats to its companions and on the efficacy of their weapons.

4. We demonstrate in Section 3.5 using extensive computational experiments that both single and multiperiod formulations of the FDP can be solved online for instances of typical size. We also show the effectiveness of the suggested RO-based communication and cooperation protocol in comparison with a completely decentralized setting with no communication among ships and the idealistic centralized coordination.

The lazy constraints methodology and the approach for modeling multiple periods (that we develop in Sections 3.2 and 3.2, respectively) may be useful for addressing other challenging problems, such as transportation problems with nonlinear system effects [Manne, 1958] and constrained resource allocation problems [Leboucher et al., 2013]. The way we introduce communication decision variables (Section 3.4) may also be useful to model cooperation between agents in multi-agent optimization frameworks as developed in [Terelius et al., 2011] and [Lobel et al., 2011].

3.2 An Optimization Model for the Single Period FDP

In this section, we describe a nonlinear MIO formulation of the single period FDP that incorporates both hard-kill and soft-kill weapons [Becker et al., 2013]. We also develop a new lazy constraints algorithm that efficiently solves this nonlinear problem by leveraging separating hyperplanes for polynomial constraints.

3.2.1 A Centralized Mathematical Formulation

Following Becker et al. [2013], we assume in this chapter that all threats have a lethality value of 1, that is a ship is destroyed if it is hit by any given threat. In a centralized setting, we also presume that all ships in the fleet have common knowledge about where the incoming threats are headed, as well as assets can accurately estimate their weapons efficacy against these threats.

Notation:

A. Indices

s	Ship index
i	Weapon index
j, k	Threat index

B. Sets

S	Ships in a fleet
T	Incoming threats
W	All available weapons
H	Hard-kill weapons
H_s	Hard-kill weapons on board ship s
K	Soft-kill weapons
K_s	Soft-kill weapons on board ship s
A_{ij}	Set of threats affected, if threat j is engaged by soft-kill weapon i
B_k	Set of all soft-kill interactions that afffect threat k

C. Data and Parameters

Q_{js}	Probability that threat j targets ship s
P_{ij}	Probability that hard-kill weapon i destroys threat j, if weapon i is fired against j
$R(j, s \mid i, k)$	Probability that threat j targets ship s, if soft-kill weapon i is fired against threat k
γ_s	Asset survivability threshold
α_s	Asset priority value
r_i	Weapon i firing cost
λ	Sufficiently large penalty for lack of survival probability

D. Decision variables

x_{ij}	Binary decision to fire weapon i against threat j
y_j	Binary decision to have no soft-kill interactions with threat j
D_{js}	Probability that threat j will leak through the defense and destroy ship s
w_s	Multiplicative slack variable for survival constraint of ship s

The fleet consists of S ships and needs to protect itself from the set of incoming threats T. Each ship s in the fleet has two sets of weapons: hard-kill (denoted by H_s) and soft-kill (K_s). The binary decision variable x_{ij} is equal to 1, if weapon $i \in W = \bigcup_{s \in S} (H_s \cup K_s)$ is to against the threat $j \in T$. An assignment of weapon i to any given threat incurs firing cost r_i. There is a threat targeting probability matrix $\mathbf{Q} = \{Q_{js}\}$ for $j \in T$, $s \in S$, as well as hard-kill weapon efficiency matrix $\mathbf{P} = \{P_{ij}\}$ for $i \in H$, $j \in T$.

We assume that an assignment of a soft-kill weapon $i \in K = \bigcup_{s \in S} K_s$ to threat $k \in T$ may affect other threats, and introduce in this case new threat targeting probabilities $\mathbf{R} = \{R(j, s \mid i, k)\}$ for $j \in T$, $s \in S$, $i \in K$, $k \in T$. For all soft-kill weapons $i \in K$ and threats $j \in T$, set A_{ij} consists of all threats affected, if weapon i is assigned to threat j. We denote the set of all soft-kill assignments $(i, j) \in K \times T$ that can affect threat $k \in T$ as $B_k = \{(i, j) : k \in A_{ij}\}$.

In this formulation, we do not allow any given threat j to be affected by more than one soft-kill weapon. In order to model this, we introduce auxiliary binary variable

y_j equal to 1, if threat j is not affected by any of soft-kill weapons. We also denote the probability that threat j hits an asset s by $D_{js} \in [0,1]$, for $j \in T$, $s \in S$.

Each ship $s \in S$ in the fleet has a couple of predefined and globally known parameters: α_s is an asset priority that reflects the importance of the ship, and γ_s is a probabilistic survival threshold. We assume that ship s does not need additional protection if its survival probability is at least γ_s.

We introduce a continuous multiplicative slack variable w_s that indicates by how much the survival probability of ship s is below its survival threshold. If ship s is protected with probability at least γ_s, then $w_s = 1$, and w_s is strictly greater than one, otherwise (see Eq. (3.1c) below). The latter case implies a penalty in the objective function (see Eq. (3.1a) below). If the penalty λ is large enough, then the decision maker always prefers to fire an additional missile (if there is one available) to reduce a slack w_s, rather than to save this weapon.

Having introduced necessary notations, we formulate the centralized multi-asset single period FDP in the form of nonlinear mixed-binary optimization problem as follows:

$$\min_{\substack{\mathbf{x},\mathbf{y},\\ \mathbf{w},\mathbf{D}}} \quad \sum_{i\in W}\sum_{j\in T} r_i\, x_{ij} + \lambda \cdot \sum_{s\in S} \alpha_s (w_s - 1) \tag{3.1a}$$

$$\text{s.t.} \quad 1 \le D_{js}\Big((Q_{js})^{-1} \cdot y_j + \sum_{(i,k)\in B_j} \big(R(j,s \,|\, i,k)\big)^{-1} \cdot x_{ik} \Big) \prod_{i\in H} \Big(1 + \frac{P_{ij}\, x_{ij}}{1 - P_{ij}}\Big), \quad \forall s \in S, j \in T \tag{3.1b}$$

$$\gamma_s \le w_s \prod_{j\in T} (1 - D_{js}), \quad \forall s \in S \tag{3.1c}$$

$$\sum_{j\in T} x_{ij} \le 1, \quad \forall i \in W \tag{3.1d}$$

$$y_j + \sum_{(i,k)\in B_j} x_{ik} = 1, \quad \forall j \in T \tag{3.1e}$$

$$1 \le w_s, \quad \forall s \in S \tag{3.1f}$$

$$0 \le D_{js} \le 1, \quad \forall s \in S,\, j \in T \tag{3.1g}$$

$$x_{ij},\, y_j \in \{0,1\}, \quad \forall i \in W,\, j \in T. \tag{3.1h}$$

The objective function (3.1a) consists of two parts. The first double sum is equal to the total weapon firing cost, while the second one represents the total penalty in case some ships have survival probabilities below predefined thresholds. If problem data admit a solution with all ships having survival probabilities greater than or equal to their corresponding survival thresholds, then all auxiliary multiplicative slacks $w_s = 1$ for all $s \in S$, and the cost function of the FDP (3.1a) is to minimize total weapons used. In a stressed scenario, when it is impossible to protect all ships in the fleet with probabilities above their survival thresholds, the cost function models a preferential defense strategy trying to maximize survival probabilities of the most important ships of the fleet with higher priority ranks α_s.

Given that higher priority ships have larger values of parameter α_s, the optimizer always prefers to protect important assets to their survivability thresholds before less important ones.

The probability D_{js} that threat j will leak through the defense and destroy ship s is given by:

$$D_{js} = \left\{ (Q_{js})^{y_j} \prod_{(i,k)\in B_j} \Big(R(j,s \,|\, i,k) \Big)^{x_{ik}} \right\} \prod_{i \in H} (1 - P_{ij})^{x_{ij}}, \tag{3.2}$$

where Q_{js} denotes the initial probability that threat j targets ship s. If there are no soft-kill interactions with threat j, that is $y_j = 1$, then the second product in brackets

$$\prod_{(i,k)\in B_j} \Big(R(j,s \,|\, i,k) \Big)^{x_{ik}} = 1,$$

since $x_{ik} = 0$ for all $(i,k) \in B_j$ according to Eq. (3.1e), and therefore the targeting probability Q_{js} remains unchanged. But the initial targeting probability Q_{js} can be substituted with another value $R(j,s \,|\, i,k)$, if soft-kill weapon $i \in K$ is fired against threat $k \in T$, that is x_{ik}=1, and this interaction (i,k) is in a set of soft-kill assignments

B_j that affect threat j. The last product

$$\prod_{i \in H} (1 - P_{ij})^{x_{ij}}$$

in the right hand side of (3.2) defines the probability that threat j is not destroyed by any of the independent hard-kill weapons $i \in H$. Given that all decision variables x_{ij} and y_j in Eq. (3.2) are binary, an alternative way to the define destruction probability D_{js} is as follows:

$$D_{js} = \left\{ (Q_{js})^{-1} y_j + \sum_{(i,k) \in B_j} \Big(R(j, s \,|\, i, k) \Big)^{-1} x_{ik} \right\}^{-1} \cdot \prod_{i \in H} \Big(1 + \frac{P_{ij}\, x_{ij}}{1 - P_{ij}} \Big)^{-1}. \tag{3.3}$$

In order to be able to invert elements of matrices $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ in Eq. (3.3), we assume that all probabilities are between ε and $1 - \varepsilon$, for some small predefined value $\varepsilon > 0$. Thus, Ineq. (3.1b) expresses that the probability D_{js} that threat j destroys a ship s is bounded below by the probability that j is attacking asset s and it is not ruined by all hard-kill weapons fired against j.

Since the lethality value of all threats equals one, the probability that ship $s \in S$ survives in a raid is equal to the probability that ship is not damaged by all independent threats, which we can express as

$$\prod_{j \in T} (1 - D_{js}). \tag{3.4}$$

Given that the auxiliary multiplicative slack w_s is heavily penalized in the cost function (3.1a), we will always have $w_s = 1$ in an optimal solution for ship s, whose survival probability (3.4) is at least γ_s. Hence, Ineq. (3.1c) states the survivability threshold for ship s.

We express the requirement that each weapon can be engaged no more than once by Ineq. (3.1d). Equation (3.1e) states that each threat $j \in T$ can be affected by at most one soft-kill weapon. The last group of constraints (3.1f) – (3.1h) defines upper and lower bounds or imposes integrality for decision variables of the problem.

The only nonlinear constraints of the problem (3.1) are (3.1b) and (3.1c). Both of them can be easily reformulated in the following form:

$$1 \leq \prod_{i \in \mathcal{I}} u_i, \qquad \text{where variables } u_i \text{ are non-negative, and } \mathcal{I} \text{ is some finite index set.} \tag{3.5}$$

Becker et al. [2013] proved that inequality of the form (3.5) defines a convex set and has a representation as a projection of a system of rotated second-order cone constratints. Hence, the FDP (3.1) can be solved offline by commercial solvers (for instance, Gurobi Optimization Inc. [2016]) as a convex quadratically-constrained binary optimization problem [Boyd and Vandenberghe, 2004] within minutes depending on the instance size. While this approach yields an optimal solution to the FDP that can serve as a benchmark, it cannot be used in real time.

3.2.2 Lazy Constraints Algorithm for the FDP

In this subsection, we propose a new efficient algorithm for solving problem (3.1) leveraging the special structure of nonlinear inequalities (3.1b), (3.1c) and MIO lazy constraints techniques [IBM Knowledge Center, 2017].

Algorithm 3.1 LAZY method for the FDP

procedure LAZY($\mathbf{Q}, \mathbf{P}, \mathbf{R}, \boldsymbol{\alpha}, \boldsymbol{\gamma}$)
  Predefine computational tolerance $\delta > 0$.
  Solve relaxation of (3.1) with the omitted nonlinear constrains (3.1b), (3.1c).
  Denote the feasible set of the relaxed problem by $\mathcal{P}$ and its optimal solution by $\mathbf{X} = (\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*, \mathbf{D}^*)$.
  while solution $\mathbf{X}$ does not satisfy constraints (3.1b), (3.1c) with tolerance δ **do**
    Find a hyperplane π separating the current solution $\mathbf{X}$ from the feasible set $\mathcal{P}$.
    Update $\mathcal{P}$ by adding to the set of its constraints the separating inequality that represents π.
    Re-solve the relaxation of problem (3.1) with the augmented $\mathcal{P}$.
    Update the optimal solution $\mathbf{X}$.
  return $\mathbf{x}^*$, i.e., assignments of weapons to threats.

In order to implement LAZY algorithm, we next derive in closed form the separating

hyperplanes associated with nonlinear constraints (3.1b), (3.1c) of the form (3.5).

Lemma 3.1. Let us consider a convex set $\Sigma = \left\{ \mathbf{x} \in \mathbb{R}^n_+ \,|\, \gamma \leq \prod_{i=1}^{n} x_i \right\}$ for some $\gamma > 0$. Then any point $\mathbf{y} \in \mathbb{R}^n_+ \backslash \Sigma$ can be separated from set Σ by the separating hyperplane

$$\sum_{i=1}^{n} \frac{x_i}{\hat{y}_i} = n, \tag{3.6}$$

where point $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_n) \in \mathbb{R}^n_+ \cap \Sigma$ is defined in one of the following ways:

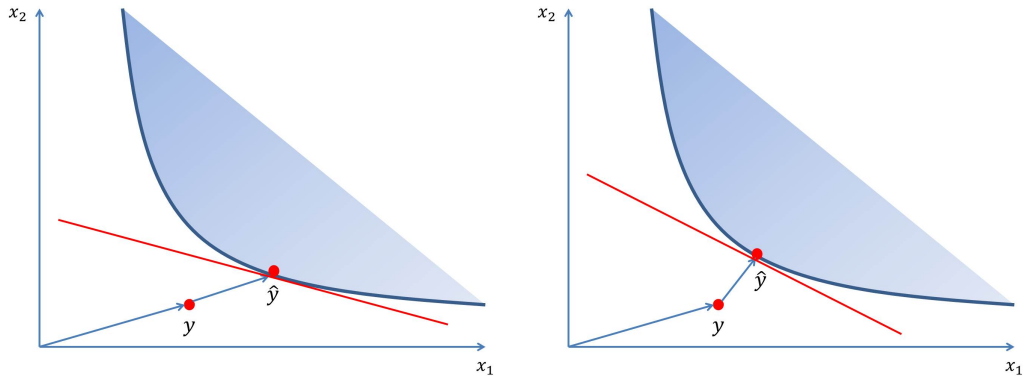
$$1) \quad \hat{y}_i = y_i\, t^*, \;\; \text{for} \;\; i = 1, \ldots, n \;\; \text{and} \; t^* = \sqrt[n]{\frac{\gamma}{y_1 y_2 \ldots y_n}}; \text{ or} \tag{3.7a}$$

$$2) \quad \hat{y}_i = \frac{1}{2}\big(y_i + \sqrt{y_i^2 - 4\mu^* \gamma}\big), \quad \text{for } i = 1, \ldots, n, \text{ and } \mu^* \text{ is the unique root} \tag{3.7b}$$

of the equation $g(\mu) = 0$, where

$$g(\mu) = \prod_{i=1}^{n} \left(\frac{y_i + \sqrt{y_i^2 - 4\mu\gamma}}{2} \right) - \gamma. \tag{3.8}$$

The first definition of point $\hat{\mathbf{y}}$ (3.7a) characterizes *linear* projection of point $\mathbf{y}$ onto the boundary of a convex set Σ, while the second one (3.7b) defines an *orthogonal* projection (Figure 3-1).



(a) Linear projection of point $\mathbf{y}$ (b) Orthogonal projection of point $\mathbf{y}$

Figure 3-1: Separating hyperplanes for set Σ.

Proof. Case 1 (3.7a). Let us introduce the function $f(\mathbf{x}) = \prod_{i=1}^{n} x_i - \gamma$, for $\mathbf{x} \in \mathbb{R}^n_+$. If point $\mathbf{y} \in \mathbb{R}^n_+$ does not belong to set Σ, then it is below the graph of the function

$f(\mathbf{x})$ (Figure 3-1a). Hence, we consider point $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ such that the components of $\hat{\mathbf{y}}$ are proportional to the components of $\mathbf{y}$, and point $\hat{\mathbf{y}}$ belongs to the surface $f(\mathbf{x}) = 0$, that is, the boundary of convex set Σ. In other words, for some $t > 0$ the following system of equations holds:

$$\begin{cases} \hat{y}_i = y_i\, t, \quad i = 1, \ldots, n \\ \prod_{i=1}^{n} \hat{y}_i - \gamma = 0. \end{cases}$$

We denote the unique solution of this system of equations with respect to variable t as

$$t^* = \sqrt[n]{\frac{\gamma}{y_1 y_2 \ldots y_n}} > 1.$$

We next find a tangent hyperplane to the smooth function $f(\mathbf{x})$ at point $\hat{\mathbf{y}}$ according to the formula

$$\nabla f(\hat{\mathbf{y}})^\top (\mathbf{x} - \hat{\mathbf{y}}) = 0.$$

Since $\frac{\partial f}{\partial x_i}(\hat{\mathbf{y}}) = \prod_{j \neq i} \hat{y}_j = \frac{\gamma}{\hat{y}_i}$ for all $i = 1, \ldots, n$, then the hyperplane has the form

$$\sum_{i=1}^{n} \frac{\gamma}{\hat{y}_i}(x_i - \hat{y}_i) = 0,$$

what is equivalent to (3.6). This tangent hyperplane indeed separates point $\mathbf{y}$ from the convex set Σ considering

$$\sum_{i=1}^{n} \frac{\gamma}{\hat{y}_i}(y_i - \hat{y}_i) = \sum_{i=1}^{n} \frac{\gamma \cdot y_i}{\hat{y}_i}(1 - t^*) = \sum_{i=1}^{n} \frac{\gamma\,(1 - t^*)}{t^*} < 0.$$

Case 2 (3.7b). The orthogonal projection $\hat{\mathbf{y}}$ of the point $\mathbf{y}$ onto the convex set Σ is the unique solution of the optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \\ \text{s.t.} \quad & x_1 x_2 \ldots x_n = \gamma \\ & x_i \geq 0, \quad i = 1, \ldots, n. \end{aligned}$$

The Lagrangian function for this problem has the form

$$L(\mathbf{x}, \mu) = \frac{1}{2}\sum_{i=1}^{n}(x_i - y_i)^2 + \mu(x_1 x_2 \dots x_n - \gamma).$$

The Karush-Kuhn-Tucker optimality conditions [Boyd and Vandenberghe, 2004] imply that

$$x_i = y_i - \mu^* \frac{\gamma}{x_i}, \quad x_i > 0, \quad \text{for } i = 1, \dots, n,$$

while a non-positive optimal Lagrange multiplier μ^* can be found from the equation

$$\prod_{i=1}^{n} x_i = \prod_{i=1}^{n} \frac{y_i + \sqrt{y_i^2 - 4\mu\gamma}}{2} = \gamma.$$

Since $y_i \geq 0$ for $i = 1, \dots, n$ and $\gamma > 0$, it is easy to see that for $\mu \leq 0$ the function $g(\mu)$ defined in (3.8) is strictly monotonically decreasing as a composition of decreasing functions. Moreover, $g(0) = \prod_{i=1}^{n} y_i - \gamma < 0$ and

$$g\big(-\gamma^{\frac{2}{n}-1}\big) = \prod_{i=1}^{n} \frac{y_i + \sqrt{y_i^2 + 4\gamma^{\frac{2}{n}}}}{2} - \gamma \geq \prod_{i=1}^{n} \frac{\sqrt{4\gamma^{\frac{2}{n}}}}{2} - \gamma \geq 0.$$

Thus, there is a unique solution μ^* of the equation $g(\mu) = 0$ in the interval $[-\gamma^{\frac{2}{n}-1}, 0]$, that can be found numerically with binary search techniques. Having found the optimal value of Lagrange multiplier μ^*, we determine the orthogonal projection $\hat{\mathbf{y}}$ in closed form (3.7b). Finally, we construct a tangent hyperplane to the convex set Σ at $\hat{\mathbf{y}}$ of the form (3.6). Separation of point $\mathbf{y}$ by the tangent hyperplane from Σ is due to the inequality

$$\sum_{i=1}^{n} \frac{\gamma}{\hat{y}_i}(y_i - \hat{y}_i) = \sum_{i=1}^{n} \frac{2\gamma}{y_i + \sqrt{y_i^2 - 4\mu^*\gamma}} \frac{y_i - \sqrt{y_i^2 - 4\mu^*\gamma}}{2} =$$

$$\sum_{i=1}^{n} \frac{\left(y_i - \sqrt{y_i^2 - 4\mu^*\gamma}\right)^2}{4\mu^*} < 0,$$

that completes the proof since μ^* is strictly negative. $\square$

3.3 An Optimization Model for the Multiperiod FDP

This section develops a nonlinear MIO formulation of the FDP for the case of multiple consecutive attacks. The extended multiperiod setting has to address a tradeoff between securing the survival probabilities above the thresholds in the immediate attack and weapon conservation objectives for future attacks. We also model a possibility that some of the assets of the fleet can be destroyed in the current raid, and therefore there is no need to protect them in consecutive attacks. Furthermore, we assume that in the multiperiod FDP it is prohibited to use weapons of destroyed ships in subsequent raids.

Notation (continued):

A. Indices

τ	Time-step, attack number

B. Sets

T^τ	Incoming threats at time-step τ
A_{ij}^τ	Set of threats affected, if at time-step τ threat j is engaged by soft-kill weapon i
B_k^τ	Set of all soft-kill interactions at time-step τ that afffect threat k

C. Data and Parameters

Q_{js}^τ	Probability that at time-step τ threat j targets ship s
P_{ij}^τ	Probability that at time-step τ hard-kill weapon i destroys threat j, if i is fired against j
$R^\tau(j, s \mid i, k)$	Probability that at time-step τ threat j targets ship s, if soft-kill weapon i is fired against threat k
N	Number of time-steps (attacks)
M	Large enough number

D. Decision variables at time-step τ

x_{ij}^{τ} Binary decision to fire weapon i against threat j

y_{j}^{τ} Binary decision to have no soft-kill interactions with threat j

D_{js}^{τ} Probability that threat j will destroy ship s

w_{s}^{τ} Multiplicative slack variable for survival constraint of ship s

u_{s}^{τ} Binary destruction indicator equals 1 if s is destroyed before time-step τ

Similar to the single period setting, each of the N attacks (indexed by $\tau = 1, \ldots, N$) consists of a set of threats T^{τ}. We denote the probability that at time-step τ threat $j \in T^{\tau}$ targets ship $s \in S$ as Q_{js}^{τ}. The corresponding weapon efficacy matrices are represented by $\mathbf{P}^{\tau} = \{P_{ij}^{\tau}\}$ for all raids $\tau = 1, \ldots, N$, weapons $i \in W$ and threats $j \in T^{\tau}$. The possible changes in threat targeting probabilities induced by the implementation of soft-kill weapons are described by matrices $\mathbf{R}^{\tau} = \{R^{\tau}(j, s\,|i, k)\}$. Finally, sets of soft-kill interactions B_{k}^{τ} that affect threat $k \in T^{\tau}$ are defined as in Section 3.2.1 for all time-steps $\tau = 1, \ldots, N$.

The global objective of the multiperiod FDP is to protect its ships in all attacks in the most resource-efficient way, that is minimizing weapons used. If the ideal scenario when all ships are protected with survival probabilities above predefined thresholds is infeasible, we implement preferential defense strategy that ensures that the most valuable assets are defended with a higher priority. In order to model this strategy, we use the same decision variables $x_{ij}^{\tau}, y_{j}^{\tau}, w_{s}^{\tau}, D_{js}^{\tau}$ as in single period formulation (3.1) with the only difference that now they have an additional time-index $\tau = 1, \ldots, N$. The only new variables that we need for the multiperiod setting are binary destruction indicators u_{s}^{τ} that are equal to 1, if ship s is destroyed before the beginning of attack $\tau = 1, \ldots, N$. Thus, the multiperiod FDP is given by a similar to (3.1) formulation:

$$\min_{\substack{\mathbf{x},\mathbf{y},\mathbf{w}\\ \mathbf{u},\mathbf{D}}} \quad \sum_{\tau=1}^{N} \left(\sum_{i \in W} \sum_{j \in T^{\tau}} r_i \, x_{ij}^{\tau} + \lambda \cdot \sum_{s \in S} \alpha_s (w_{s}^{\tau} - 1) \right) \tag{3.9a}$$

$$\text{s.t.} \quad 1 \le D_{js}^{\tau} \Bigg((Q_{js}^{\tau})^{-1} \cdot y_{j}^{\tau} + \sum_{(i,k) \in B_{j}^{\tau}} \big(R^{\tau}(j, s \,|\, i, k) \big)^{-1} \cdot x_{ik}^{\tau} \Bigg) \prod_{i \in H} \Big(1 + \frac{P_{ij}^{\tau}}{1 - P_{ij}^{\tau}} \, x_{ij}^{\tau} \Big),$$

$$\forall\, \tau = 1, \ldots, N;\; s \in S,\; j \in T^{\tau} \tag{3.9b}$$

$$\gamma_s \leq \left(w_s^\tau + M u_s^\tau\right) \prod_{j \in T^\tau} (1 - D_{js}^\tau), \quad \forall\, \tau = 1, \ldots, N;\ s \in S \tag{3.9c}$$

$$\sum_{\tau=1}^{N} \sum_{j \in T^\tau} x_{ij}^\tau \leq 1, \quad \forall i \in W \tag{3.9d}$$

$$y_j^\tau + \sum_{(i,k) \in B_j^\tau} x_{ik}^\tau = 1, \quad \forall\, \tau = 1, \ldots, N;\ j \in T^\tau \tag{3.9e}$$

$$u_s^1 = 0, \quad \forall s \in S \tag{3.9f}$$

$$\frac{w_s^\tau - 1}{M} \leq u_s^{\tau+1} \leq M(w_s^\tau - 1), \quad \forall\, \tau = 1, \ldots, N-1;\ s \in S \tag{3.9g}$$

$$u_s^\tau \leq u_s^{\tau+1}, \quad \forall\, \tau = 1, \ldots, N-1;\ s \in S \tag{3.9h}$$

$$x_{ij}^\tau \leq 1 - u_s^\tau, \quad \forall \tau = 1, \ldots N;\ s \in S,\ i \in H_s \cup K_s,\ j \in T^\tau \tag{3.9i}$$

$$1 \leq w_s^\tau, \quad \forall\, \tau = 1, \ldots, N;\ s \in S \tag{3.9j}$$

$$0 \leq D_{js}^\tau \leq 1, \quad \forall\, \tau = 1, \ldots, N;\ s \in S,\ j \in T \tag{3.9k}$$

$$x_{ij}^\tau,\ y_j^\tau, u_s^\tau \in \{0, 1\}, \quad \forall\, \tau = 1, \ldots, N;\ i \in W,\ j \in T,\ s \in S. \tag{3.9l}$$

The objective function (3.9a) models the preferential defense strategy, while each constraint of type (3.9b) defines a lower bound on the probability $D_{j,s}^\tau$ that threat $j \in T^\tau$ ruins ship s at time-step τ. Similarly as before, Ineq. (3.9c) states the survivability threshold for asset s at time-step τ. The only difference is that now we substitute multiplicative slack w_s^τ with a sum

$$w_s^\tau + M u_s^\tau, \tag{3.10}$$

where M is a large enough number. If ship s is not destroyed at the beginning of time-step τ and we need to protect it, that is $u_s^\tau = 0$, then factor (3.10) does not change and equals w_s as before (3.1). At the same time, if ship s was destroyed in one of the preceding raids, that is $u_s^\tau = 1$, then Ineq. (3.9c) does not impose any constraints on the remaining decision variables $\mathbf{w}, \mathbf{D}$ and is automatically satisfied. It means that we do not need to defend already destroyed ships.

Inequalities (3.9d) express that we cannot fire a weapon more than once, while equations of type (3.9e) state that at any given time-step $\tau = 1, \ldots, N$ each threat

$j \in T^\tau$ can be affected by at most one soft-kill weapon to avoid interdependencies.

Constraints (3.9f) – (3.9h) define destruction indicators u_s^τ. In the beginning of the first raid $\tau = 1$ all ships in the fleet are operating (3.9f). If at time-step τ ship s is protected with probability at least γ_s, then $w_s = 1$ and $u_s^{\tau+1} = 0$ (3.9g). At the same time, if $w_s^\tau > 1$, that is, ship s is not defended with probability at least γ_s at time-step τ, then Ineq. (3.9g) force the destruction indicator $u_s^{\tau+1}$ to switch its value to 1. In this case, we consider asset s to be destroyed in all consecutive raids $\tau + 1, \ldots, N$, which is represented by the set of constraints (3.9h).

Inequalities (3.9i) prohibit to use weapons of destroyed assets. Finally, we define upper and lower bounds or impose integrality for decision variables of the problem in the last group of constraints (3.9j) – (3.9l).

The decision maker should solve multistage optimization problem (3.9) updating uncertain parameters in the end of each attack. More precisely, binary survival indicators u_s^τ for $s \in S$ and $\tau = 1, \ldots, N$ are based on estimated survival probabilities (3.9c). At the same time, it is possible that ship s will be destroyed even if its survival probability is higher than the corresponding threshold γ_s, and vice versa. On the other hand, in the end of each attack τ a true ship survivability vector $\hat{\mathbf{u}}^\tau = (\hat{u}_1^\tau, \ldots, \hat{u}_S^\tau)$ becomes available for the decision maker, that allows her to substitute an estimated vector $\mathbf{u}^\tau$ with the accurate one $\hat{\mathbf{u}}^\tau$.

The only nonlinear constraints (3.9b) and (3.9c) in optimization problem (3.9) again have the form of (3.5). This is why it is possible to solve the multiperiod FDP as a convex mixed-binary SOCP. Moreover, we will demonstrate in Section 3.5 that it is possible to solve instances of (3.9) of typical size in real time leveraging lazy constraints algorithm designed in Section 3.2.2.

3.4 Decentralized Approach for the Fleet Defense Problem

The optimization problems introduced in Sections 3.2 and 3.3 are designed for the setting when all data about threats and weapons are globally known to all ships. However, in practice it is not always the case since each ship possesses some local information about the field of action. Furthermore, there is usually not enough time to broadcast all the local data to other ships, allow everyone to solve the same centralized optimization problem and execute the local part of the global optimal solution. This is why designing a distributed approach is important for practical applications of the FDP. In this setting, solving the global optimization problem is similar to many multi-agent communication and coordination problems, for instance, [Ren and Sorensen, 2008, Sariel and Balch, 2006, Maza et al., 2011], when several independent players cooperate in order to achieve a common goal.

In this section, we introduce a decentralized algorithm for solving the FDP leveraging and improving ideas formulated in [Becker et al., 2013]. The authors suggested an iterative cooperation protocol utilizing so-called leakage and linkage matrices that should be broadcast. These messages contain information about local weapon assignments as well as effectiveness parameters and allow other ships to update their local decisions in order to improve fleet-wide objectives. The main disadvantage of the existing communication protocol (ECP) from [Becker et al., 2013] is that the total size of the broadcast messages can be prohibitively large for practical applications. A brief comparison of the ECP and the new communication protocol (NCP) that we present here is summarized in Section 3.4.2.

Our new approach for distributed operations and communication is optimization-based. Namely, we do not broadcast all information about the field of action. In contrast, we determine essential fragments of information that should be shared among ships as an optimal solution of the specific two-stage RO problem (see Eq. (3.11) and Eq. (3.14) below).

We design the new protocol for the case of a single attack ($N = 1$) and the presence

of hard-kill weapons only ($K = \emptyset$).

Notation (continued):

A. Indices

s', s''	Ship index

B. Sets

T_s	Incoming threats that target ship s with non-zero probability
$U_{js'}(s)$	Uncertainty set for targeting probability $\tilde{Q}_{js'}(s)$
$\mathbf{U}(s)$	Cartesian product of sets $U_{js'}(s)$ with respect to j and s'
$\mathbf{f}^*$	Optimal values of communication decision variables

C. Data and Parameters

δ_s	Minimum accuracy of weapons of ship s
$\tilde{Q}_{js'}(s)$	Uncertain probability that threat j targets ship s' after the first stage of the protocol computed from the perspective of ship s
$\hat{Q}_{js'}(s)$	Upper bound on probability $\tilde{Q}_{js'}(s)$
$\hat{F}_s$	Maximum number of messages that ship s can broadcast

D. Decision variables

f_{js}	Binary decision of ship s to broadcast a message that it fires at least one weapon against threat j

3.4.1 The Cooperation Protocol

Following [Becker et al., 2013], we assume that all data of the FDP described in Section 3.2 are split into globally known and individually known blocks. On one hand, every ship s in the fleet possesses the following global data:

- Initial targeting probabilities: $Q_{js'}, \quad \forall s' \in S, \;\; j \in T$
- Probabilistic survival thresholds: $\gamma_{s'}, \quad \forall s' \in S$
- Asset priorities: $\alpha_{s'}, \quad \forall s' \in S$

- Predefined minimum accuracy of hard-kill interceptors: $\delta_{s'}, \quad \forall s' \in S$.

On the other hand, each ship s has some private data about its weapons that are not available to other ships:

- Efficiency of the weapons against all threats: $P_{ij}, \quad \forall i \in H_s, \quad j \in T$
- Cost of own weapons: $r_i, \quad \forall i \in H_s$.

Moreover, when ship s makes some local (possibly tentative) weapon assignment decisions, they are also not available for the rest of the fleet unless explicitly broadcast.

Having defined two types of data, the architecture of cooperation among ships under the assumption of limited time for communication can be outlined as three consecutive steps:

1. Each ship independently solves an optimization problem that simultaneously generates an initial self-defense strategy and determines which information about local weapon assignments should be broadcast.
2. Ships broadcast messages across the fleet.
3. Ships update their initial solutions based on revealed information from companions.

The details of each of the steps including explicit optimization formulations to be solved, connections between weapon assignments and messages to broadcast as well as updating rules for targeting probabilities are discussed below.

Step 1: Initial Solution

In this step, each ship calculates a preliminary solution that employs globally known data and local information about its own weapons. We propose an optimization-based variant of (3.1) for the initial solution that realizes a self-defense strategy. In this model, each ship tries to maximize the survival probabilities across the fleet with a much higher emphasis on defending itself. On top of this, each ship simultaneously

tries to identify which information about its weapon assignments would be the most useful to know for companions and therefore should be broadcast.

This decentralized optimization problem for a fixed ship $s \in S$ can be represented as follows:

$$z_1(s) = \min_{\substack{\mathbf{x},\mathbf{w},\\ \mathbf{f},\mathbf{D}}} \quad \sum_{i\in H_s}\sum_{j\in T} r_i\, x_{ij} + \lambda \cdot \alpha_s(w_s - 1) + \mu \cdot \sum_{s'\in S\setminus\{s\}} \alpha_{s'}(w_{s'} - 1) \tag{3.11a}$$

$$\text{s.t.} \quad 1 \le (Q_{js})^{-1} D_{js} \prod_{i\in H_s} \Big(1 + \frac{P_{ij}\, x_{ij}}{1 - P_{ij}}\Big), \quad \forall j \in T_s \tag{3.11b}$$

$$D_{js'} = Q_{js'}(1 - \delta_s f_{js}), \quad \forall s' \in S\setminus\{s\}, j \in T_{s'} \tag{3.11c}$$

$$\sum_{j\in T} f_{js} \le \hat{F}_s \tag{3.11d}$$

$$\sum_{i\in H_s} x_{ij} \ge f_{js}, \quad j \in T \tag{3.11e}$$

$$\gamma_{s'} \le w_{s'} \prod_{j\in T} (1 - D_{js'}), \quad \forall s' \in S \tag{3.11f}$$

$$\sum_{j\in T} x_{ij} \le 1, \quad \forall i \in H_s \tag{3.11g}$$

$$1 \le w_{s'}, \quad \forall s' \in S \tag{3.11h}$$

$$0 \le D_{js'} \le 1, \quad \forall s' \in S,\, j \in T \tag{3.11i}$$

$$x_{ij}, f_{js} \in \{0, 1\}, \quad \forall i \in H_s,\, j \in T. \tag{3.11j}$$

Optimization problem (3.11) employs new binary decision variables $\mathbf{f}$: we set variable f_{js} to be equal to 1, if a ship $s \in S$ decides to broadcast a message that it fires at least one weapon at threat $j \in T$. It is important to note that, due to communication constraints, ship s does not specify neither the total amount of weapons fired at j, nor the efficiency of the assigned weapons.

The objective function (3.11a) is similar to the objective from the centralized case (3.1a) with the only difference that now the importance of saving the ship s (measured by penalty factor λ) is much higher than the importance of protecting other companions (represented by a factor μ). Inequalities (3.11b) define destruction

probabilities D_{js} for the ship s as before (3.1b), conditional on the presence of hard-kill weapons only.

On the other hand, we define destruction probabilities $D_{js'}$ for the rest of the ships $s' \neq s$ in a new way (3.11c). The key idea is that another ship s' knows that ship s fires at least one weapon at threat $j \in T_{s'}$ only if ship s notifies everyone about it, which corresponds to the case $f_{js} = 1$. In this scenario, ship s' realizes that the initial targeting probability $Q_{js'}$ can be decreased. More precisely, it can be multiplied by $(1 - \delta_s)$, where δ_s is a globally known minimum accuracy of weapons of ship s. Thus, from the perspective of ship s, Eq. (3.11c) illustrates by how much ship s will help ship s', if it fires at least one weapon against threat $j \in T_{s'}$ and broadcasts a corresponding message.

Inequality (3.11d) states that the total number of messages that ship s can broadcast is at most $\hat{F}_s$, while constraint (3.11e) makes weapon assignments $\mathbf{x}$ and communication decisions $\mathbf{f}$ consistent with each other. The rest of the constraints are identical to the centralized formulation (3.1).

It is possible to solve the decentralized version of the FDP (3.11) with $K_s = \emptyset$ by lazy constraints techniques almost instantly (see Section 3.5). This formulation is computationally easier than the centralized version since there are only hard-kill weapons and a large number of nonlinear constraints of type (3.11b) are substituted with linear equations (3.11c). Since these computations are organized independently for all assets $s \in S$, then the initial solutions become available within a matter of seconds after a set of threats $T = \{T_s : s \in S\}$ is detected.

It is worth mentioning that the fully decentralized scenario of the FDP without communication when ships independently try to protect only themselves also has the form of optimization problem (3.11). In this case, it is sufficient to assume zero communication capacities $\hat{F}_s = 0$ for all $s \in S$ and zero penalty $\nu = 0$ for violations of survival of other ships.

Step 2: Broadcasting Messages

At the second stage of the protocol, ships broadcast information about their initial weapon assignments according to the optimal values of communication variables $\mathbf{f}^* =$

$\{f^*_{js} \,|\, j \in T, s \in S\}$ derived by the solutions of (3.11).

Step 3: Updating Weapon Allocation

Based on the messages from Step 2, each ship s may update estimations of targeting probabilities $\mathbf{Q}$ since some of the threats become less dangerous. We introduce uncertain parameters $\tilde{Q}_{js'}(s)$, for $j \in T$, $s \in S$ and $s' \in S \backslash \{s\}$ representing a probability that threat j targets ship s' after the first stage of the protocol measured from the perspective of ship s.

Given that variables $\mathbf{f}$ contain only a portion of information about weapon assignments and the number of these messages is limited, for all ships in a fleet targeting probabilities $\tilde{\mathbf{Q}}$ remain uncertain. We can introduce the corresponding uncertainty sets as

$$U_{js'}(s) = \big\{\tilde{Q}_{js'}(s) \,|\, 0 \leq \tilde{Q}_{js'}(s) \leq \hat{Q}_{js'}(s)\big\}, \tag{3.12}$$

where upper bound $\hat{Q}_{js'}(s)$ is determined by globally available data:

$$\hat{Q}_{js'}(s) = Q_{js'} \prod_{s'' \neq s} (1 - f^*_{js''}\delta_{s''}). \tag{3.13}$$

Equation (3.13) states that from the perspective of ship $s \in S$ threat $j \in T$ targets ship $s' \in S$ with initial probability $Q_{js'}$ discounted by $(1 - f^*_{js''}\delta_{s''})$ for any ship s'' different from s that fires at least one weapon against this threat.

We denote by $\mathbf{U}(s)$ the cartesian product of sets defined in (3.12) that aggregate estimations of targeting probabilities $\tilde{Q}_{js'}(s)$ from the perspective of ship s as follows

$$\mathbf{U}(s) = \Big\{ \big(\tilde{Q}_{11}(s), \ldots, \tilde{Q}_{TS}(s)\big) \,|\, \tilde{Q}_{js'}(s) \in U_{js'}(s) \text{ for } j \in T, s' \in S \Big\}.$$

In this step of the protocol, each ship s solves the FDP under uncertainty with respect to $\tilde{\mathbf{Q}}(s)$ incorporating additional information modeled by the set $\mathbf{U}(s)$ as follows:

$$z_3(s) = \min_{\mathbf{x},\mathbf{w},\mathbf{D}} \max_{\tilde{\mathbf{Q}}(s)\in\mathbf{U}(s)} \quad \sum_{i\in H_s}\sum_{j\in T} r_i\, x_{ij} + \lambda\cdot\alpha_s(w_s-1) + \mu\cdot\sum_{s'\in S\setminus\{s\}}\alpha_{s'}(w_{s'}-1) \tag{3.14a}$$

$$\text{s.t.} \qquad 1 \le \left(\tilde{Q}_{js'}(s)\right)^{-1} D_{js'} \prod_{i\in H_{s'}}\left(1+\frac{P_{ij}\,x_{ij}}{1-P_{ij}}\right), \quad \forall s'\in S,\, j\in T_{s'} \tag{3.14b}$$

$$\sum_{i\in H_s} x_{ij} \ge f^*_{js}, \quad j\in T \tag{3.14c}$$

$$\gamma_{s'} \le w_{s'}\prod_{j\in T}(1-D_{js'}), \quad \forall s'\in S \tag{3.14d}$$

$$\sum_{j\in T} x_{ij} \le 1, \quad \forall i\in H_s \tag{3.14e}$$

$$1 \le w_{s'}, \quad \forall s'\in S \tag{3.14f}$$

$$0 \le D_{js'} \le 1, \quad \forall s'\in S,\, j\in T \tag{3.14g}$$

$$x_{ij}\in\{0,1\}, \quad \forall i\in H_s,\, j\in T. \tag{3.14h}$$

Problem (3.14) is a nonlinear robust optimization problem [Ben-Tal et al., 2009, Bertsimas et al., 2011]. The objective function (3.14a) reflects that the decision maker controls only weapon assignment variables $\mathbf{x}$ (and auxiliary dependent variables $\mathbf{w}$ and $\mathbf{D}$), while targeting probabilities $\tilde{\mathbf{Q}}(s)$ are uncertain parameters.

Inequalities (3.14b) determine the destruction probabilities $D_{js'}$ as before with the only difference that now the first factor $\left(\tilde{Q}_{js'}(s)\right)^{-1}$ is not a constant, but rather an uncertain parameter. Given that each inequality (3.14b) should hold for all values of $\tilde{Q}_{js'}(s)$ in set $U_{js'}(s)$, it is easy to see that (3.14b) should remain feasible for the smallest possible value of $\left(\tilde{Q}_{js'}(s)\right)^{-1}$ or, equivalently, for the largest possible value of $\tilde{Q}_{js'}(s)$. This value is known to be equal to $\hat{Q}_{js'}(s)$ as defined in (3.13). Thus, the robust optimization problem (3.14) is equivalent to a regular optimization problem if all uncertain parameters $\tilde{Q}_{js'}(s)$ are substituted with the largest possible values

$\hat{Q}_{js'}(s)$, that represent the most risk-averse scenario, and maximization with respect to $\tilde{\mathbf{Q}}(s)$ in (3.14a) is omitted. As such, formulation (3.14) can be efficiently solved by a combination of standard RO techniques [Bertsimas et al., 2011] and the LAZY algorithm from Section 3.2.2.

Inequalities (3.14c) state that if ship s announced at the first step of the communication protocol to allocate at least one weapon to threat $j \in T$ (i.e. $f^*_{js} = 1$), then it should stick to the broadcast decision. The rest of the constraints are identical to formulations (3.1) and (3.11).

The implementation of the robust optimization methodology for solving the FDP (3.14) delivers two major benefits. First, it allows us to model uncertainty with respect to targeting probabilities $\tilde{\mathbf{Q}}(s)$ remaining after communication at Step 2 by means of sets $\mathbf{U}(s)$. Second, consideration of the most risk-averse scenario represented by inner maximization in (3.14) is a reasonable choice given the danger and liability involved when solving the fleet defense problem.

3.4.2 Discussion

The main differences between the ECP from Becker et al. [2013] and the NCP designed in Section 3.4.1 can be summarized as follows:

1. The ECP is based on a linearized heuristic of the binary second-order cone optimization problem (3.1) aiming to make it fast enough for real world scenarios, while the NCP efficiently solves exact nonlinear problems (3.11) and (3.14) leveraging lazy constraints techniques.

2. The ECP involves an iterative process of alternation between local computations and message broadcasting until a stopping condition is met. In contrast, the NCP considers a more stressful scenario with sharp time limits for communication, and therefore it has exactly three steps.

3. The architecture of the ECP assumes that information that ships share among each other (leakage and linkage matrices) is predefined and always the same

regardless of its importance. At the same time, a new protocol broadcasts not all data, but only its most useful pieces. Messages that ships transmit according to the NCP are defined as a part of an optimal solution of the optimization problem (3.11).

4. When the ECP is employed, then the total size of transmitted messages (consisting of several matrices) is fixed and may be too large for many practical situations. In contrast, the NCP allows the decision maker to control a number of messages that can be broadcast by tuning input parameters of the optimization problem (3.11) that represent communication capacities.

Another important advantage of the NCP designed in Section 3.4.1 is that its implementation can dramatically improve the global fleet objective

$$\mathrm{Z}_{total} = \sum_{s \in S} z(s), \tag{3.15}$$

where

$$z(s) = \sum_{i \in H_s} \sum_{j \in T} r_i \, x_{ij} + \lambda \cdot \alpha_s (w_s - 1) + \mu \cdot \sum_{s' \in S \setminus \{s\}} \alpha_{s'} (w_{s'} - 1). \tag{3.16}$$

The value of $z(s)$ represents the total weapon costs fired by ship s and a penalty if some of the ships in the fleet are not protected with required probabilities. We can compute this sum Z_{total} before and after implementation of the protocol. More precisely,

$$\mathrm{Z}_{total}^{start} = \sum_{s \in S} z_1(s) \quad \text{and} \quad \mathrm{Z}_{total}^{final} = \sum_{s \in S} z_3(s)$$

where $z_1(s)$ and $z_3(s)$ are the optimal solution values of (3.11) and (3.14), respectively.

The main idea of possible improvement is that the primary goal of the protocol is to increase survival probabilities of insufficiently protected ships and thereby reduce heavily penalized slack variables w_s (3.16). In other words, it is not necessarily true that implementation of the protocol leads to a reduction of total weapon costs in

comparison with the completely decentralized scenario, since some ships may spend additional weapons to protect insecure companions and therefore increase the total weapon expenses. At the same time, there is a dominating factor λ representing the paradigm that the fleet is much more concerned about survival of the ships than weapon costs. As a result, a solution with higher survival probabilities may result in a significant reduction of the objective value Z_{total}.

The cooperation protocol described in Section 3.4.1 may only decrease targeting probabilities for all ships in the fleet. Namely, Step 2 implies that $\hat{Q}_{js'}(s) \leq Q_{js'}$ for all ships $s, s' \in S$ and threats $j \in T$ according to (3.13). Thus, the main advantage of the suggested communication protocol is that survival probabilities of all ships may only increase and the objective values $z(s)$ of all ships may only decrease going from Step 1 to Step 3, which we prove formally in the following proposition.

Proposition 3.1. For any input data $\mathbf{Q}, \mathbf{P}, \hat{\mathbf{F}}, \mathbf{r}, \boldsymbol{\alpha}, \boldsymbol{\delta}, \lambda, \mu$ and any ship $s \in S$ the following inequality between objective values of optimization problems (3.11) and (3.14) holds:

$$z_3(s) \leq z_1(s). \tag{3.17}$$

Proof. In order to prove (3.17) we will show that the optimal solution $(\mathbf{x}^*, \mathbf{w}^*, \mathbf{f}^*, \mathbf{D}^*)$ of optimization problem (3.11) is feasible for problem (3.14). Combining definitions (3.12) and (3.13) we obtain that

$$\tilde{Q}_{js'}(s) \leq \hat{Q}_{js'}(s) \leq Q_{js}, \quad j \in T, \;\; s', s \in S. \tag{3.18}$$

For any fixed index $j \in T_s$, Ineq. (3.11b) defining destruction probability D_{js} at optimum has the following form

$$D^*_{js} \geq Q_{js} \Big(\prod_{i \in H_s} \Big(1 + \frac{P_{ij}\, x^*_{ij}}{1 - P_{ij}} \Big) \Big)^{-1}.$$

If targeting probabilities Q_{js} are substituted with lower values $\tilde{Q}_{js}(s)$ as in (3.14b), then the current solution $\mathbf{D}^*$ is still feasible due to

$$D^*_{js} \geq Q_{js}\Big(\prod_{i\in H_s}(1+\frac{P_{ij}\,x^*_{ij}}{1-P_{ij}})\Big)^{-1} \geq \tilde{Q}_{js}(s)\Big(\prod_{i\in H_s}(1+\frac{P_{ij}\,x^*_{ij}}{1-P_{ij}})\Big)^{-1},$$

which holds for all $\tilde{Q}_{js}(s) \in U_{js}(s)$ defined by (3.12).

Now we need to prove that the destruction probabilities $D_{js'}$ defined by Eq. (3.11c) are also feasible for inequalities (3.14b). Indeed, if $f^*_{js} = 0$, then inequality

$$1 \leq \Big(\tilde{Q}_{js'}(s)\Big)^{-1} Q_{js'}(1-\delta_s f^*_{js}) \prod_{i\in H_{s'}}\Big(1+\frac{P_{ij}\,x^*_{ij}}{1-P_{ij}}\Big) \tag{3.19}$$

is straightforward, given that (3.18) is true and each factor in the last product is greater or equal than 1. If $f^*_{js} = 1$, then Ineq. (3.11e) states that there exist at least one weapon, say $i_0 \in H_s$, that ship s fires at threat j, i.e. $x^*_{i_0\,j} = 1$. In this case, Ineq. (3.19) still holds, since

$$(1-\delta_s f^*_{js}) \prod_{i\in H_{s'}}\Big(1+\frac{P_{ij}\,x^*_{ij}}{1-P_{ij}}\Big) \geq (1-\delta_s)\Big(1+\frac{P_{i_0\,j}}{1-P_{i_0\,j}}\Big) = \frac{1-\delta_s}{1-P_{i_0\,j}} \geq 1,$$

by definition of minimum accuracy of hard-kill weapons δ_s.

The rest of the constraints in (3.14) are identical to formulation (3.11) and, therefore, remain valid. □

Combining result (3.17) from Proposition 3.1 and definition (3.15), we imply that $Z^{final}_{total} \leq Z^{start}_{total}$.

3.5 Computational Results

In this section, we empirically demonstrate the practical effectiveness of the MIO approach and lazy constraints techniques for both centralized and distributed counterparts of the FDP. Together with RO-based communication protocol, they allow the decision maker to obtain high-quality solutions of the FDP of typical size in real

time.

3.5.1 Advantage of the MIO Approach

In this experiment, we present a simple heuristic algorithm that can be employed as a benchmark for solving the FDP. We empirically validate that the MIO approach to the FDP developed in Section 3.2 significantly outperforms this heuristic algorithm providing a better protection for the fleet and spending less weapons for this defense.

One possible way to introduce a heuristic method (that we denote by GREEDY) is to sort all incoming threats from the most dangerous to the fleet to the least dangerous, and then sequentially assign the most efficient weapons against these sorted threats. A rigorous definition of the algorithm is as follows.

Algorithm 3.2 GREEDY method for the FDP

1: **procedure** GREEDY($\mathbf{Q}, \mathbf{P}, \boldsymbol{\alpha}, \boldsymbol{\gamma}$)
2: Set weapon assignment matrix $\mathbf{x} = \mathbf{0}$.
3: **while** there exists a ship s with survival probability below γ_s **and** there exists an unused weapon **do**
4: Sort threats $j \in T$ by $\sum_{s \in S} \alpha_s Q_{js}$ in a decreasing order.
5: Pick the first (and therefore the most dangerous) threat; denote it by j_0 .
6: Assign the most efficient unused weapon i_0 against j_0, i.e., $i_0 = \operatorname{argmax}_{i \in H} P_{ij_0}$ and set $x_{i_0 j_0} = 1$.
7: Update $Q_{j_0 s} \leftarrow Q_{j_0 s}(1 - P_{i_0 j_0})$ for $s \in S$.
8: **return** $\mathbf{x}$, i.e., assignments of weapons to threats.

In order to demonstrate that in practice the MIO approach generates much more efficient solutions to the FDP than GREEDY method, we consider an example of the fleet defense scenario as in [Becker et al., 2013]. This one time-period problem is described by the following parameters:

- Fleet: There are three ships of equal importance, i.e., $S = 3$ and $\alpha_s = 1$ for $s \in S$. Asset survival thresholds γ_s are generated uniformly between 0.7 and 0.95, and $\lambda = 1000$.

- Threats: There are nine threats in the field of action ($T = 9$). Targeting matrix

Q is defined by nonzero elements:

$$Q_{11} = Q_{13} = Q_{41} = Q_{42} = Q_{72} = Q_{73} = 0.5 \quad \text{and}$$

$$Q_{21} = Q_{31} = Q_{52} = Q_{62} = Q_{83} = Q_{93} = 1.$$

- Weapons: Each ship has 5 hard-kill missiles on board. Components of efficacy matrix **P** are uniformly generated at random between 0.5 and 1. Firing costs r_i are equal to 1 for all weapons $i \in W$.

We run 1000 Monte Carlo simulations solving generated FDP instances by the GREEDY and the MIO methods and compare solutions that both methods yield. In 46% of the scenarios the solutions turned out to be identical, while in 54% of the scenarios the solutions produced by the GREEDY method were less efficient. The GREEDY method allocated on average 14.5% more weapons for fleet defense than the MIO approach. Moreover, in 3.5% of the cases the MIO algorithm protected all the ships in the fleet with required probabilities, while the GREEDY algorithm allocated all available weapons and nevertheless some of the ships were left insecure.

3.5.2 Performance of the Lazy Constraints Method

This section compares the computational performance of the two formulations of the centralized fleet defense problem (3.1): the SOCP formulation, that can be derived from the results of paper [Becker et al., 2013], and a formulation with lazy constraints presented in Section 3.2.2. The lazy constraints algorithm may employ linear projection (3.7a) or orthogonal projection (3.7b) of a testing point. Therefore, in order to model the nonlinear constraints (3.1b) it is possible to use linear projections, orthogonal projections or both, i.e., there are three available variants. The same independent decisions can be made for constraints (3.1c), that gives us in total 9 different methods to solve the problem with lazy constraints techniques.

Based on the different ways to project testing points, we define the *parallelized method* as follows. For each instance of the FDP, we run 9 methods described above

in parallel, and terminate computation when one the arms solves the optimization problem to optimality.

In order to demonstrate that the parallelized lazy constraint method is remarkably faster than the SOCP method we consider the same FDP scenario as described in the previous experiment. The only difference is that now we add soft-kill weapons.

- Weapons: Each ship has 5 hard-kill missiles and 1 soft-kill weapon on board. Components of the efficacy matrix $\mathbf{P}$ are generated uniformly between 0.5 and 1. A soft-kill weapon can be fired by a ship only at threats that target this ship with probability 1. This assignment destroys a threat with probability 0.3 and redirects a threat to a neighboring asset with probability 0.2. It does not make any effect with probability 0.3. Firing costs r_i are equal to 1 for all weapons $i \in W$.

We evaluated two methods to solve the FDP - the SOCP and the parallelized lazy constraints algorithm - measuring computational time needed to find an optimal weapon allocation and to prove its optimality. The results of the second-order conic formulation are presented in Table 3.1, while computational times for an algorithm based on lazy constraints are reported in Table 3.2.

Table 3.1: Computational time of the SOCP method.

	Average time	Minimum time	Maximum time
First integer solution	32 s	3 s	73 s
Optimal solution found	**389** s	3 s	570 s
Optimality is proven	545 s	125 s	956 s

The algorithm with iteratively added linear separating hyperplanes has starkly better performance than SOCP method. In this particular example, an average acceleration of finding the optimal solution of the FDP is equal to $\frac{389}{3.93} = 98.9$ times.

If instead of the parallelized method we randomly choose one of the nine lazy constraints methods described in the beginning of this section, then the average time needed to find an optimal solution of this FDP is 6.4 s. It implies that even without solving this problem in parallel, we can guarantee a significant acceleration of computation.

Table 3.2: Computational time of the parallelized lazy constraints method.

	Average time	Minimum time	Maximum time
First integer solution	0.19 s	0.01 s	2.1 s
Optimal solution found	**3.93** s	0.4 s	7.4 s
Optimality is proven	16.91 s	4.1 s	27.8 s

3.5.3 Sensitivity Analysis

In this experiment, we explore the influence of major parameters (survival threshold, weapons efficiency and number of threats) on the computational time needed to solve the FDP. The default setting is defined as follows:

- Fleet: There is one ship $S = 1$ with priority $\alpha_1 = 1$. Number of periods $N = 1$, threshold γ is randomly generated between 0.7 and 0.95, penalty $\lambda = 1000$.
- Threats: The total number of threats $T = 5$ and components of targeting vector $\mathbf{Q} = (Q_{j1}, \ldots, Q_{j\,T})$ are generated uniformly at random between 0 and 1.
- Weapons: The number of hard-kill interceptors $H = 10$ with uniformly random efficiency between 0.5 and 1. There is also one soft-kill weapon $K = 1$, which deflects the threat with probability 0.6 and does not change anything with probability 0.4.

Having fixed default parameters, we sequentially perturb each of them to find out how it affects the computational time for solving the corresponding optimization problem. For each of the scenarios we generate 1000 Monte Carlo simulations and solve them with the parallelized method. In Table 3.3 we report 95% quantiles of computational time needed to find the optimal solution and to prove that this solution is indeed optimal. In other words, in 95% of all simulations the optimal solution was found and proved to be optimal faster than times indicated in the corresponding cells of the table. The time needed to obtain the first integer solution was always below 1 second, and therefore omitted.

Case 1: Changes in survival threshold γ. These empirical results allow us to make the following practical conclusion. If the survival threshold of ship s belongs, for

instance, to the interval [0.9, 0.95] and the decision maker terminates the computation of the FDP after 0.6 s, then with 95 % probability the best integer solution found by this time is optimal. Furthermore, we infer that the growth of survival threshold implies the growth of computational time since it becomes more difficult to protect a ship with a higher probability.

Table 3.3: Sensitivity of the computational time with respect to problem parameters.

Case 1: Range for γ	$[0.7, 0.75]$	$[0.7, 0.95]$	$[0.9, 0.95]$	$[0.95, 1]$
95% Quantile (Opt found)	0.17 s	0.43 s	**0.6** s	10.7 s
95% Quantile (Opt proven)	0.37 s	1.53 s	3.7 s	15.7 s
Case 2: Range for P_{ij}	$[0.5, 0.7]$	$[0.6, 1]$	$[0.75, 1]$	$[0.9, 1]$
95% Quant. (Opt found)	9.41 s	0.39 s	0.46 s	0.12 s
95% Quant. (Opt proven)	30.1 s	2.39 s	1.67 s	0.32 s
Case 3: Number of threats T	$T = 3$	$T = 5$	$T = 7$	$T = 9$
95% Quantile (Opt found)	0.06 s	3.93 s	14.7 s	26.4 s
95% Quantile (Opt proved)	0.11 s	16.9 s	28.1 s	30.2 s

Case 2: Changes in weapons efficiency P_{ij}. This case study validates that the growth of weapons accuracy naturally implies the simplification of the fleet defense optimization problem, what reduces the overall computational time.

Case 3: Changes in a number of threats T. The last section of Table 3.3 demonstrates the growth of computational time needed to solve more and more stressing scenarios with a sequentially increasing number of incoming threats.

The conducted numerical experiments make evident that the FDP can be solved by the parallelized method within a matter of seconds (that is, online) for the instances of practical size and reasonable values of underlying parameters. The more stressful the scenario becomes (described by higher survival probability thresholds, lower weapon efficiency or a larger number of incoming threats), the more computational time is needed to find the optimal solution.

3.5.4 Scaling Experiments

In this subsection, we discuss the scalability of the fleet defense optimization problem with respect to the size of the fleet and the total number of attacks. As opposed to

the previous experiment, we now increase all major parameters of the problem proportionally to one selected parameter, rather than change a single parameter keeping the rest of them fixed. The default setting is defined as follows:

- Fleet: Survival thresholds γ_s for $s \in S$ are uniformly generated at random between 0.7 and 0.95, asset priorities α_s for $s \in S$ are random integers from 1 to 10 and $\lambda = 1000$.

- Threats: Total number of threats is defined by a vector $(T^1, \ldots, T^N)$, where N is a number of subsequent attacks. Components Q^τ_{js} of targeting matrices $\mathbf{Q}^\tau$ are randomly generated for $\tau \in \{1, \ldots, N\}$, $j \in T^\tau$ and $s \in \{1, \ldots, S\}$ in such a way, that any missile j cannot target more than 3 ships.

- Weapons: Each ship has 5 hard-kill weapons per time period with a random efficiency P^τ_{ij} between 0.6 and 1. Moreover, each ship holds one soft-kill weapon per period. Each soft-kill weapon deflects a threat from the fleet with probability 0.3, it redirects a threat to another asset with probability 0.4 and does not affect the missile with probability 0.4.

Similarly to the previous experiment, we generate 1000 Monte Carlo simulations in order to measure 95% quantile of computational times needed to find an optimal solution and to prove its optimality.

Case 1: Scaling with respect to a number of ships S. This scenario is defined by a fixed number of attacks $N = 1$ and a changing size of the fleet S. A total number of hard-kill weapons H, soft-kill weapons K as well as a total number of incoming threats T depend on S as follows:

$$H = 5 \cdot S, \quad K = S \quad \text{and} \quad T = 3 \cdot S.$$

Case 2: Scaling with respect to a number of time periods N. In this setting, we keep a number of ships S fixed and equal to 3, while the number of subsequent attacks

N varies. We define major parameters of the FDP to be linear functions of N:

$$H_s = 5 \cdot N; \quad K_s = N \text{ for } s \in S\,; \quad T^\tau = 3 \text{ for } \tau \in \{1, \ldots, N\}.$$

Table 3.4: Computational time of the parallelized method.

Size of the fleet S	$S = 2$	$S = 4$	$S = 6$	$S = 8$
95% Quantile (Opt found)	1.23 s	6.01 s	15.4 s	22.8 s
95% Quantile (Opt proved)	25.4 s	30.3 s	41.3 s	63 s

Number of attacks N	$N = 1$	$N = 2$	$N = 3$
95% Quantile (Opt found)	0.15 s	0.84 s	1.81 s
95% Quantile (Opt proved)	0.26 s	2.42 s	8.54 s

Table 3.4 demonstrates one more time that the FDP can be solved within a matter of seconds for instances of typical size with reasonable sizes of a campaign S and a number of time periods N. The results from Table 3.4 also imply that the computational time needed to find the optimal solution of the FDP and to prove its optimality exhibits a moderate growth as a function of parameters S and N.

3.5.5 Advantage of the Multiperiod Approach

In this experiment, we consider the multiperiod scenario for the FDP described in Section 3.3. Our primary goal is to demonstrate that in this case the method based on the multiperiod formulations (3.9) significantly outperforms the method based on myopic single period formulations (3.1). More precisely, in case of a sequence of attacks (indexed by $\tau = 1, \ldots, N$), the decision maker at any given time-step τ can organize the fleet defense in two different ways:

Method A. She can solve the *single period* optimization problem of type (3.1) that comprises data describing only the current time period τ, i.e., $\mathbf{P}^\tau, \mathbf{Q}^\tau, \mathbf{R}^\tau$.

Method B. She can solve the *multiperiod* optimization problem of type (3.9) that comprises data describing the current and the future time periods, i.e., $\mathbf{P}^t, \mathbf{Q}^t, \mathbf{R}^t$ for $t = \tau, \ldots, N$.

The numerical experiment is defined as follows:

- Fleet: The size of the fleet $S = 4$ and ship priorities are $\alpha_s = s$ for $s = 1, \ldots, S$. The number of periods $N = 3$, thresholds γ_s are randomly generated between 0.7 and 0.95, and the penalty $\lambda = 1000$.

- Threats: The number of threats per period $T^\tau = 4$ for $\tau = 1, \ldots, N$. At each time-step $\tau = 1, \ldots, N$, there is one threat that targets a randomly chosen ship with probability 1, while the remaining threats can target up to two ships with probabilities generated uniformly at random.

- Weapons: The number of hard-kill interceptors $H = 11$ with uniformly random efficiency between 0.5 and 1. Each ship carries one soft-kill weapon, which destroys the threat with probability 0.3, redirects it to a neighboring ship with probability 0.4 and does not change anything with probability 0.3.

At the end of each time-step $\tau = 1, \ldots, N$ we simulate a possible destruction of the ships as follows. Let us assume that ship s was protected with a probability p_s^τ. If p_s^τ is at least as high as γ_s, then we assume that this ship is secure. Otherwise, the survival of ship s at the end of time-step τ is modeled by Bernoulli random variable ξ_s^τ with parameter p_s^τ.

We compare the performance of the methods A and B in terms of four characteristics describing the quality of the FDP solution:

1. The total number of ships that survive the sequence of attacks, $\sum\limits_{s \in S} \max\Big(\mathbb{I}\big\{w_s^N = 1\big\}, \mathbb{I}\big\{w_s^N > 1 \text{ and } \xi_s^N \le p_s^N\big\}\Big)$.

2. The total *weighted* number of surviving ships, where ship priorities are used as weights, $\sum\limits_{s \in S} \alpha_s \cdot \max\Big(\mathbb{I}\big\{w_s^N = 1\big\}, \mathbb{I}\big\{w_s^N > 1 \text{ and } \xi_s^N \le p_s^N\big\}\Big)$.

3. The total number of weapons fired at threats, $\sum\limits_{\tau=1}^{N} \sum\limits_{i \in W} \sum\limits_{j \in T^\tau} x_{ij}$.

4. The total number of weapons wasted due to the destruction of the ships.

Table 3.5 demonstrates the empirical results of one thousand Monte Carlo simulations. Given that Method B is exposed to more relevant information about the field of action, it is able to produce more efficient and strategic weapon assignments. We observe that the average number of surviving ships is higher and the average number of destroyed weapons is lower when Method B is employed.

Table 3.5: Comparison of methods solving the multiperiod FDP

Method	Ships survived	Weighted ships	Weapons fired	Weapon wasted
A: Single period	3.17	8.1	13.15	0.45
B: Multiperiod	3.59	8.95	13.37	0.17

3.5.6 Cooperation Protocol Results

In this experiment, we compare the performance of the three algorithms solving the FDP from the perspective of global fleet objectives:

1. Decentralized: In this setting ships in the fleet do not communicate with each other and execute independent self-defense plans.

2. Communication protocol: Ships in the fleet have some communication capability and cooperate according to the steps described in Section 3.4.

3. Centralized: All information about the field of action is globally known, so the exact solution of fleet-wide optimization problem is calculated and implemented.

Similarly to the previous experiment, we consider four characteristics that describe the quality of a FDP solution produced by an algorithm:

1. The total number of ships that survive, $\sum_{s \in S} \mathbb{I}\{w_s = 1\}$.

2. The total *weighted* number of surviving ships, where ship priorities are used as weights, $\sum_{s \in S} \alpha_s \cdot \mathbb{I}\{w_s = 1\}$.

3. The average value of a multiplicative slack variable, $\frac{1}{|S|} \sum_{s \in S} \frac{w_s - 1}{w_s}$. This value displays by how much on average a survival probability of a ship is below a corresponding survival threshold.

4. The total number of weapons spent, $\sum_{s \in S} \sum_{i \in W_s} \sum_{j \in T} x_{ij}$.

In this case study we consider a scenario defined by parameters similar to the previous experiments:

- Fleet: Number of attacks $N = 1$, the size of the campaign $S = 5$, thresholds γ_s for $s \in S$ are uniformly generated at random between 0.7 and 0.95, penalty constants $\lambda = 1000$ and $\nu = 10$, asset priorities $\alpha_s = s$ for $s = 1, \ldots, S$. Each ship can send up to 2 messages at Step 1 of the communication protocol, i.e., $\hat{F}_s = 2$ for $s = 1, \ldots, S$.

- Threats: Total number of threats T equals 5, and elements of a targeting matrix $\mathbf{Q}$ are randomly generated in such a way that every missile may target up to 5 ships.

- Weapons: Each asset has only hard-kill interceptors with random effectiveness parameters between 0.5 and 1. This makes minimum weapon accuracy δ_s equal to 0.5 for all ships $s \in S$. A total number of weapons H equals 15, and these weapons are randomly split between the ships of the fleet. Firing costs r_i are set to 1 for all interceptors.

One thousand of Monte Carlo runs reveal the comparative performance presented in Table 3.6.

Table 3.6: Comparison of algorithms solving the FDP

	Ships survived	Weighted ships	Average slack	Total weapons
Decentralized	2.15	6.56	0.281	12.6
Protocol	3.2	9.92	0.103	14.96
Centralized	4.94	14.9	7e-4	11.58

A completely distributed approach with ships solving their local survival problems without any communication with each other has the worst performance in terms of fleet-wide objectives. In this setting, on average only 2.15 out of 5 ships can protect themselves with the required probabilities. The weighted sum of secure ships is 6.56 out of perfect $15 = 1 + \cdots + 5$, and each ship has a survival probability which

is on average 28.1% below the required threshold. According to the decentralized algorithm, the fleet spends on average 12.6 weapons to organize its defense.

At the same time, the implementation of the cooperation protocol with even very limited communication among the ships significantly improves the global objectives of the fleet. More precisely, a number of ships that can be classified as secure grows by more than one ship ($3.2 - 2.15 = 1.05$), and this additional ship has one of the highest priorities in the fleet ($9.92 - 6.56 = 3.36$). An average survivability slack also improves substantially from 28.1% to 10.3%. On the other hand, a total number of weapons needed to protect the fleet goes up from 12.6 to almost maximum possible 14.96 weapons, since now ships try to assist insecure companions after protecting themselves.

Table 3.6 demonstrates that the best algorithm for solving the FDP is the idealistic centralized approach. In order to protect the fleet from 5 threats it takes on average 11.58 weapons. Moreover, almost all 5 ships of the fleet survive with a negligibly small average value of the multiplicative slack. At the same time, the centralized method has two major issues that make it impractical. First, it does not take into consideration the local availability of some data. Second, it is relatively slow from a tractability perspective. The communication protocol suggested in Section 3.4 and the lazy constraint algorithm introduced in Section 3.2 considerably alleviate both these limitations.

3.5.7 Discussion

The major contributions of this numerical section can be summarized as follows:

1. By definition, the MIO approach to the fleet defense problem allocates available weapons in the most efficient way. Therefore, this approach not only saves some of the weapons in comparison with another reasonable heuristic algorithm, but also significantly increases the survival probabilities of the ships in the campaign.

2. Despite the fact that the initial optimization problem for the FDP is highly non-linear, the lazy constraints algorithm is computationally effective. It is almost

two orders of magnitude faster than the SOCP approach that was employed before.

3. Implementation of the lazy constraint method allows us to solve the FDP in real time for instances of typical size and reasonable values of major underlying parameters. We can solve instances with several time periods and large campaign size within seconds as well. Moreover, in case of several attacks, utilization of the multiperiod formulation generates more efficient weapon assignments.

4. The suggested optimization based cooperation protocol provides a significant improvement with respect to a decentralized setting even with small communication capacities. It substantially increases the survival probabilities of the ships and a number of surviving ships.

3.6 Conclusion

In this chapter, we considered the fleet defense problem whose objective is to assign available weapons in order to protect ships from incoming threats. The exact formulation has a form of a highly nonlinear mixed-integer optimization problem. This optimization problem admits a SOCP reformulation that can be solved within minutes for instances of practical size. In order to expedite the process and be able to solve the FDP online, we developed a new approach employing lazy constraints techniques. We also designed a new extended MIO formulation for the multiperiod fleet defense problem, that is characterized by a sequence of independent attacks. In this scenario, the multiperiod optimization problem takes into account more information about the field of action and therefore improves the fleet-wide objectives of protecting the ships while minimizing the weapon costs.

Furthermore, we considered not only the centralized version of the FDP, but also its decentralized counterpart. For this setting we developed a cooperation protocol allowing the ships some limited communication and leveraging robust optimization methodologies. For this protocol, we were able to design two auxiliary optimization

problems whose optimal solutions uniquely determine information that ships should broadcast and how they should exploit additional information from companions to update their local plans.

Finally, we conducted extensive numerical experiments and empirically demonstrated that both the single and multiperiod formulations of the FDP can be solved online for instances of typical size with reasonable values of underlying parameters. We also showed effectiveness of the suggested RO-based cooperation protocol in comparison with a completely distributed setting with no communication among ships and the idealistic centralized coordination.

Chapter 4

Sequential Clinical Trials

4.1 Introduction

The annual expenditure of global pharmaceutical companies on research and development was $46.4 billion in 2010; of this, $32.5 billion was due to the high expense of clinical trials [Berndt and Cockburn, 2013]. Drug approval requires multiple phases of clinical trials that typically take years to complete [U.S. Food and Drug Administration, 2015]. We present an algorithm that can decrease the sample size needed to conduct a clinical trial by as much as 25-50% in certain settings, and thus substantially reduce both the cost of conducting clinical trials and the time it takes for novel effective therapies to reach patients.

The study of treatment allocation for controlled experiments dates back to Fisher [1935]. Randomization has been favored historically as a way to control for selection bias. However, randomization can yield another accidental bias identified by Efron [1971], in which there is an imbalance in the distributions of known or hidden covariates across randomly assigned treatment groups. There have been many attempts in the literature to address this accidental bias in both the offline and online allocation settings. For the offline problem, some prominent mechanisms are pairwise matching [Rosenbaum and Rubin, 1985, Greevy et al., 2004], rerandomization [Morgan and Rubin, 2012], and the finite selection model [Morris, 1979]. Bertsimas et al. [2015] used an alternative offline optimization-based approach.

For the online sequential allocation problem, Rosenberger and Sverdlov [2008] provide an excellent review of the available heuristics for covariate-adaptive randomization, including prestratification and biased coin designs. Many of the existing heuristics stem from variations of the biased coin design first introduced by Efron [1971], including nonrandomized minimization [Taves, 1974], randomized minimization [Pocock and Simon, 1975], and designs that attempt to minimize the variance of the treatment effect [Atkinson, 1982] or minimize loss of information [Antognini and Zagoraiou, 2011]. These biased coin designs outperform pure randomization and represent the current state of the art for online allocation. More recently, Kapelner and Krieger [2014] introduced a pooled sequential matching algorithm, which discards covariate data as soon as each subject is matched. Bhat et al. [2015] propose a dynamic programming algorithm for sequential allocation that comes with computational challenges for which they provide an approximation algorithm.

In this chapter, we develop a novel covariate-adaptive optimization mechanism for online allocation, which outperforms state-of-the-art covariate-adaptive randomization methods. We extend the offline mixed-integer optimization approach presented in Bertsimas et al. [2015] to the online setting in which patients arrive sequentially and each patient's covariate data cannot be observed until the time of her arrival. The new algorithm takes the form of a sequence of mixed-integer nonlinear optimization problems. The uncertainty about future subjects is modeled by robust optimization techniques with a quadratic uncertainty set [Ben-Tal et al., 2002, Bertsimas et al., 2011].

The new method, henceforth referred to as the covariate-adaptive robust optimization (CA-RO) algorithm, delivers the following benefits:

1. In all tested scenarios, the CA-RO method achieved statistical power at least as high as, and sometimes significantly higher than, covariate-adaptive randomization (CA-RAND) approaches. We present an example of a nonlinear covariate-response setting for which the CA-RO method achieved a desired level of statistical power at a sample size 25-50% smaller than that required with the best CA-RAND approach.

2. We present theoretical and empirical evidence that the optimization approach compares favorably with CA-RAND methods with respect to three advantages of complete randomization described by Efron [1971]: freedom from selection bias; freedom from accidental bias with respect to observed and hidden covariates; and, a reasoned basis for inference.

3. The algorithm is sufficiently general to produce assignments among multiple groups $p = 1, \ldots, m$ with multiple observed covariates per subject. The CA-RO algorithm can also be extended to the setting where it is possible to aggregate subjects into small clusters of size r prior to making group assignments.

4. The CA-RO algorithm is computationally tractable for instances of practical size. Our choice of uncertainty set allows us to extract a closed-form solution for the robust constraints. Hence, we are able to solve the optimization by enumeration with complexity $\mathcal{O}(m^r)$. In all observed instances, CA-RO provides the decision-maker with a high-quality assignment recommendation instantaneously via enumeration.

The rest of the chapter is organized as follows. In Section 4.2, we briefly revisit the optimization-based allocation algorithm for the offline setting from Bertsimas et al. [2015] This offline algorithm will form the basis for the online CA-RO approach we develop in Section 4.3. At the end of Section 4.3, we present computational results from experiments demonstrating the effectiveness of CA-RO in reducing between-group covariate imbalance. In Section 4.4, we provide empirical evidence that the CA-RO algorithm achieves a high level of statistical power with much smaller sample size as compared to CA-RAND methods when the covariate-response relationship is non-linear. In Section 4.5, we discuss the experimental bias and inference properties of the CA-RO approach and demonstrate that it compares favorably with CA-RAND methods. Section 4.6 contains concluding remarks.

4.2 Offline Optimization Approach

In this section, we describe a MIO approach to assign groups for the setting when pre-treatment covariate values of all subjects are known ahead of time [Bertsimas et al., 2015]. The decision-maker knows *a priori* the total number of subjects N in the experiment and the respective covariates $\mathbf{w} = (w_1, \ldots, w_N)$ of all subjects. Thus, she can make treatment allocations using this full information. This may be the case, for example, in laboratory cancer drug testing on mice.

The decision-maker will assign $k := N/m$ subjects to each of $m \geq 2$ treatment groups. The objective of the assignment is to minimize the maximum discrepancy between any two groups in the weighted sum of the first and second moments of the covariates. Without loss of generality, we assume that the vector of covariates $\mathbf{w}$ is normalized and has zero sample mean and unit sample variance. The parameter ρ regulates the relative weight of the first and the second moments. The binary decision variables are $\mathbf{x} = \{x_{ip} \,|\, i = 1, \ldots, N,\, p = 1, \ldots, m\}$, where $x_{ip} = 1$ if subject i is assigned to group p, and $x_{ip} = 0$, otherwise. We can express the mean and second moment of each of the groups $p \in \{1, \ldots, m\}$ as follows:

$$\mu_p(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{N} w_i \, x_{ip} \quad \text{and} \quad \sigma_p^2(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{N} w_i^2 \, x_{ip}.$$

Hence, the optimal offline assignment can be found using the following MIO problem, which we henceforth refer to as the OPT algorithm:

$$\begin{aligned}
\min_{\mathbf{x}} \max_{p<q} \quad & \Big(|\mu_p(\mathbf{x}) - \mu_q(\mathbf{x})| + \rho |\sigma_p^2(\mathbf{x}) - \sigma_q^2(\mathbf{x})| \Big) = \\
\min_{\mathbf{x},d} \quad & d \\
\text{s.t.} \quad & \forall p < q = 1, \ldots, m : \\
& \quad d \geq \mu_p(\mathbf{x}) - \mu_q(\mathbf{x}) + \rho\sigma_p^2(\mathbf{x}) - \rho\sigma_q^2(\mathbf{x}) \\
& \quad d \geq \mu_p(\mathbf{x}) - \mu_q(\mathbf{x}) + \rho\sigma_q^2(\mathbf{x}) - \rho\sigma_p^2(\mathbf{x}) \\
& \quad d \geq \mu_q\mathbf{x}) - \mu_p(\mathbf{x}) + \rho\sigma_p^2(\mathbf{x}) - \rho\sigma_q^2(\mathbf{x})
\end{aligned} \tag{4.1}$$

$$
\begin{aligned}
& d \geq \mu_q(\mathbf{x}) - \mu_p(\mathbf{x}) + \rho\sigma_q^2(\mathbf{x}) - \rho\sigma_p^2(\mathbf{x}) \\
& x_{ip} \in \{0, 1\} \\
& \sum_{i=1}^{N} x_{ip} = k, \quad \forall p = 1, \ldots, m \\
& \sum_{p=1}^{m} x_{ip} = 1, \quad \forall i = 1, \ldots, N \\
& x_{ip} = 0 \quad \forall i < p.
\end{aligned}
$$

The final constraint reduces the redundancy due to permutation symmetry in group numbering.

In all tested scenarios from Bertsimas et al. [2015], the OPT method generates groups with covariate discrepancy that is exponentially lower in the group size k than those created by randomization. The expected average covariate discrepancy decreases from $O(k^{-1/2})$ for randomization to $O(2^{-k})$ for the OPT algorithm. Furthermore, the OPT algorithm demonstrates exceptional precision in estimating small treatment effects and superior statistical power given a fixed treatment effect.

For the remainder of this chapter, the OPT algorithm will serve as a prescient benchmark for the performance of methods in the setting of sequential online allocation.

4.3 Covariate-Adaptive Optimization Algorithms

In this section, we introduce the proposed CA-RO algorithm, develop an extension in which aggregation of decisions is allowed, and describe the results of empirical experiments comparing the covariate balance of CA-RO versus CA-RAND methods.

4.3.1 CA-RO Algorithm

To extend the model from Section 4.2 to the online setting, we consider the problem of N subjects arriving sequentially. The decision-maker knows *a priori* the number of subjects k that will be assigned to each of $m \geq 2$ treatment groups, such that

$N = km$.

At each time-step $t = 1, \ldots, N$, where t indexes both the period and the subject, the decision-maker observes the covariate vector $\mathbf{w}_t \in \mathbb{R}^S$, where S is the number of covariates observed for each subject. We assume that this sequence of random covariate vectors is exchangeable, such that any ordering of the subjects' arrival is equally likely. The decision-maker then sets a decision $\{x_{tp}\}_{p=1}^m \in \{0,1\}^m$, where $x_{tp} = 1$, if the decision-maker assigns subject t to group $p \in \{1, \ldots, m\}$, and $x_{tp} = 0$, otherwise. In the CA-RO algorithm, the choice of $\{x_{tp}\}_{p=1}^m$ is made by solving one instance of robust MIO formulation (4.3) at each time-step. The data for the optimization at time-step t include the covariate observations $\{\mathbf{w}_i\}_{i=1}^t$ and assignments $\hat{\mathbf{x}} := \{\hat{x}_{ip} \,\big|\, i = 1, \ldots, t-1,\, p = 1, \ldots, m\} \in \{0,1\}^{(t-1)\times m}$ made at all previous time-steps. We define expressions for the sample mean $\bar{\mathbf{w}}_t$ and the empirical covariance matrix Σ_t at time-step t as follows:

$$\bar{\mathbf{w}}_t := \frac{1}{t}\sum_{i=1}^{t} \mathbf{w}_i \quad \text{and} \quad \Sigma_t := \frac{1}{t}\sum_{i=1}^{t} (\mathbf{w}_i - \bar{\mathbf{w}}_t)(\mathbf{w}_i - \bar{\mathbf{w}}_t)^\top.$$

We also define uncertain parameters $\tilde{\mathbf{w}} := \{\tilde{\mathbf{w}}_i \in \mathbb{R}^S\}_{i=t+1}^N$, which represent the unknown covariates for future subjects.

The objective of the CA-RO algorithm is to produce m groups whose covariate distributions are as similar as possible. We measure the proximity between two groups p and q in terms of the mean μ_p^s and *approximated* variance σ_p^s of group $p = 1, \ldots, m$ with respect to covariate $s = 1, \ldots, S$. At time-step $1 \le t \le N$, these sample statistics are defined as follows:

$$\mu_p^s := \frac{1}{k}\Big\{ \sum_{i=1}^{t-1} w_i^s \hat{x}_{ip} + w_t^s x_{tp} + \sum_{i=t+1}^{N} \tilde{w}_i^s x_{ip} \Big\},$$

$$\sigma_p^s := \frac{1}{k}\Big\{ \sum_{i=1}^{t-1} (w_i^s - \bar{w}_t^s)^2 \hat{x}_{ip} + (w_t^s - \bar{w}_t^s)^2 x_{tp} + \sum_{i=t+1}^{N} (\tilde{w}_i^s - \bar{w}_t^s)^2 x_{ip} \Big\},$$

where $\mathbf{x} := \{x_{ip} \in \{0,1\} \,\big|\, i = t, \ldots, N,\, p = 1, \ldots, m\}$ are the binary assignment decision variables. We model the decision at each time-step $t = 1, \ldots, N$ by the

following optimization problem:

$$\min_{\mathbf{x}} \max_{p<q} \sum_{s=1}^{S} |\mu_p^s - \mu_q^s| + \rho|\sigma_p^s - \sigma_q^s|. \tag{4.2}$$

Given that the values of future covariates $\tilde{\mathbf{w}}$ are unknown, we employ robust optimization [Ben-Tal et al., 2002] to model formulation (4.2) under uncertainty:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{M},\mathbf{V},z} \quad & z \\
\text{s.t.} \quad & z \geq \sum_{s=1}^{S} M_{pq}^s + \rho V_{pq}^s, \qquad \forall p < q \\
& M_{pq}^s \geq \mu_p^s - \mu_q^s, \qquad \forall p < q,\ s = 1, \ldots, S, \quad \forall \tilde{\mathbf{w}} \in U_w \\
& M_{pq}^s \geq \mu_q^s - \mu_p^s, \qquad \forall p < q,\ s = 1, \ldots, S, \quad \forall \tilde{\mathbf{w}} \in U_w \\
& V_{pq}^s \geq \sigma_p^s - \sigma_q^s, \qquad \forall p < q,\ s = 1, \ldots, S, \quad \forall \tilde{\mathbf{w}} \in U_w \\
& V_{pq}^s \geq \sigma_q^s - \sigma_p^s, \qquad \forall p < q,\ s = 1, \ldots, S, \quad \forall \tilde{\mathbf{w}} \in U_w \\
& \sum_{i=1}^{t-1} \hat{x}_{ip} + x_{tp} + \sum_{i=t+1}^{N} x_{ip} = k, \quad \forall p = 1, \ldots, m \\
& \sum_{p=1}^{m} x_{ip} = 1, \quad \forall i = t, \ldots, N
\end{aligned} \tag{4.3}$$

In this formulation, we use the uncertainty set U_w defined as follows:

$$U_w = \Big\{ \tilde{\mathbf{w}} \in \mathbb{R}^{(N-t)\times S} \quad \Big| \quad \tilde{\mathbf{w}}_i = \bar{\mathbf{w}}_t + (\Sigma_t)^{\frac{1}{2}} \boldsymbol{\varepsilon}_i,\ i = t+1, \ldots, N, \boldsymbol{\varepsilon} \in U_\varepsilon \Big\},$$

where perturbation vector $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_{t+1}, \ldots, \boldsymbol{\varepsilon}_N)$ belongs to the ellipsoidal uncertainty set U_ε:

$$U_\varepsilon = \Big\{ \boldsymbol{\varepsilon} \in \mathbb{R}^{(N-t)\times S} \quad \Big| \quad ||\boldsymbol{\varepsilon}||_2 = \sqrt{\sum_{i=t+1}^{N} \sum_{s=1}^{S} (\varepsilon_i^s)^2} \leq \Gamma \sqrt{(N-t)S} \Big\}. \tag{4.4}$$

The robustness parameter Γ controls the size of the ellipsoid and represents the level

of conservatism of the uncertainty set. In order to protect against experimental biases, we suggest that Γ should be chosen independently at random for each time-step (see Section 4.5).

Formulation (4.3) takes the form of a mixed-binary quadratic robust optimization problem with conic uncertainty set. In Section A.1 of the Appendix, we demonstrate that the following auxiliary optimization problems with respect to the uncertain variables $\tilde{\mathbf{w}}$,

$$\max_{\tilde{\mathbf{w}} \in U_w} \left(\mu_p^s - \mu_q^s\right) \quad \text{and} \quad \max_{\tilde{\mathbf{w}} \in U_w} \left(\sigma_p^s - \sigma_q^s\right),$$

admit closed-form solutions. Therefore, formulation (4.3) is equivalent to a mixed-binary optimization problem that can be solved via simple enumeration of m scenarios.

4.3.2 Aggregated CA-RO algorithm

The development of a partially online method is motivated by the opportunity presented when multiple subjects enroll in a clinical trial within a short period of time. Under these circumstances, the decision-maker may be able to make a joint decision regarding the simultaneous assignment of this sub-cohort of subjects to treatment groups.

For this analysis, we will distinguish the notion of time from the arrival of subjects. Time will be indexed by periods $t = 1, \ldots, T$. Subjects will be indexed separately by $i = 1, \ldots, N$ with covariate vectors $\mathbf{w}_i \in \mathbb{R}^S$, where $N \geq T$. Both the number of periods T and the number of subjects N are known *a priori*. We assume that at time t the decision-maker has observed the covariate values for $r_t \geq 1$ unassigned subjects who have arrived during time period t. Let us define $n_t := \sum_{j=1}^{t} r_j$ to represent the number of subjects who have arrived as of time t. We also introduce the vector $\mathbf{r}_t := \{r_j\}_{j=1}^{t}$. We can then define the following expressions to represent the sample mean and approximated variance of group p with respect to covariate s at time t:

$$\mu_p^s(\mathbf{r}_t) = \frac{1}{k}\Big\{ \sum_{i=1}^{n_{t-1}} w_i^s \hat{x}_{ip} + \sum_{i=n_{t-1}+1}^{n_t} w_i^s x_{ip} + \sum_{i=n_t+1}^{N} \tilde{w}_i^s x_{ip} \Big\}, \quad \text{and}$$

$$\sigma_p^s(\mathbf{r}_t) = \frac{1}{k}\Big\{ \sum_{i=1}^{n_{t-1}} (w_i^s - \bar{w}_t^s)^2 \hat{x}_{ip} + \sum_{i=n_{t-1}+1}^{n_t} (w_i^s - \bar{w}_t^s)^2 x_{ip} + \sum_{i=n_t+1}^{N} (\tilde{w}_i^s - \bar{w}_t^s)^2 x_{ip} \Big\}.$$

In the aggregated CA-RO algorithm, we solve formulation (4.3) at each time-step t, but we replace the expressions μ_p^s and σ_p^s with their generalized counterparts $\mu_p^s(\mathbf{r}_t)$ and $\sigma_p^s(\mathbf{r}_t)$, respectively. The optimal solutions $\{x_{ip}^* \in \{0,1\} \,|\, i = n_t - r_t + 1, \ldots, n_t,\, p = 1, \ldots, m\}$ to the corresponding MIO problem are used to make the assignments at period t for r_t subjects. The problem can be solved at time t by enumeration with complexity $\mathcal{O}(m^{r_t})$, and is therefore computationally tractable for instances of practical size.

If the aggregation level is uniform across time such that $r_t = r$ for all $t = 1, \ldots, T-1$ and $r_T = N - (T-1)r$, we define the CA-RO(r) algorithm with aggregation level r. We observe that the CA-RO(1) algorithm is equivalent to the fully online CA-RO algorithm and the CA-RO(N) algorithm is equivalent to the OPT algorithm from Section 4.2.

It is reasonable to assume that larger values of the aggregation parameter r lead to better performance of the partially online algorithm in terms of both covariate balance and statistical power. With a higher level of aggregation, the decision-maker has more information at the time of each decision. In Section 4.3.4, we provide empirical evidence for this relationship.

4.3.3 Practical Considerations

When using the CA-RO algorithm in practice, we suggest a few modifications and parameter selection guidelines.

1. At the beginning of the assignment process, group indices $p = 1, \ldots, m$ can be randomly assigned to each of the treatment conditions. In this way, the CA-RO algorithm is used to identify groups that are well-balanced with respect to observed covariates, but plays no role in determining which group should receive which treatment.

2. In the objective of formulation (4.3), the parameter ρ controls the tradeoff between imbalance in the sample mean and the approximated variance. In practice, to facilitate an intuitive choice of ρ, it is convenient to substitute the objective $\max_{p<q} \sum_{s=1}^{S} \left[M_{pq}^{s} + \rho \sqrt{V_{pq}^{s}} \right]$, which puts the expressions for first and second moments on the same scale. This substitution of a nonlinear objective is tractable because we are able to solve the optimization efficiently by enumeration. In the experiments that follow, we use this nonlinear objective with $\rho = 6$, which we found to yield strong results across many instances that were robust to perturbations of ρ.

3. At the beginning of the time horizon, we ensure that all groups have been randomly assigned at least one subject before we apply the optimization in formulation (4.3).

4. Toward the end of the time horizon, we set the robustness parameter $\Gamma = 0$ so as to make our algorithm more greedy and avoid overly conservative assignment decisions.

In all tested experiments, the CA-RO(r) algorithms for $r \in \{1, 3, 5\}$ produced assignment recommendations instantaneously, which suggests that the method can be used not just for clinical trials, but also for settings requiring real-time decisions, such as Internet applications.

4.3.4 Empirical Performance

In this subsection, we evaluate the empirical performance of the CA-RO algorithm. First, we review four state-of-the-art CA-RAND methods, which serve as benchmarks for the CA-RO algorithm. Second, we compare the performance of the CA-RO algorithm at various aggregation levels with pure randomization and these CA-RAND methods.

When evaluating the performance of CA-RO, we consider pure randomization (RAND) along with the matching on-the-fly algorithm of Kapelner and Krieger [2014]

(KK), and three biased coin designs: the minimization method of Pocock and Simon [1975] (PS), the D_A-optimal design of Atkinson [1982] (DA), and the covariate-adaptive biased coin design of Antognini and Zagoraiou [2011] (AZ). The biased coin design methodology with $m = 2$, generically defined as

$$\phi_t = \Pr\Big(x_{t1} = 1 \Big| \, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{t-1};\, w_1, \ldots, w_t\Big) = F\Big(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{t-1};\, w_1, \ldots, w_t\Big),$$

forms the basis of the PS, DA and AZ methods, with function $F(\cdot)$ defined separately for each method. For the PS method,

$$\phi_t = \begin{cases} \frac{1}{2}, & \text{if } D(t) = 0, \\ p, & \text{if } D(t) < 0, \\ 1 - p, & \text{if } D(t) > 0, \end{cases}$$

where p is the bias parameter and $D(t)$ represents the covariate imbalance between the two groups after $t - 1$ subjects have been assigned. For the DA method,

$$\phi_t = \frac{(1-\zeta)^2}{(1-\zeta)^2 + (1+\zeta)^2}, \quad \text{where} \quad \zeta = \frac{w_t \sum_{i=1}^{t-1} w_i (x_{i1} - x_{i2})}{\sum_{i=1}^{t-1} w_i^2}$$

for the case of one-dimensional covariates. For the AZ method, we have $\phi_t = G_j(D_t(\omega_j))$, where

$$G_j(\zeta) = \begin{cases} \frac{1}{2}, & 0 \le \zeta \le 1, \\ (\zeta^J + 1)^{-1}, & \zeta > 1, \end{cases} \qquad \text{and} \quad G_j(-\zeta) + G_j(\zeta) = 1, \, \forall \zeta \in \mathbb{Z},$$

for discrete levels of the covariate space indexed by $j = 0, \ldots, J$. In this description, $D_t(\omega_j)$ denotes the imbalance between the two groups within the level ω_j. In the KK method, subjects are either randomized to treatment groups or paired via a matching criterion based on the pairwise Mahalanobis distance. In the latter case, the new paired subject is assigned to the treatment opposite its pair in order to balance the

groups.

We now discuss the empirical performance of the various algorithms with respect to covariate balance. For $N \in \{20, 60, 100\}$ and $m = 2$, we simulated 3,000 unique sets of covariate values drawn *i.i.d.* from a standard normal distribution. We evaluated nine algorithms - RAND, PS, DA, AZ, KK and CA-RO(r) (with four different values of r) - to measure the average worst pairwise difference in generalized moments across groups (Table 4.1). For this and all subsequent experiments when evaluating the CA-RO algorithm at any level of aggregation, we chose the robustness parameter Γ in uncertainty set (4.4) independently and uniformly at random from the interval [0.5, 4] at each time-step. In terms of the discrepancy in the first moment, CA-RO was always among the best methods. The discrepancy in the second moment, which closely approximates the discrepancy in the variance in this setting, was always lower for CA-RO than for the best CA-RAND method. The discrepancy in higher moments, generalized moments of $\log(|w|)$, and $1/w$ for CA-RO methods was always comparable with other CA-RAND algorithms. As we expected, the advantage of optimization increases with r and the offline OPT algorithm has starkly better performance than other approaches.

We found similar results from additional experiments in which the covariates were generated from alternative distributions, including uniform and long-tailed Cauchy distributions.

4.4 Statistical Power of CA-RO Algorithm

A common pre-condition for the approval of any clinical trial is to demonstrate that the trial will have a sample size sufficient to make sound statistical inferences with high probability. These inferences include both statistical power, the ability to detect a positive treatment effect when one exists, and a low type I error rate, the ability to correctly identify an ineffective treatment. In classical statistical models, the power of a randomized controlled trial can be derived from the sample size and significance level, given an estimated treatment effect. Randomized allocation can yield an acci-

dental imbalance in covariates between treatment groups that can impact the ability to make experimental inferences. Traditionally, when estimating treatment effects, practitioners have been satisfied to control for this covariate imbalance *a posteriori* via regression methods [Lin, 2013].

We provide strong empirical evidence that such post hoc adjustments may produce suboptimal effect estimation, particularly when the relationship between covariates and response is nonlinear. By testing a variety of covariate-response models, we show that, at any fixed sample size, the statistical power of a clinical trial is at least as high when covariate-adaptive optimization is used rather than covariate-adaptive randomization. In settings where the covariate-response relationship was nonlinear, we observe that the power under the CA-RO algorithm is significantly higher than for state-of-the-art CA-RAND methods. Therefore, in certain settings, the use of covariate-adaptive optimization could allow decision-makers to achieve desired levels of statistical power with significantly smaller sample size as compared with CA-RAND mechanisms. Given the high cost of enrolling human subjects in clinical trials, the ability to achieve needed statistical power with much smaller sample size can result in significant cost savings for the healthcare industry and society at large.

4.4.1 Test for Statistical Power

In order to compare statistical power under CA-RAND and CA-RO online allocation procedures, we apply a hypothesis testing framework based on simulation [Bertsimas et al., 2015].

Let us assume there are $m = 2$ groups: a treatment group, which will be administered a given therapy, and a control group, which will be administered a placebo. There are N subjects in the trial, such that $k = N/2$ subjects will be assigned to each of the groups. At each time-step $t = 1, \ldots, N$, the decision-maker observes the values of a covariate vector $\mathbf{w}_t$ and makes a binary assignment x_t, where $x_t = 1$ indicates the treatment group (1) and $x_t = 0$ indicates the control group (0). Let v_t be the response measured after the assigned treatment was administered for subject t. We adopt the potential outcomes framework of Rosenbaum and Rubin [1983], such that

each subject has a pair of potential outcomes $(v_t^{(1)}, v_t^{(0)})$, where the superscript indicates treatment or control and only one of these two outcomes can be observed. Under this framework, we have the following relationship between the observed response and the potential outcomes: $v_t = v_t^{(1)} x_t + v_t^{(0)}(1 - x_t)$.

Given v_t for each subject $t = 1, \ldots, N$, we can estimate the average treatment effect $\hat{\delta}$. We adopt two estimators for $\hat{\delta}$ from Lin [2013]:

1. $\hat{\delta}_{\text{unadj}} := \frac{1}{k}\Big[\sum_{t=1}^{N} v_t x_t - \sum_{t=1}^{N} v_t(1 - x_t)\Big]$
2. $\hat{\delta}_{\text{adj}} := \beta_x$, where β_x is the estimated coefficient on x_t in the ordinary least squares regression $v_t = \beta_0 + \beta_x x_t + \boldsymbol{\beta}_w^\top \mathbf{w}_t$.

To test the significance of this observed effect $\hat{\delta}$, we adopt Fisher's sharp null hypothesis [Fisher, 1935], which states that every subject $t = 1, \ldots, N$ would have had the same response to treatment regardless of which treatment was assigned. In other words, under the sharp null hypothesis, we have $v_t = v_t^{(1)} = v_t^{(0)}$. Equipped with a complete set of potential outcomes for each subject, we can estimate the average treatment effect under alternative random allocations of subjects $1, \ldots, N$. If we compute the average treatment effect δ_b as our test statistic for each alternative allocation $b = 1, \ldots, B$, we can then estimate the p-value for our observed $\hat{\delta}$, using a two-sided test, as

$$p = \frac{1}{1 + B}\Big(1 + \sum_{b=1}^{B} \mathbb{I}\Big[|\delta_b| \geq |\hat{\delta}|\Big]\Big).$$

We reject the null hypothesis if $p \leq \alpha$ for some pre-specified significance level α; otherwise, we accept the null hypothesis.

In order to estimate the statistical power under a given algorithm $\mathbb{A}$, we generate Q random samples of N subjects with covariates drawn *i.i.d.* from a fixed distribution. We apply the hypothesis test described above for all random samples, and measure the number of samples Q_{reject} for which the null hypothesis is rejected. We evaluate the probability that the null hypothesis will be rejected by computing the ratio $\lambda := Q_{\text{reject}}/Q$. If the true treatment effect δ_0 is nonzero, then λ estimates the power of the experiment; otherwise, λ estimates the type I error rate.

The alternative allocations for the hypothesis test can be generated randomly using Monte Carlo simulation to approximate the distribution of possible allocations under random assignment mechanism $\mathbb{A}$. If Monte Carlo simulation does not yield a sufficiently diverse set of allocations within computational limits, one can generate bootstrapped resamples of covariate vectors $\mathbf{w}_t^b$, $t = 1, \ldots, N$ drawn uniformly at random from the set $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ [Efron and Tibshirani, 1994]. Based on the observations from the original experiment and under the null hypothesis, we have complete mappings $v^{(1)}(\cdot) : \mathcal{W} \to \mathbb{R}$ and $v^{(0)}(\cdot) : \mathcal{W} \to \mathbb{R}$, which represent the potential outcomes under treatment and control, respectively, for individuals with covariates in $\mathcal{W}$. Therefore, for each subject in a given bootstrapped sample, we observe the response under her random allocation x_t^b as $v_t^b = v^{(1)}(\mathbf{w}_t^b) \cdot x_t^b + v^{(0)}(\mathbf{w}_t^b) \cdot (1 - x_t^b)$.

4.4.2 Computational Results

To evaluate the statistical power of the CA-RO algorithm relative to CA-RAND methods, we simulated clinical trials under three different hidden realities, each characterized by a unique model relating treatment response to subject covariates. We assumed each subject $t = 1, \ldots, N$ had a covariate vector $\mathbf{w}_t = (w_t^1, w_t^2)$ of dimension $S = 2$, whose components were drawn *i.i.d.* from a standard normal distribution. The covariate-response models were as follows:

- Nonlinear (NL): $v_t = \delta_0 x_t + (w_t^1)^2 - (w_t^2)^2 + \epsilon_t$,
- Linear (LIN): $v_t = \delta_0 x_t + 2(w_t^1) + 2(w_t^2) + \epsilon_t$,
- No relationship (NR): $v_t = \delta_0 x_t + \epsilon_t$,

where δ_0 is the ground-truth additive treatment effect and ϵ_t is a Gaussian noise term with mean 0 and standard deviation 0.75.

For each covariate-response model, we evaluated statistical power λ under the CA-RAND and CA-RO algorithms by applying the hypothesis test described in Section 4.4.1 with both estimators $\hat{\delta}_{\text{unadj}}$ and $\hat{\delta}_{\text{adj}}$ (Figure 4-1). We considered $N \in$

$\{40, 80, 120\}$ with $\delta_0 = 0.5$, $Q = 800$, $B = 500$, and significance level $\alpha = 0.05$. For all scenarios, the power of the experiment increases with N.

- In the NR scenario, post hoc regression adjustment does not improve power for any of the methods. All methods yield similar power since there is no benefit from covariate balance.

- Conversely, in the linear response setting (LIN), regression adjustment increases statistical power substantially for all methods. When using the $\hat{\delta}_{\text{unadj}}$ estimator, CA-RO(1) yields higher power relative to randomization and CA-RAND methods. However, post hoc regression adjustment using ordinary least squares, which exactly replicates the covariate-response model with additive treatment effect, reduces the need for the *a priori* covariate balance provided by CA-RO. Power evaluated using $\hat{\delta}_{\text{adj}}$ is equally high across all methods.

- Finally, in the nonlinear response scenario described above (NL), there is virtually no benefit to using regression adjustment. In this setting, CA-RO(1) yields much higher statistical power than pure randomization and CA-RAND methods. The advantage of CA-RO grows with the sample size N.

We conducted additional experiments under a variety of nonlinear models and found, in all tested scenarios, that CA-RO had power at least as high as (and often higher than) randomization-based methods. The NL scenario is an example in which the benefit of CA-RO was particularly dramatic.

We also ran simulations in which the p-values were estimated using a one-sided test rather than a two-sided test. As one might expect, for fixed δ_0 and distribution of noise ϵ, power was higher for all methods under the one-sided test. However, in all tested scenarios, the CA-RO algorithm maintained a similar advantage relative to other methods.

In Table 4.2, we show the results for the NL setting under the CA-RO(r) assignment mechanism with aggregation levels $r \in \{1, 3, 5, N\}$ for $N = 40$. As we expect, the power increases with the aggregation level r.

In Figure 4-1, we show that, under some covariate-response models, CA-RO yields higher power at fixed sample sizes than other methods. This motivates a complementary question: What is the sample size required to achieve a desired level of statistical power? We considered the NL scenario and tested values of δ_0 from 0.75 to 1.75 to estimate $N^*_{\mathbb{A}}(\delta_0)$, the minimum number of subjects per group needed to achieve power of at least 80% when assignment mechanism $\mathbb{A}$ is employed (Figure 4-2a). With a large effect size of $\delta_0 = 1.75$, statistical power of 80% was achieved with a sample size of 22 using the CA-RO(1) algorithm compared with a sample size of 30 using the best CA-RAND method (in this case, PS). With a small effect size of $\delta_0 = 0.75$, the advantage of optimization was even bigger; a sample size of 58 was sufficient to achieve 80% power, compared with a sample size of 122 using the best CA-RAND method (again, PS). For a given treatment effect δ_0, the threshold sample size needed to achieve 80% power under the CA-RO algorithm was reduced by at least 25% relative to the best CA-RAND method (Figure 4-2b). If we consider the NL setting with $\delta_0 = 0.75$ as an example, the CA-RO method may enable the execution and analysis of some clinical trials that would otherwise be infeasible given the prohibitively large sample size required to achieve a sufficient level of statistical power when CA-RAND methods are employed.

Table 4.3 demonstrates that the minimum sample size $N^*_{\mathbb{A}}(\delta_0)$ decreases further as the aggregation level r of CA-RO(r) algorithm grows. Relative to state-of-the-art CA-RAND methods, the CA-RO approach can dramatically reduce the number of subjects enrolled in a trial without sacrificing statistical power.

We also evaluate the rate of type I errors for CA-RO(1) with $\delta_0 = 0$ with $Q = 800$ and $B = 500$ for $N \in \{40, 80, 120\}$ and for each of the three covariate-response scenarios described above, using the regression-adjusted treatment effect estimator (Table 4.4). We observe that, for each setting, the type I error is a decreasing function of N. Type I error rates for all algorithms tested are shown in Section A.2 of the Appendix. PS, the CA-RAND method with the best statistical power in this experiment, had a mean type I error rate that was uniformly higher than that produced by CA-RO(1).

4.5 Unbiasedness of CA-RO Approach

In this section, we provide empirical and theoretical evidence that the CA-RO algorithm introduced in Section 4.3 exhibits the same statistical advantages ascribed to complete randomization by Efron [1971]: freedom from selection bias, freedom from accidental bias with respect to observed and hidden covariates, and a reasoned basis for inference.

4.5.1 Freedom from Selection Bias

The CA-RO algorithm protects against selection bias, the possibility that an investigator could consciously or unconsciously influence the order of subject enrollment based on deterministic knowledge of the next treatment assignment. Through computer simulation, we demonstrate that, by selecting the robustness parameter Γ independently and uniformly at random with support [0.5, 4] at each time-step, the CA-RO method yields sufficiently random treatment assignments as to protect against this type of selection bias. For N from 30 to 100, we randomly generated 30 unique sequences of covariates $\mathbf{w} \in \mathbb{R}^N$ drawn independently from $\mathcal{N}(0, 1)$. We used the CA-RO(1) algorithm to generate 3,000 random assignments of the N subjects to two groups. We observe that one cannot determine the sequence of future assignments based on knowledge of the algorithm because, on average, the total number of possible allocations is large (Figure 4-3a) and no individual assignment sequence has a likelihood higher than 6% (Figure 4-3b).

A concern that directly competes with selection bias is certifiability. When used with a fixed and predefined sequence of robustness parameters Γ_t, $t = 1, \ldots, N$, the CA-RO algorithm is a sequence of deterministic optimization problems, each of which can be reproduced. This reproducibility provides a natural method for certifying *a posteriori* that the algorithm's recommendation was followed, given knowledge of the subjects' covariates and arrival order. If certifiability is deemed to be of greater concern than selection bias in the context of a particular experimental setting, one can apply the CA-RO algorithm with fixed robustness parameters in order to achieve full

certifiability of assignments. Conversely, certifiability is not achievable using randomized methods unless the random seed used to initialize the algorithm is provided.

4.5.2 Freedom from Accidental Covariate Imbalance

We have shown in the empirical results from Section 4.3.4 that the CA-RO method produces consistently better balance in the first two moments across groups compared with simple randomization and other existing CA-RAND approaches. In this subsection, we show that, despite only considering the observed covariates $\mathbf{w} \in \mathbb{R}^N$ when making assignment decisions, the CA-RO algorithm provides the same level of protection as CA-RAND methods against irregular allocation with respect to other, potentially unseen factors.

We consider two natural cases for the dependence of hidden factors on the observed covariates $\mathbf{w}$: no correlation and continuous dependence.

1. If there is a hidden factor that is uncorrelated with observed covariates $\mathbf{w}$, the CA-RO algorithm generates an allocation which is as random with respect to the hidden covariates as that produced by randomized methods.

2. The second case, when the unseen covariate is a continuous function of the observed covariate, warrants further discussion. We see empirically that, when unseen factor f has a polynomial or logarithmic conditional expectation in scalar random variable w, the discrepancy in higher moments and generalized moments $f = \log(|w|)$ and $f = 1/w$ for CA-RO methods is always comparable with (and often lower than) the mismatch produced by CA-RAND algorithms (Table 4.1). In the remainder of this subsection, we present formal theoretical evidence that this empirical relationship extends to the general case of continuous dependence.

To examine this general case, we assume that there are two different assignment algorithms $\mathbb{A}$ and $\mathbb{B}$ (e.g. CA-RO(1) and PS), and an unseen factor f that can be modeled in the form

$$f = g(w) + \epsilon, \tag{4.5}$$

where $g(\cdot)$ is a Lipschitz function with constant L and ϵ is some noise function. When generating groups of size k by algorithm $\mathbb{A}$, let us denote the maximum discrepancy in means with respect to unseen covariate f by:

$$z_{\mathbb{A}}^{f} := \max_{p<q} \frac{1}{k} \Big| \sum_{i \in \mathrm{I}_p(\mathbb{A})} g(w_i) - \sum_{i \in \mathrm{I}_q(\mathbb{A})} g(w_i) \Big|,$$

where $I_p(\mathbb{A})$, $I_q(\mathbb{A}) \subset \{1, \ldots, N\}$ are disjoint index sets respectively describing groups p and q produced by algorithm $\mathbb{A}$. The maximum discrepancy $z_{\mathbb{B}}^{f}$ between groups generated by algorithm $\mathbb{B}$ is defined analogously.

In Proposition 4.1, we derive probabilistic upper bounds on the the values of $z_{\mathbb{A}}^{f}$ and $|z_{\mathbb{A}}^{f} - z_{\mathbb{B}}^{f}|$, where $\mathbb{A}$ is the CA-RO(1) approach and $\mathbb{B}$ is any CA-RAND method. The first upper bound on $z_{\mathbb{A}}^{f}$, given by (4.6a), demonstrates that the maximum discrepancy in means with respect to unseen covariate f is controlled by the corresponding discrepancy with respect to the observed covariate w. The second upper bound on $|z_{\mathbb{A}}^{f} - z_{\mathbb{B}}^{f}|$, given by (4.6b), indicates that the maximum discrepancy in means with respect to unseen covariate f is as well-controlled under CA-RO(1) as under any other CA-RAND algorithm.

Proposition 4.1. Let us consider the simplest case where subjects with scalar covariates $w_i, i = 1, \ldots, 2k$ are assigned to $m = 2$ groups. For any assignment algorithms $\mathbb{A}$ and $\mathbb{B}$ that produce groups of equal size k, and for any Lipschitz function $g(\cdot) \in \mathrm{Lip}(L)$, the following inequalities hold:

$$z_{\mathbb{A}}^{f} \leq L \cdot \theta^{*}(\mathbb{A}), \tag{4.6a}$$

$$|z_{\mathbb{A}}^{f} - z_{\mathbb{B}}^{f}| \leq 2L \cdot \xi^{*}(\mathbb{A}, \mathbb{B}). \tag{4.6b}$$

In (4.6a), $\theta^{*}(\mathbb{A})$ is the optimal objective value of the auxiliary pairwise matching problem:

$$\theta^{*}(\mathbb{A}) := \min_{\mathbf{y}} \qquad \frac{1}{k} \sum_{i \in \mathrm{I}_1(\mathbb{A})} \sum_{j \in \mathrm{I}_2(\mathbb{A})} |w_i - w_j| \, y_{ij}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i \in \mathrm{I}_1(\mathbb{A})} y_{ij} = 1, \quad \forall j \in \mathrm{I}_2(\mathbb{A}) \\ & \sum_{j \in \mathrm{I}_2(\mathbb{A})} y_{ij} = 1, \quad \forall i \in \mathrm{I}_1(\mathbb{A}) \\ & y_{ij} \in \{0, 1\}. \end{aligned} \tag{4.7}$$

In (4.6b), we define the value $\xi^*(\mathbb{A}, \mathbb{B}) := \min_{c=1,2} \xi_c(\mathbb{A}, \mathbb{B})$, where $\xi_c(\mathbb{A}, \mathbb{B})$ is the optimal value of the problem

$$\begin{aligned} \xi_c(\mathbb{A}, \mathbb{B}) := \min_{\mathbf{y}} \quad & \sum_{i \in S_1^c} \sum_{j \in S_2^c} |w_i - w_j| \, y_{ij} \\ \text{s.t.} \quad & \sum_{i \in S_1^c} y_{ij} = 1, \quad \forall j \in S_2^c \\ & \sum_{j \in S_2^c} y_{ij} = 1, \quad \forall i \in S_1^c \\ & y_{ij} \in \{0, 1\}, \end{aligned} \tag{4.8}$$

and auxiliary sets of indices S_α^β, for $\alpha, \beta = 1, 2$ have the form

$$\begin{aligned} S_1^1 &= \mathrm{I}_1(\mathbb{A}) \cap \mathrm{I}_2(\mathbb{B}) \quad \text{and} \quad S_2^1 = \mathrm{I}_2(\mathbb{A}) \cap \mathrm{I}_1(\mathbb{B}); \\ S_1^2 &= \mathrm{I}_1(\mathbb{A}) \cap \mathrm{I}_1(\mathbb{B}) \quad \text{and} \quad S_2^2 = \mathrm{I}_2(\mathbb{A}) \cap \mathrm{I}_2(\mathbb{B}). \end{aligned}$$

(These sets describe the differences between the groups produced by algorithms $\mathbb{A}$ and $\mathbb{B}$.)

Proof. The proof of Proposition 4.1 is presented in Section A.3 of the Appendix. □

Having obtained theoretical upper bounds (4.6), we conducted numerical experiments to measure the values of the average pairwise distances $\theta^*(\mathbb{A})$ and $\xi^*(\mathbb{A}, \mathbb{B})$ defined in (4.7) and (4.8), respectively. We fixed $\mathbb{A}$ = CA-RO(1) and chose $\mathbb{B}$ from among RAND, PS, DA, AZ and KK, where the randomization methods were modified to ensure they would produce equal-sized groups at the end of the horizon. We randomly generated populations of size N between 60 and 100, where each subject had a scalar standard normal covariate w_i. After executing both chosen algorithms $\mathbb{A}$ and $\mathbb{B}$, we identified the index sets S_α^β, for $\alpha, \beta = 1, 2$ and solved the auxiliary optimization

problems (4.7) and (4.8). After 3,000 simulations, we observed that, in more than 99% of instances, the average discrepancy $\theta^*\big(\text{CA-RO}(1)\big) \leq 0.35$; the corresponding upper bounds for $\xi^*\big(\text{CA-RO}(1), \mathbb{B}\big)$ for various choices of $\mathbb{B}$ are reported in Table 4.5. Given that, by definition, the distances θ^* and ξ^* scale linearly with respect to the covariates w_i, $i = 1, \ldots, N$, one may derive the probabilistic counterparts of upper bounds (4.6):

$$z^f_{\text{CA-RO}(1)} \leq 0.35\, L \cdot \sigma$$
$$\max_{\mathbb{B} \in \{\text{RAND, PS, DA, AZ, KK}\}} |z^f_{\text{CA-RO}(1)} - z^f_{\mathbb{B}}| \leq 0.51\, L \cdot \sigma$$

where σ is the standard deviation of attributes w. The constant 0.51 in the right-hand side of the second bound is derived from the maximum discrepancy among CA-RAND methods in Table 4.5.

The result of Proposition 4.1 can be extended to the case of general continuous functions $g(\cdot)$ under the assumption that the support $K \subset \mathbb{R}^S$ of covariates $\mathbf{w}$ is a compact set. Indeed, any continuously differentiable function $g(\cdot)$ defined on a compact set K (including any polynomial function) is in a Lipschitz class with $L = \max_{x \in K} |g'(x)|$. Since any continuous function on K can be approximated with arbitrary precision by some polynomial according to the Weierstrass theorem, the upper bounds (4.6) hold for any continuous function $g(\cdot)$ on the set K.

4.5.3 Reasoned Basis for Inference

The results from Figures 4-3a and 4-3b, which demonstrate the variety of unique allocations that can result under the CA-RO approach, indicate that CA-RO provides a sufficient degree of randomization to be used as a reasoned basis for inference. While the probability distribution of these allocations does not appear to be uniform (see Figure 4-3b), the fact that diverse allocations arise motivates us to conduct randomization-inspired tests for statistical significance such that the power of the CA-RO method can be estimated under various scenarios in Section 4.4.

4.6 Conclusions

In this chapter, we introduced a covariate-adaptive optimization algorithm for the problem of online allocation of subjects in randomized controlled trials. Our method leverages robust mixed-integer quadratic optimization to improve upon state-of-the-art covariate-adaptive randomization methods. We demonstrated many desirable properties of the new CA-RO approach, including computational tractability, smaller between-group covariate imbalance as compared with randomization-based methods, and a low potential for common experimental biases. In all tested scenarios, the CA-RO method performed competitively with CA-RAND approaches, and sometimes significantly outperformed these methods, as measured by statistical power. We presented a setting with a nonlinear covariate-response model for which the CA-RO method achieved a desired level of statistical power at a sample size 25-50% smaller than the best CA-RAND method. Thus, the proposed CA-RO algorithm has significant potential to reduce both the cost and duration of clinical trials. The CA-RO algorithm can be used to make assignments to any arbitrary number of treatment groups and for any number of observed covariates. Finally, we constructed an extension of the CA-RO method for the setting in which it is possible to aggregate decision-making. We believe that the proposed CA-RO algorithm is an efficient alternative to covariate-adaptive randomization that can significantly strengthen experimental power in clinical trials and many other disciplines exploiting controlled experiments.

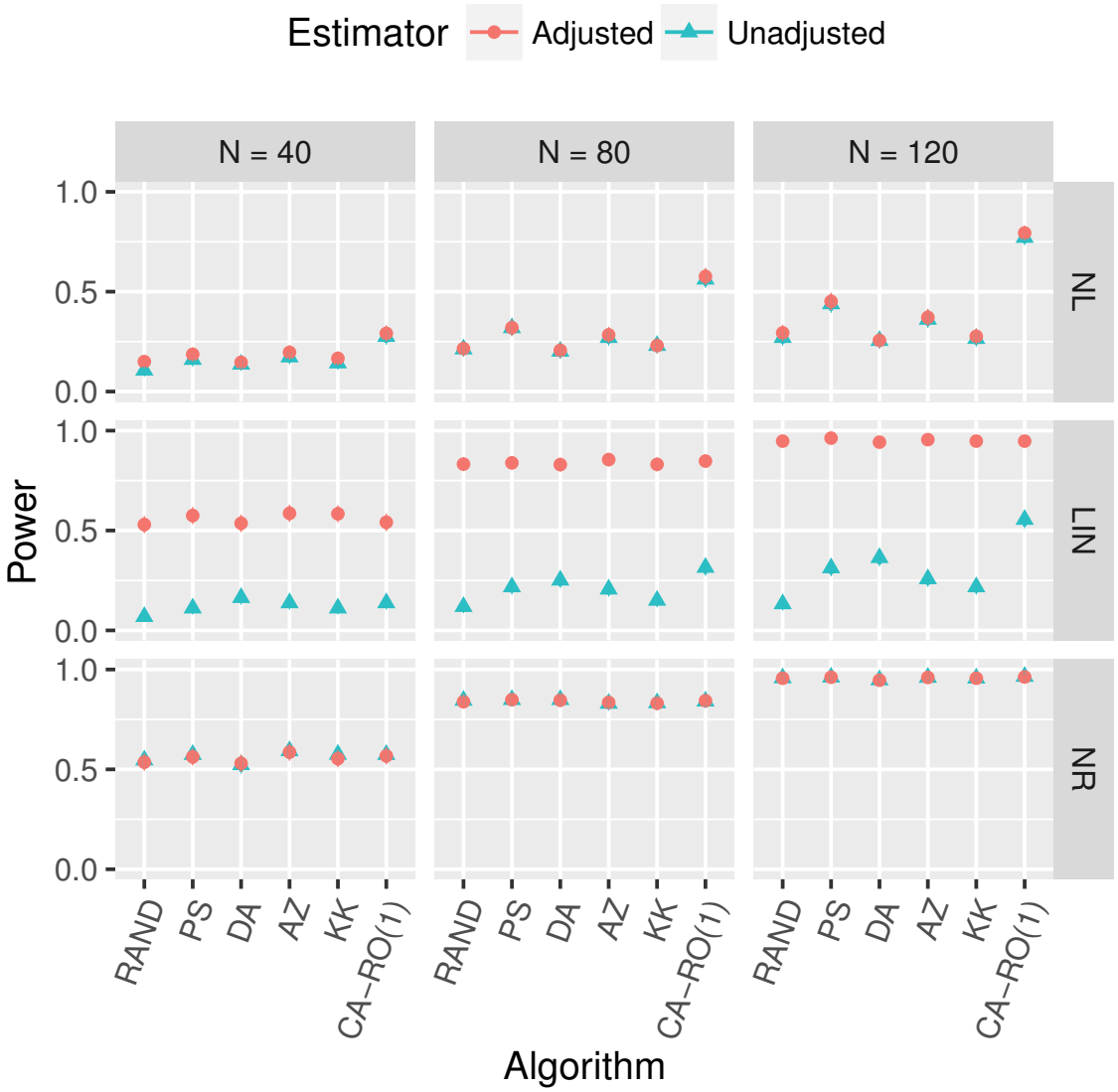
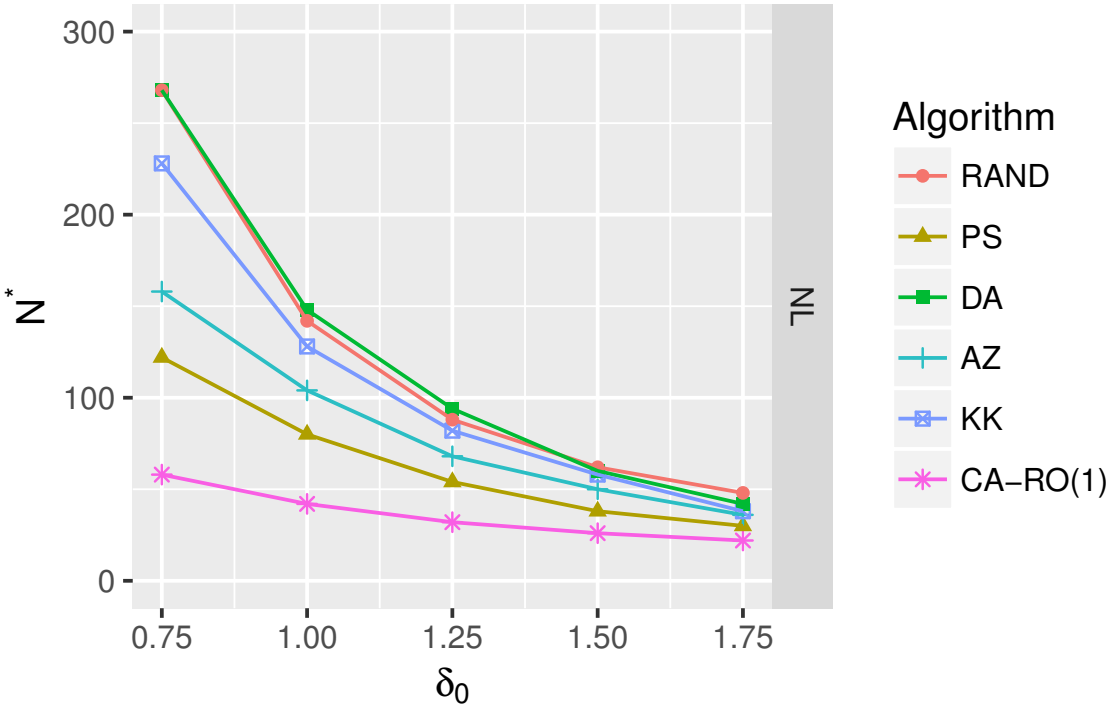
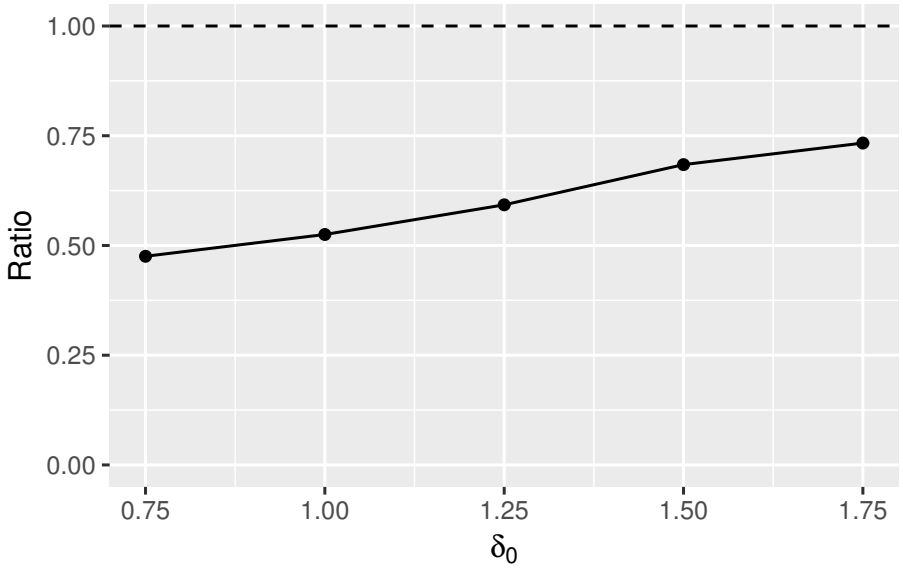


Figure 4-1: Statistical power (with 95% confidence intervals) under CA-RO(1) vs. CA-RAND methods for $N \in \{40, 80, 120\}$ under various response models, using both adjusted and unadjusted treatment effect estimators.

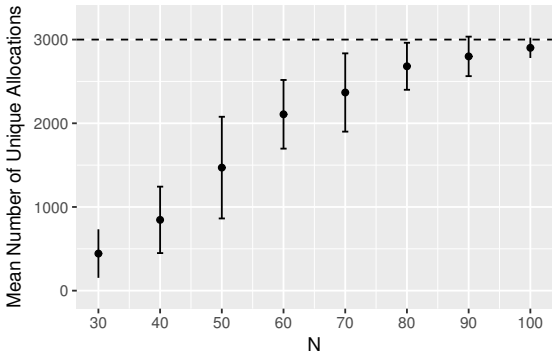


(a) Sample size needed to achieve statistical power of at least 80%.

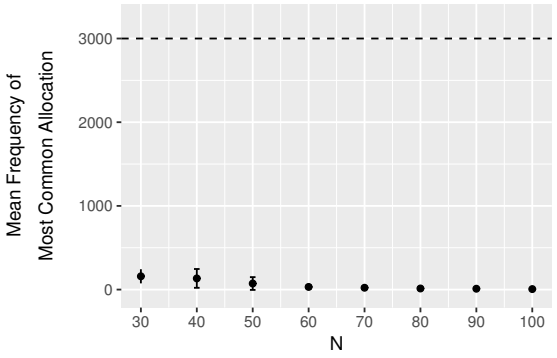


(b) Ratio of $N^*(\delta_0)$ for CA-RO(1) vs. best CA-RAND algorithm (PS).

Figure 4-2: Statistical power under CA-RO vs. CA-RAND methods.



(a) Mean (with standard deviation) number of unique allocations among 3,000 simulations.



(b) Mean (with standard deviation) frequency of
most common allocation among 3,000 simulations.

Figure 4-3: Analysis of distribution of unique allocations under CA-RO(1).

Table 4.1: Average between-group absolute discrepancy in moments under allocation algorithms with $m = 2$ and $S = 1$.

		Moment						
N	Algorithm	1	2	3	4	5	$\log(\lvert w\rvert)$	$1/w$
20	RAND	0.358	0.498	1.321	2.955	8.108	0.387	13.521
	PS	0.260	0.509	1.120	2.872	7.389	0.524	13.150
	DA	0.167	0.616	0.931	3.172	6.953	0.719	13.492
	AZ	0.286	0.553	1.228	3.070	7.881	0.558	13.143
	KK	0.221	0.416	1.046	2.706	7.186	0.305	13.307
	CA-RO(1)	0.250	0.269	1.179	2.196	7.759	0.335	13.439
	CA-RO(3)	0.251	0.226	1.228	2.033	8.022	0.340	13.639
	CA-RO(5)	0.254	0.186	1.224	1.954	7.969	0.343	13.621
	OPT	0.024	0.010	0.960	1.517	7.348	0.354	13.702
60	RAND	0.205	0.292	0.793	1.869	5.396	0.224	10.020
	PS	0.125	0.250	0.625	1.640	4.746	0.257	9.850
	DA	0.092	0.350	0.560	1.935	4.700	0.408	9.990
	AZ	0.125	0.278	0.663	1.788	4.981	0.257	9.763
	KK	0.172	0.274	0.708	1.788	5.144	0.214	9.853
	CA-RO(1)	0.099	0.139	0.629	1.378	4.992	0.214	9.979
	CA-RO(3)	0.095	0.090	0.645	1.176	5.044	0.210	9.999
	CA-RO(5)	0.096	0.067	0.653	1.128	5.112	0.206	10.028
	OPT	0.001	3.33×10^{-4}	0.531	1.046	4.668	0.272	10.255
100	RAND	0.161	0.225	0.604	1.470	4.257	0.177	10.507
	PS	0.083	0.178	0.438	1.242	3.619	0.182	10.251
	DA	0.072	0.274	0.436	1.538	3.816	0.324	10.080
	AZ	0.088	0.190	0.482	1.320	3.839	0.181	9.933
	KK	0.133	0.218	0.544	1.449	4.107	0.171	10.195
	CA-RO(1)	0.066	0.116	0.479	1.132	4.034	0.168	10.223
	CA-RO(3)	0.063	0.073	0.488	0.984	4.091	0.165	10.368
	CA-RO(5)	0.064	0.051	0.485	0.901	4.041	0.163	10.551
	OPT	0.001	1.14×10^{-4}	0.402	0.806	3.692	0.218	10.240

Table 4.2: Statistical power under CA-RO(r) for NL scenario with $N = 40$.

Aggregation level, r	1	3	5	N
Power, λ	29.1%	29.8%	31.9%	36.4%

Table 4.3: Minimum number of subjects per group $N^*_{\mathbb{A}}(\delta_0)$ needed for power over 80%.

	Treatment effect, δ_0				
Algorithm, $\mathbb{A}$	0.75	1	1.25	1.5	1.75
RAND	268	142	88	62	48
PS	122	80	54	38	30
DA	268	148	94	60	42
AZ	158	104	68	50	36
KK	228	128	82	58	38
CA-RO(1)	58	42	32	26	22
CA-RO(3)	52	36	28	22	18
CA-RO(5)	48	32	26	22	18
OPT	26	22	18	16	14

Table 4.4: Type I error under CA-RO(1).

	Sample size, N		
Scenario	40	80	120
Nonlinear (NL)	7.1%	4.6%	4.5%
Linear (LIN)	7.0%	6.0%	4.5%
No relationship (NR)	6.5%	5.3%	5.6%

Table 4.5: Empirical upper bound on $\xi^*\big(\text{CA-RO(1)}, \mathbb{B}\big)$.

Algorithm $\mathbb{B}$	RAND	PS	DA	AZ	KK
$\xi^*\big(\text{CA-RO(1)}, \mathbb{B}\big)$	0.255	0.187	0.200	0.185	0.215

Chapter 5

Conclusions and Future Research Directions

In this chapter, we summarize the main contributions of the thesis as well as discuss possible directions for future research.

5.1 Summary of the Thesis

The primary topic of this thesis is a class of online optimization problems that are characterized by incrementally revealed input data. In the online setting, the algorithm must produce sequential unchangeable decisions under uncertainty based on the past and current fragments of problem data, while the future fragments remain undetermined. We proposed a framework that leverages mixed integer and robust optimization to design and analyze new efficient online algorithms for many important decision-making domains, such as scheduling, balancing, and dynamic resource allocation. We empirically demonstrated a large number of merits of RO-based algorithms: tractability, better quality of recommendations, robustness and universality.

In Chapter 2, we combined ideas behind the Work Function Algorithm and RO, and constructed a new algorithm for the K-server problem that simultaneously incorporates historical data about the system behavior and available forecasts about the future. The new approach is (a) computationally tractable, (b) outperforms other

online methods regardless of the amount of available information about the future, and (c) is robust with respect to potential errors in the assumptions about the future.

In Chapter 3, we considered centralized and distributed versions of the fleet defense problem. We demonstrated that highly nonlinear mixed integer FDP formulation (a) can be efficiently solved with lazy constraints techniques within a matter of seconds, (b) can be extended to the multiperiod setting, and (c) admits an extension with auxiliary communication variables between ships for the decentralized counterpart of the problem.

In Chapter 4, we presented RO-based optimization algorithm for sequential allocation of subjects in online clinical trials. The new method (a) improves a between-group covariate balance in comparison with existing assignment mechanisms, (b) yields statistical power at least as high as, and sometimes significantly higher than, randomization-based algorithms, and (c) has a low potential for selection, investigator and accidental bias.

5.2 Future Research Directions

We believe that Mathematical Optimization is a powerful and universal tool for modeling and solving not only deterministic formulations, but also online multiperiod problems with intrinsic uncertainty of input data. In this thesis, we considered three examples of online problems that can be solved much more efficiently, if RO-based algorithms are employed. We do not doubt, that there are many more important applications that can benefit significantly from implementation of the generic schema described in Section 1.2. For instance, the K-server problem is a special case of a more generic Metrical Task System framework [Borodin and El-Yaniv, 2005]. The problem of sequential allocation of subjects in clinical trials can be considered from the perspective of online Internet applications, such as A/B-testing. The fleet defense problem is only one particular example of multi-agent communication and cooperation decision-making domain.

Finally, each of the three problems presented in the thesis has a large number of

remaining important open questions and possible practical extensions. We conclude with the brief outline of some promising future research directions for each of the applications.

Open questions. The K-server problem.

1. What is the competitive ratio of HARO algorithm? How does it compare with the competitive ratio of the WFA?

2. How to modify the MIO formulation of the K-server problem for the case of nonzero travel and service times?

3. Is there a tractable way to model the K-server problem for an arbitrary distance metric, rather than metrics induced by the first and the infinity norm? How to efficiently model the scenario with predefined distances between locations?

Open questions. The fleet defense problem.

1. Given that we found out that nonlinear inequalities of type

$$x_1 x_2 \dots x_n \geq \gamma$$

   can be efficiently modeled by the lazy constraints techniques, are there any extensions (for instance, for formulations with *posynomials*) for which the same technique can be applied?

2. How to design an optimization-based cooperation protocol for the scenario when the ships possess both hard-kill and soft-kill weapons?

3. Does the MIO formulation of the FDP have a robust counterpart with uncertain targeting probabilities $\mathbf{Q}$ and weapon effectiveness $\mathbf{P}$?

Open questions. Sequential allocation in clinical trials.

1. Is it possible to derive a more specific description of a class of response functions $v = v(\mathbf{w})$, for which CA-RO algorithm outperforms other methods in terms of statistical power?

2. Is it possible to find theoretical or probabilistic upper bounds on $\theta^*(\mathbb{A})$ and $\xi^*(\mathbb{A}, \mathbb{B})$ presented in Section 4.5 if the probability distribution of covariates w is given?

3. Can the MIO formulation for the CA-RO algorithm be adjusted to different variants of the online bin packing problem?

Appendix A

Supplementary Material to Sequential Clinical Trials

A.1 Tractability of the CA-RO Algorithm

Lemma 1. Consider robust optimization problem (4.3) with ellipsoidal uncertainty set U_ε as defined in (4.4). To find the optimal objective value of this discrete optimization problem and the optimal current assignment at time t, it is sufficient to inspect the following easily specified set $\mathcal{X}$ consisting of not more than m points:

$$\begin{aligned}\mathcal{X} := \bigcup_{p=1}^{m} \{\, & x_{tp} = 1; \\ & x_{tq} = 0, \quad \forall q = 1, \ldots, m,\ q \neq p; \\ & x_{iu} = 0, \quad \forall i = t+1, \ldots, N,\ u = 1, \ldots, m; \\ & \sum_{i=1}^{t-1} \hat{x}_{ip} + x_{tp} \leq k \,\}.\end{aligned}$$

Proof. In order to model the constraints for each p, q, s from optimization problem (4.3) that should hold for all possible realizations of uncertain vector $\tilde{\mathbf{w}} \in U_w$, we will

find a closed-form solution to the following auxiliary optimization problems:

$$\max_{\tilde{\mathbf{w}}\in U_w}(\mu_p^s - \mu_q^s) \quad \text{and} \quad \max_{\tilde{\mathbf{w}}\in U_w}(\sigma_p^s - \sigma_q^s).$$

Step 1. Optimization of the linear term.

Let us define a parameter $\tilde{\Gamma} := \Gamma^2(N-t)S$, where Γ is the robustness parameter from (4.4). Then, for any fixed values of p, q, s, and $\tilde{\Gamma}$, we consider the optimization problem

$$\max_{\tilde{\mathbf{w}}\in U_w}(\mu_p^s - \mu_q^s). \tag{A.1}$$

We have

$$k(\mu_p^s - \mu_q^s) = \sum_{i=1}^{t-1} w_i^s(\hat{x}_{ip} - \hat{x}_{iq}) + w_t^s(x_{tp} - x_{tq}) + \sum_{i=t+1}^{N} \tilde{w}_i^s(x_{ip} - x_{iq}),$$

where only the last term of the right-hand side depends on uncertain $\tilde{\mathbf{w}}$. Therefore, we need to solve the following optimization problem for fixed values of components of $\mathbf{x}$:

$$\begin{aligned}\max_{\tilde{\mathbf{w}}\in U_w} \sum_{i=t+1}^{N} \tilde{w}_i^s(x_{ip} - x_{iq}) &= \max_{\boldsymbol{\varepsilon}\in U_\varepsilon} \sum_{i=t+1}^{N} (\bar{w}_t^s + \mathbf{v}_{(s)}^\top \boldsymbol{\varepsilon}_i)(x_{ip} - x_{iq}) \\ &= \bar{w}_t^s \sum_{i=t+1}^{N} (x_{ip} - x_{iq}) + \max_{\boldsymbol{\varepsilon}\in U_\varepsilon} \sum_{i=t+1}^{N} (\mathbf{v}_{(s)}^\top \boldsymbol{\varepsilon}_i)(x_{ip} - x_{iq}).\end{aligned}$$

where $\mathbf{v}_{(s)}$ denotes the s-th row of the matrix $(\Sigma_t)^{\frac{1}{2}}$. The optimization problem

$$\max_{\boldsymbol{\varepsilon}\in U_\varepsilon} \sum_{i=t+1}^{N} (\mathbf{v}_{(s)}^\top \boldsymbol{\varepsilon}_i)(x_{ip} - x_{iq})$$

can be rewritten in the following form:

$$\begin{aligned} \max_{\boldsymbol{\varepsilon}} \quad & (\mathbf{a}^{pqs})^\top \boldsymbol{\varepsilon} \\ \text{s.t.} \quad & \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} \le \tilde{\Gamma}, \end{aligned} \tag{A.2}$$

where vector $\mathbf{a}^{pqs}$ of dimension $(N-t)\times S$ is defined by $(\mathbf{a}^{pqs})_{is'} = (x_{ip}-x_{iq})(\Sigma_t^{\frac{1}{2}})_{ss'}$ for $i = t+1,\ldots,N$, $s' = 1,\ldots,S$.

Simple application of the Karush-Kuhn-Tucker conditions yields that the optimal value of the optimization problem (A.2) is equal to

$$\sqrt{\tilde{\Gamma}}\cdot\|\mathbf{a}^{pqs}\|_2 = \sqrt{\tilde{\Gamma}}\sqrt{\sum_{s'=1}^{S}\sum_{i=t+1}^{N}(x_{ip}-x_{iq})^2((\Sigma_t^{\frac{1}{2}})_{ss'})^2} = \sqrt{\tilde{\Gamma}}\|\mathbf{v}_{(s)}\|_2\sqrt{\sum_{i=t+1}^{N}(x_{ip}-x_{iq})^2}.$$

The last factor can be simplified and expressed in terms of the current time-step decision variables as follows:

$$\sum_{i=t+1}^{N}(x_{ip}-x_{iq})^2 = \sum_{i=t+1}^{N}(x_{ip}^2-2x_{ip}x_{iq}+x_{iq}^2) = \sum_{i=t+1}^{N}(x_{ip}+x_{iq}) = 2k-\sum_{i=1}^{t-1}(\hat{x}_{ip}+\hat{x}_{iq})-(x_{tp}+x_{tq}),$$

where the second equality is due to the fact that x_{ip} and x_{iq} are binary variables with $x_{ip}x_{iq} = 0$. Thus, the analysis of optimization problem (A.1) allows us to write a closed-form counterpart of the linear terms in (4.3) that depends only on the current time-step decision variables x_{tp} for $p = 1,\ldots,m$, such that:

$$\begin{aligned} M_{pq}^s \geq \mu_p^s - \mu_q^s, \quad \forall\tilde{\mathbf{w}}\in U_w \quad &\Longleftrightarrow \\ k\,M_{pq}^s \geq \sum_{i=1}^{t-1}(w_i^s-\bar{w}_t^s)(\hat{x}_{ip}-\hat{x}_{iq}) &+ (w_t^s-\bar{w}_t^s)(x_{tp}-x_{tq}) + \\ &+\sqrt{\tilde{\Gamma}}\|\mathbf{v}_{(s)}\|_2\sqrt{2k-\sum_{i=1}^{t-1}(\hat{x}_{ip}+\hat{x}_{iq})-(x_{tp}+x_{tq})}. \end{aligned}$$

Step 2. Optimization of the variance term.

Similarly to Step 1, we fix values of p, q, s and $\tilde{\Gamma}$ and consider the optimization problem

$$\max_{\tilde{\mathbf{w}}\in U_w}(\sigma_p^s-\sigma_q^s). \tag{A.3}$$

As before, only the term representing the future time periods depends on the uncertain parameters $\boldsymbol{\varepsilon}$. Therefore, the primary goal of this step is to find a closed-form solution

to the auxiliary optimization problem

$$\max_{\boldsymbol{\varepsilon}\in U_\varepsilon} \sum_{i=t+1}^{N} (\mathbf{v}_{(s)}^\top \boldsymbol{\varepsilon}_i)^2 (x_{ip} - x_{iq}) = \max_{\|\boldsymbol{\varepsilon}\|_2^2 \le \tilde{\Gamma}} \boldsymbol{\varepsilon}^\top A \boldsymbol{\varepsilon} = \tilde{\Gamma} \cdot \lambda_{\max}(A). \tag{A.4}$$

In (A.4), $\lambda_{\max}(A)$ denotes the maximum eigenvalue of the square block matrix A:

$$A = \begin{bmatrix} (x_{t+1,p} - x_{t+1,q})B & 0 & 0 & \dots & 0 \\ 0 & (x_{t+2,p} - x_{t+2,q})B & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & (x_{Np} - x_{Nq})B \end{bmatrix},$$

where matrix $B = \mathbf{v}_{(s)}(\mathbf{v}_{(s)})^\top$.

The maximum eigenvalue $\lambda_{\max}(A)$ depends not only on the values of $\mathbf{x}$, but also on the dimension S of the covariate space.

- **Case 1.** $S \ge 2$. In this case, the eigenvalues of matrix B are 0 and $\|\mathbf{v}_{(s)}\|_2^2$, and the maximum eigenvalue of matrix A can be determined as a function of $\mathbf{x}$ as follows:

$$\lambda_{\max}(A) = \begin{cases} 0, & \text{if } x_{ip} - x_{iq} \le 0 \text{ for all } i = t+1, \dots, N, \\ \|\mathbf{v}_{(s)}\|_2^2, & \text{if } x_{ip} - x_{iq} = 1 \text{ for at least one } i = t+1, \dots, N. \end{cases}$$

  By construction, the condition $x_{ip} - x_{iq} = 1$ for at least one $i = t+1, \dots, N$ holds if and only if group p is not full after the current time-step assignment, i.e.,

$$k - \sum_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} \ge 1.$$

  Thus, optimization problem (A.4) has the following closed-form solution that depends only on the current time-step decision variables:

$$\max_{\boldsymbol{\varepsilon}\in U_\varepsilon} \sum_{i=t+1}^{N} (\mathbf{v}_{(s)}^\top \boldsymbol{\varepsilon}_i)^2 (x_{ip} - x_{iq}) = \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \mathbb{I}\Big\{k - \sum_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} \ge 1\Big\}.$$

Now we can exploit the closed-form solution for optimization problem (A.3) within (4.3), as follows:

$$\begin{aligned} V^s_{pq} \geq \sigma^s_p - \sigma^s_q, \quad \forall \tilde{\mathbf{w}} \in U_w \quad \Longleftrightarrow \\ k\, V^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)^2 (\hat{x}_{ip} - \hat{x}_{iq}) + (w^s_t - \bar{w}^s_t)^2 (x_{tp} - x_{tq}) + \\ + \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \mathbb{I}\Big\{ k - \sum_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} \geq 1 \Big\}. \end{aligned} \tag{A.5}$$

- **Case 2.** $S = 1$. In this case, matrix B is one-dimensional and its only eigenvalue is $\|\mathbf{v}_{(s)}\|_2^2$. Hence,

$$\lambda_{\max}(A) = \begin{cases} \|\mathbf{v}_{(s)}\|_2^2, & \text{if } x_{ip} = 1 \text{ for at least one } i = t+1, \ldots, N, \\ -\|\mathbf{v}_{(s)}\|_2^2, & \text{if } x_{ip} = 0 \text{ and } x_{iq} = 1 \text{ for all } i = t+1, \ldots, N, \\ 0, & \text{if } x_{ip} = 0 \text{ for all } i = t+1, \ldots, N \text{ and} \\ & \quad x_{iq} = 0 \text{ for at least one } i = t+1, \ldots, N. \end{cases}$$

This is equivalent to the formulation: $\lambda_{\max}(A) = \|\mathbf{v}_{(s)}\|_2^2 \cdot \Theta_{pq}(\hat{\mathbf{x}}, \mathbf{x})$, where

$$\Theta_{pq}(\hat{\mathbf{x}}, \mathbf{x}) = \begin{cases} 1, & \text{if } k - \sum\limits_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} \geq 1, \\ -1, & \text{if } k - \sum\limits_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} = 0 \text{ and } \sum\limits_{i=1}^{t-1} \hat{x}_{iq} + x_{tq} + (N-t) = k, \\ 0, & \text{if } k - \sum\limits_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} = 0 \text{ and } \sum\limits_{i=1}^{t-1} \hat{x}_{iq} + x_{tq} + (N-t) > k. \end{cases} \tag{A.6}$$

Thus, optimization problem (4.3) modeling the CA-RO algorithm with ellipsoidal uncertainty set has the following closed form for $S \geq 2$:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{M},\mathbf{V},z} \quad & z \\ \text{s.t.} \quad & z \geq \sum_{s=1}^{S} M^s_{pq} + \rho V^s_{pq}, \qquad \forall p < q \end{aligned}$$

$$
\begin{aligned}
&\forall p < q, \quad s = 1, \ldots, S: \\
&k\, M^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)(\hat{x}_{ip} - \hat{x}_{iq}) + (w^s_t - \bar{w}^s_t)(x_{tp} - x_{tq}) + \\
&\qquad\qquad + \sqrt{\tilde{\Gamma}} \|\mathbf{v}_{(s)}\|_2 \sqrt{2k - \sum_{i=1}^{t-1} (\hat{x}_{ip} + \hat{x}_{iq}) - (x_{tp} + x_{tq})} \\
&k\, M^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)(\hat{x}_{iq} - \hat{x}_{ip}) + (w^s_t - \bar{w}^s_t)(x_{tq} - x_{tp}) + \\
&\qquad\qquad + \sqrt{\tilde{\Gamma}} \|\mathbf{v}_{(s)}\|_2 \sqrt{2k - \sum_{i=1}^{t-1} (\hat{x}_{ip} + \hat{x}_{iq}) - (x_{tp} + x_{tq})} \\
&k\, V^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)^2 (\hat{x}_{ip} - \hat{x}_{iq}) + (w^s_t - \bar{w}^s_t)^2 (x_{tp} - x_{tq}) + \\
&\qquad\qquad + \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \mathbb{I}\Big\{ k - \sum_{i=1}^{t-1} \hat{x}_{ip} - x_{tp} \geq 1 \Big\} \\
&k\, V^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)^2 (\hat{x}_{iq} - \hat{x}_{ip}) + (w^s_t - \bar{w}^s_t)^2 (x_{tq} - x_{tp}) + \\
&\qquad\qquad + \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \mathbb{I}\Big\{ k - \sum_{i=1}^{t-1} \hat{x}_{iq} - x_{tq} \geq 1 \Big\} \\
&\sum_{i=1}^{t-1} \hat{x}_{ip} + x_{tp} \leq k, \quad \forall p = 1, \ldots, m \\
&\sum_{p=1}^{m} x_{tp} = 1 \\
&x_{ip} \in \{0, 1\}, \quad \forall i = t, \ldots, N,\ p = 1, \ldots, m.
\end{aligned}
\tag{A.7}
$$

The second-to-last constraint guarantees that no group will be assigned more than k subjects and is therefore a sufficient replacement for the second-to-last constraint of formulation (4.3).

A similar formulation for the case $S = 1$ is given by

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{M},\mathbf{V},z} \quad & z \\
\text{s.t.} \quad & z \geq \sum_{s=1}^{S} M^s_{pq} + \rho V^s_{pq}, \qquad \forall p < q
\end{aligned}
$$

$$
\begin{aligned}
&\forall p < q, \quad s = 1, \ldots, S: \\
&k\, M^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)(\hat{x}_{ip} - \hat{x}_{iq}) + (w^s_t - \bar{w}^s_t)(x_{tp} - x_{tq}) + \\
&\qquad\qquad + \sqrt{\tilde{\Gamma}} \|\mathbf{v}_{(s)}\|_2 \sqrt{2k - \sum_{i=1}^{t-1} (\hat{x}_{ip} + \hat{x}_{iq}) - (x_{tp} + x_{tq})} \\
&k\, M^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)(\hat{x}_{iq} - \hat{x}_{ip}) + (w^s_i - \bar{w}^s_t)(x_{iq} - x_{ip}) + \\
&\qquad\qquad + \sqrt{\tilde{\Gamma}} \|\mathbf{v}_{(s)}\|_2 \sqrt{2k - \sum_{i=1}^{t-1} (\hat{x}_{ip} + \hat{x}_{iq}) - (x_{tp} + x_{tq})} \\
&k\, V^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)^2 (\hat{x}_{ip} - \hat{x}_{iq}) + (w^s_t - \bar{w}^s_t)^2 (x_{tp} - x_{tq}) + \\
&\qquad\qquad + \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \Theta_{pq}(\hat{\mathbf{x}}, \mathbf{x}) \\
&k\, V^s_{pq} \geq \sum_{i=1}^{t-1} (w^s_i - \bar{w}^s_t)^2 (\hat{x}_{iq} - \hat{x}_{ip}) + (w^s_t - \bar{w}^s_t)^2 (x_{tq} - x_{tp}) + \\
&\qquad\qquad + \tilde{\Gamma} \cdot \|\mathbf{v}_{(s)}\|_2^2 \cdot \Theta_{qp}(\hat{\mathbf{x}}, \mathbf{x}) \\
&\sum_{i=1}^{t-1} \hat{x}_{ip} + x_{tp} \leq k, \quad \forall p = 1, \ldots, m \\
&\sum_{p=1}^{m} x_{tp} = 1 \\
&x_{ip} \in \{0, 1\}, \quad \forall i = t, \ldots, N,\ p = 1, \ldots, m,
\end{aligned}
\tag{A.8}
$$

where $\Theta_{pq}(\hat{\mathbf{x}}, \mathbf{x})$ and $\Theta_{qp}(\hat{\mathbf{x}}, \mathbf{x})$ are as defined in (A.6).

Formulations (A.7) and (A.8) depend only on current time-step decisions x_{tp}, for $p = 1, \ldots, m$. Given that these variables are binary and the subject with index t must be assigned to exactly one group, it is sufficient to inspect the set $\mathcal{X}$, with cardinality at most m, to solve (4.3) for the optimal current assignment. $\square$

A.2 Type I Error Rates

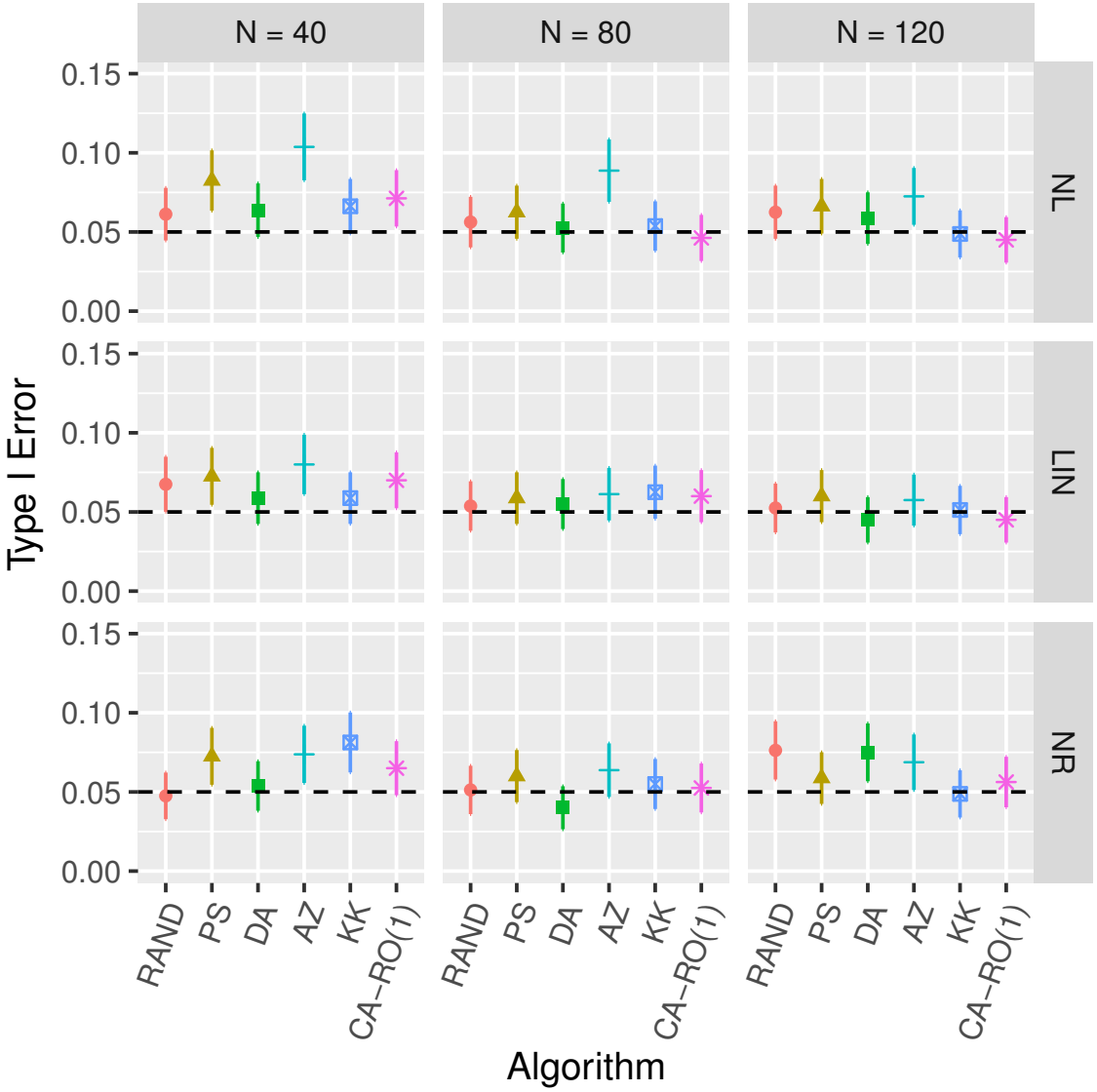


Figure A-1: Type I error (with 95% confidence intervals) with CA-RO(1) vs. CA-RAND methods for $N \in \{40, 80, 120\}$ under various response models, using the adjusted treatment effect estimator. Dashed line indicates 0.05 significance level.

A.3 Proof of Proposition 4.1

Proof. In order to verify inequality (4.6a) for $m = 2$, we note that optimization problem (4.7) uniquely determines a pairwise matching of sets $\mathrm{I}_1(\mathbb{A})$ and $\mathrm{I}_2(\mathbb{A})$ with

minimum average distance between pairs. We denote the resulting pairs as $\{(i_l, j_l) : l = 1, \ldots, k\}$, where k is the number of indices in each set. By the definition of $z^f_{\mathbb{A}}$, we derive

$$z^f_{\mathbb{A}} = \frac{1}{k}\Big| \sum_{i\in \mathrm{I}_1(\mathbb{A})} g(w_i) - \sum_{i\in \mathrm{I}_2(\mathbb{A})} g(w_i)\Big| \leq \frac{1}{k}\sum_{l=1}^{k} |g(w_{i_l}) - g(w_{j_l})| \leq \frac{L}{k}\sum_{l=1}^{k} |w_{i_l} - w_{j_l}| = L\cdot\theta^*(\mathbb{A}). \tag{A.9}$$

Similar reasoning is applicable for the second inequality (4.6b). First, it is easy to see that the cardinality of both sets S^1_1 and S^1_2 is the same:

$$\gamma := |S^1_1| = |S^1_2|.$$

By symmetry, the cardinalities of the complementary sets are also identical:

$$|S^2_1| = |S^2_2| = k - \gamma.$$

The next step is to express the between-group discrepancies in the means generated by algorithms $\mathbb{A}$ and $\mathbb{B}$ as follows:

$$z^f_{\mathbb{A}} = \frac{1}{k}\Big|\sum_{i\in S^2_1} g(w_i) + \sum_{i\in S^1_1} g(w_i) - \sum_{i\in S^1_2} g(w_i) - \sum_{i\in S^2_2} g(w_i)\Big| = |a + b|.$$

$$z^f_{\mathbb{B}} = \frac{1}{k}\Big|\sum_{i\in S^2_1} g(w_i) - \sum_{i\in S^1_1} g(w_i) + \sum_{i\in S^1_2} g(w_i) - \sum_{i\in S^2_2} g(w_i)\Big| = |a - b|,$$

where

$$a := \frac{1}{k}\Big(\sum_{i\in S^2_1} g(w_i) - \sum_{i\in S^2_2} g(w_i)\Big) \quad \text{and} \quad b := \frac{1}{k}\Big(\sum_{i\in S^1_1} g(w_i) - \sum_{i\in S^1_2} g(w_i)\Big).$$

Hence, analogously to argument (A.9), one may obtain upper bounds:

$$|a| \leq L\cdot\xi_2(\mathbb{A},\mathbb{B}) \quad \text{and} \quad |b| \leq L\cdot\xi_1(\mathbb{A},\mathbb{B}). \tag{A.10}$$

A simple corollary from the triangle inequality is that, for any a and b,

$$\big||a+b|-|a-b|\big| \leq 2\min\big(|a|,|b|\big). \tag{A.11}$$

This corollary, taken together with (A.10), implies that

$$|z^f_{\mathbb{A}} - z^f_{\mathbb{B}}| \leq 2L \cdot \min\big\{\xi_1(\mathbb{A},\mathbb{B}),\, \xi_2(\mathbb{A},\mathbb{B})\big\} \leq 2L \cdot \xi^*(\mathbb{A},\mathbb{B}).$$

This proposition has a straighforward extension to the cases of $m > 2$ groups and multidimensional covariates. The proofs have a similar structure to the case considered here, and thus are omitted. $\square$

Bibliography

R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55(6):1136–1146, 2007.

S. Albers. *Competitive online algorithms*. Computer Science Department, University of Aarhus, 1996.

A. B. Antognini and M. Zagoraiou. The covariate-adaptive biased coin design for balancing clinical trials in the presence of prognostic factors. *Biometrika*, 98(3): 519–535, 2011.

A. C. Atkinson. Optimum biased coin designs for sequential clinical trials with prognostic factors. *Biometrika*, 69(1):61–67, 1982.

C. Bandi and D. Bertsimas. Robust option pricing. *European Journal of Operational Research*, 239(3):842–853, 2014.

Y. Bartal and E. Grove. The harmonic k-server algorithm is competitive. *Journal of the ACM (JACM)*, 47(1):1–15, 2000.

M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.

A. Becker, D. Bertsimas, J. Cofield, D. Finkel, S.-H. Son, and J. Stigile. Weapon coordination in a fleet: Centralized and distributed optimization. MIT Lincoln Laboratory. Working paper, 2013.

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of operations research*, 23(4):769–805, 1998.

A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.

A. Ben-Tal and A. Nemirovski. Selected topics in robust convex optimization. *Mathematical Programming*, 112(1):125–158, 2008.

A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

R. Bent and P. Van Hentenryck. Dynamic vehicle routing with stochastic requests. Technical report cs-03-10. *Brown University*, 2003.

E. R. Berndt and I. M. Cockburn. Price indexes for clinical trial research: a feasibility study. Technical report, National Bureau of Economic Research, 2013.

D. P. Bertsekas. The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces*, 20(4):133–149, 1990.

D. Bertsimas and D. B. Brown. Constructing uncertainty sets for robust linear optimization. *Operations research*, 57(6):1483–1495, 2009.

D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical programming*, 98(1):49–71, 2003.

D. Bertsimas and M. Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

D. Bertsimas, G. J. Lauprete, and A. Samarov. Shortfall as a risk measure: properties, optimization and applications. *Journal of Economic Dynamics and control*, 28(7): 1353–1381, 2004.

D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, 2013.

D. Bertsimas, M. Johnson, and N. Kallus. The power of optimization over randomization in designing experiments involving small samples. *Operations Research*, 63 (4):868–876, 2015.

D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Math. Program., Ser. A*, 2017.

N. Bhat, V. F. Farias, and C. C. Moallemi. Optimal A-B testing. *Working paper*, 2015.

S. Bisht. Hybrid genetic-simulated annealing algorithm for optimal weapon allocation in multilayer defence scenario. *Defence Science Journal*, 54(3):395, 2004.

Z. Bogdanowicz and N. Coleman. Optimization of sensor/weapon-target pairings based on auction algorithm. *WSEAS Transactions on Mathematics*, 6(6):730, 2007.

Z. Bogdanowicz and N. Coleman. Advanced algorithm for optimal sensor-target and weapon-target pairings in dynamic collaborative engagement. Technical report, DTIC Document, 2008a.

Z. R. Bogdanowicz and N. P. Coleman. New algorithm for optimization of effect-based weapon-target pairings. In *CSC*, volume 8, pages 3–8, 2008b.

A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge university press, 2005.

A. Borodin, N. Linial, and M. E. Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM (JACM)*, 39(4):745–763, 1992.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

W. R. Burley. Traversing layered graphs using the work function algorithm. *Journal of Algorithms*, 20(3):479–511, 1996.

H. Cai, J. Liu, Y. Chen, and H. Wang. Survey of the research on dynamic weapon-target assignment problem11this project was supported by the national defense pre-research foundation of china. *Journal of Systems Engineering and Electronics*, 17(3):559–565, 2006.

A. Charnes and W. W. Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

Y. Cheung and J. H. Chung. Semi-autonomous collaborative control for multi-weapon multi-target pairing. In *Control Automation and Systems (ICCAS), 2010 International Conference on*, pages 540–545. IEEE, 2010.

M. Chrobak, H. Karloof, T. Payne, and S. Vishwnathan. New ressults on server problems. *SIAM Journal on Discrete Mathematics*, 4(2):172–181, 1991.

G. B. Dantzig. Linear programming under uncertainty. *Management Science*, 1: 197–206, 1955.

D.-Z. Du and F. K. Hwang. Competitive group testing. *Discrete Applied Mathematics*, 45(3):221–232, 1993.

I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

B. Efron. Forcing a sequential experiment to be balanced. *Biometrika*, 58(3):403–417, 1971.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. CRC Press, 1994.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

R. A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

A. Floratos and R. Boppana. The on-line k-server problem. *TR1997-732, NYU, CS Department*, 1997.

B. L. Gorissen, İ. Yanıkoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics*, 17(2):416–429, 1969.

R. Greevy, B. Lu, J. H. Silber, and P. Rosenbaum. Optimal multivariate matching before randomization. *Biostatistics*, 5(2):263–275, 2004.

Gurobi Optimization Inc. Gurobi optimizer reference manual, 2016. URL `http://www.gurobi.com`.

H. H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. *Proc. of AAAI/IAAI*, pages 22–29, 2000.

P. Hosein and M. Athans. Preferential defense strategies. Part 1: The static case. Technical report, MIT LIDS, 1990.

IBM Knowledge Center. Differences between user cuts and lazy constraints. `http://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.1/ilog.odms.cplex.help/CPLEX/UsrMan/topics/progr_adv/usr_cut_lazy_constr/04_diffs.html`, 2017. Accessed: 2017-04-01.

P. Jaillet and M. R. Wagner. Online optimization - An introduction. In *Risk and Optimization in an Uncertain World*, pages 142–152. INFORMS, 2010.

M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey. *50 years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media, 2009.

A. Kapelner and A. Krieger. Matching on-the-fly: Sequential allocation with higher power and efficiency. *Biometrics*, 70(2):378–388, 2014.

A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 244–254. IEEE, 1986.

R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. *Proc. of ACM STOC*, pages 352–358, 1990.

E. Koutsoupias. Weak adversaries for the k-server problem. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 444–449. IEEE, 1999.

E. Koutsoupias and C. H. Papadimitriou. On the k-server conjecture. *Journal of the ACM (JACM)*, 42(5):971–983, 1995.

E. Koutsoupias and D. S. Taylor. The CNN problem and other k-server variants. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 581–592. Springer, 2000.

C. Leboucher, H. Shin, P. Siarry, R. Chelouah, S. Menec, and A. Tsourdos. A two-step optimisation method for dynamic weapon target assignment problem, recent advances on meta-heuristics and their application to real scenarios. *InTech. doi*, 10, 2013.

W. Lin. Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique. *The Annals of Applied Statistics*, 7(1):295–318, 2013.

I. Lobel, A. Ozdaglar, and D. Feijer. Distributed multi-agent optimization with state-dependent communication. *Mathematical Programming*, 129(2):255–284, 2011.

M. Lubin and I. Dunning. Computing in operations research using Julia. *INFORMS Journal on Computing*, 27(2):238–248, 2015.

A. Malhotra and R. Jain. Genetic algorithm for optimal weapon allocation in multi-layer defence scenario. *Defence science journal*, 51(3):285, 2001.

M. S. Manasse, L. A. McGeoch, and D. D. Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11(2):208–230, 1990.

A. S. Manne. A target-assignment problem. *Operations Research*, 6(3):346–351, 1958.

I. Maza, F. Caballero, J. Capitan, J. R. Martinez-de Dios, and A. Ollero. A distributed architecture for a robotic platform with aerial sensor transportation and self-deployment capabilities. *Journal of Field Robotics*, 28(3):303–328, 2011.

A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22, 2007.

R. H. Möhring, F. J. Radermacher, and G. Weiss. Stochastic scheduling problems I: General strategies. *Zeitschrift für Operations Research*, 28(7):193–260, 1984.

K. L. Morgan and D. B. Rubin. Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, 40(2):1263–1282, 2012.

C. Morris. A finite selection model for experimental design of the health insurance study. *Journal of Econometrics*, 11(1):43–61, 1979.

R. A. Murphey. Target-based weapon target assignment problems. In *Nonlinear Assignment Problems*, pages 39–53. Springer, 2000.

J. Oyola, H. Arntzen, and D. L. Woodruff. The stochastic vehicle routing problem, a literature review. 2016.

S. J. Pocock and R. Simon. Sequential treatment assignment with balancing for prognostic factors in the controlled clinical trial. *Biometrics*, 31(1):103–115, 1975.

P. Raghavan. A statistical adversary for on-line algorithms. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 7:79–83, 1992.

P. Raghavan and M. Snir. Memory versus randomization in on-line algorithms. In *International Colloquium on Automata, Languages, and Programming*, pages 687–703. Springer, 1989.

W. Ren and N. Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, pages 41–55, 1983.

P. R. Rosenbaum and D. B. Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38, 1985.

W. F. Rosenberger and O. Sverdlov. Handling covariates in the design of clinical trials. *Statistical Science*, 23(3):404–419, 2008.

M. H. Rothkopf. Scheduling with random service times. *Management Science*, 12(9): 707–713, 1966.

T. Rudec, A. Baumgartner, and R. Manger. A fast work function algorithm for solving the k-server problem. *Central European Journal of Operations Research*, 21 (1):187–205, 2013.

S. Sariel and T. Balch. A distributed multi-robot cooperation framework for real time task achievement. In *Distributed Autonomous Robotic Systems 7*, pages 187–196. Springer, 2006.

R. Sitters. The generalized work function algorithm is competitive for the generalized 2-server problem. *SIAM Journal on Computing*, 43(1):96–125, 2014.

D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

D. R. Taves. Minimization: a new method of assigning patients to treatment and control groups. *Clinical pharmacology and therapeutics*, 15(5):443, 1974.

H. Terelius, U. Topcu, and R. M. Murray. Decentralized multi-agent optimization via dual decomposition. *IFAC Proceedings Volumes*, 44(1):11245–11251, 2011.

A. Thiele. *A robust optimization approach to supply chains and revenue management*. PhD thesis, Massachusetts Institute of Technology, 2004.

U.S. Food and Drug Administration. *The Drug Development Process: Clinical Research*, 2015. http://www.fda.gov/ForPatients/Approvals/Drugs/ucm405622.htm.

A. C.-C. Yao. New algorithms for bin packing. *Journal of the ACM*, 27(2):207–227, 1980.