HIGH-DIMENSIONAL OPTIMIZATION
PROBLEMS IN DECISION-MAKING AND
DISCRETE GEOMETRY

ELAHESADAT NAGHIB

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
OPERATIONS RESEARCH AND FINANCIAL ENGINEERING
ADVISER: PROFESSOR ROBERT VANDERBEI

JUNE 2019
Abstract

This dissertation is organized in two separate parts focusing on two optimization problems; a framework for scheduling of modern telescopes, and optimization problems with Fourier-analytic structures.

In the first part, we show that traditional operational schemes cannot optimally utilize the new generation of fast astronomical instruments. Then we introduce an approximate Markovian Decision Process (MDP) to model the hybrid system of telescope-environment. Given the MDP model, we present an adaptive decision-making strategy to optimally operate a ground-based instrument. Our strategy is a framework that can be adopted and customized for a wide variety of astronomical missions. It can be automatically and efficiently trained with different sets of mission objectives and constraints. In addition to our theoretical work, we developed, based on the proposed decision-making framework, an open-source software that will be used to schedule the Large Synoptic Survey Telescope (LSST\textsuperscript{1}). We compare the performance of our scheduler with the previous LSST scheduler that is designed and engineered based on traditional methods.

In the second part, we discuss how optimization problems with Fourier-analytic structures appear in continuous relaxations of some fundamental combinatorial problems. Then we explain the problem of packing with convex bodies and Turán Extremal Problem. They can be expressed as Fourier-analytic optimization problems and appear in discrete geometry and number theory respectively. Then we introduce a framework and computational tool to bridge the gap between theoretical questions and computational intuitions. The problems that we address are notoriously difficult and have long been only a subject of theoretical approaches in pure mathematics. In

\textsuperscript{1}LSST is the primary ground-based survey telescope of the next decade which is located in Chile. It will image half of the sky every few nights starting from 2021.
this study we introduce a computational approach to provide approximations, insights and intuitions for the solution of these problems. Finally, we present a formulation of a more general set of Fourier-analytic optimization problems with applications in efficient utility allocation. We also present a proposal for the future studies that can be built upon the results of this dissertation.
Acknowledgements

First, I would like to express my gratitude towards my advisor Professor Robert Vanderbei for his wisdom, encouragement, and flexibility. Having the opportunity to learn from his approach to problem solving is an invaluable part of my graduate studies. In addition to the academic guidance that I received, I treasure what I learned from Professor Vanderbei’s unique enthusiasm and genuine curiosity about everything under the umbrella of science. This has been and will be my compass for navigating and enjoying academia.

Special thanks to Professor Andrew Connolly for his support and mentorship during and after the period I visited the University of Washington, and for taking the time to read my thesis. I am also grateful for the insights that Professor Michael Strauss and Professor Han Liu provided as members of my generals committee.

I am very grateful for Professor Amirali Ahmadi’s kind supports from the very first day that I joined the graduate program, until the very last day by taking the time to be on my committee. I tremendously benefited from his generosity with his time, insights, and availability. I would also like to thank Professor Mengdi Wang for being on my committee, sharing her comments about my research and for her encouragement during the class I took with her.

During my graduate studies, I was very fortunate to have had the opportunity to collaborate with and learn from Dr. Peter Yoachim, Dr. Michael Kelly, Dr. Jacob Carruth, and Dr. Lynne Jones. I am also grateful to Professor Salvatore Torquato, Professor Leslie Greengard, Professor Miranda Holmes-Cerfon, Professor Rachel Ward, and Professor Daphne Klotsa for sharing their valuable insights about my research, its future directions and potential applications.

My research tremendously benefited from the numerous comments that various research groups provided during my talks. I am in particular grateful to Mathematics, Information, and Computation Group at Courant Institute, Theory Group v
in Bell Labs, Cryptography Group at IBM T.J. Watson Research Center, Cornell’s School of Operations Research and Information Engineering, and the Large Synoptic Survey Telescope (LSST) all-hands meeting committee for their kind invitation, and enthusiastic audiences.

It is impossible to thank my family enough for their consistent support from such long distance. Thanks and love to my friends without whom it would be difficult to navigate life in graduate school. Koushiki Bose with her calmness and generous advices has been a dependable friend to go to. Sinem Uysal with her supporting and cheerful spirit has been a pleasant presence in my life at Princeton. I also treasure my friendships with Yair Shenfeld, Mohammad Shahrad, Georgina Hall, Cemil Dibek, Jeffery Zhang, Malte Möser, Mojgan Ghasemi, and Arthur Baczyk during the graduate school.

I would also like to express my gratitude to the department of Operations Research and Financial Engineering administrative staff, especially to Mrs. Kimberly Lupinacci, Mr. Michael Bino, and Mrs. Melissa Thomas for making my experience of graduate school seamless.

I would like to thank the Large Synoptic Survey Telescope team, and the DIRAC Institute’s faculty and researchers for providing insights and expertise. In particular, Professor Željko Ivezić whose consistent attention and insightful comments greatly assisted my work. I would also like to express my gratitude to Professor Christopher Stubbs, and Dr. Robert Lupton for their insightful comments spanning from the idea stage of the telescope scheduler to the end of the project.

I was fortunate to be financially supported by the National Science Foundation under Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory.
To my Ph.D. and Undergraduate advisors

Robert Vanderbei and Amin Nobakhti.
Contents

Abstract .................................................. iii
Acknowledgements ........................................ v
List of Tables ............................................. ix
List of Figures ............................................ x

1 Introduction ............................................ 1

2 Scheduling of Modern-day Astronomical Instruments in Fast-paced, Uncertain Environments 7

2.1 Introduction ........................................ 7

2.2 Scheduling Framework ............................... 10

2.2.1 Markovian Representation ..................... 12

2.2.2 Markovian Approximation ..................... 15

2.3 Scheduler Optimization ............................. 16

2.3.1 Reinforcement Learning ......................... 17

2.3.2 Global Optimization ............................ 19

2.4 Problem of Scheduling for the LSST .............................. 20

2.4.1 Features of the Telescope Scheduler ................. 23

2.4.2 Basis Functions of the Telescope Scheduler .......... 25

2.4.3 Survey-specific Constraints .................... 29

2.4.4 Scheduler Optimization ....................... 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5 Performance of a Modified Feature-Based Scheduler for LSST</td>
<td>34</td>
</tr>
<tr>
<td>2.5.1 Sky Coverage Uniformity</td>
<td>36</td>
</tr>
<tr>
<td>2.5.2 Pairs</td>
<td>38</td>
</tr>
<tr>
<td>2.5.3 AltAz and Air-mass Distributions</td>
<td>41</td>
</tr>
<tr>
<td>2.5.4 Signal-to-noise Ratio</td>
<td>43</td>
</tr>
<tr>
<td>2.6 Discussion</td>
<td>44</td>
</tr>
<tr>
<td>3 Fourier-analytic Optimization</td>
<td>47</td>
</tr>
<tr>
<td>3.1 Introduction and Preliminaries</td>
<td>47</td>
</tr>
<tr>
<td>3.1.1 An Overview</td>
<td>47</td>
</tr>
<tr>
<td>3.1.2 Definitions</td>
<td>49</td>
</tr>
<tr>
<td>3.2 Turan’s Extremal Problem</td>
<td>50</td>
</tr>
<tr>
<td>3.3 An Approximate Formulation of TEP</td>
<td>52</td>
</tr>
<tr>
<td>3.3.1 TEP Relaxation</td>
<td>53</td>
</tr>
<tr>
<td>3.3.2 Computational Efficiency</td>
<td>54</td>
</tr>
<tr>
<td>3.4 Computational Turán Extremal Problem</td>
<td>58</td>
</tr>
<tr>
<td>3.4.1 Discretization Resolution</td>
<td>59</td>
</tr>
<tr>
<td>3.4.2 Extra Constraints</td>
<td>61</td>
</tr>
<tr>
<td>3.4.3 Numerical Results for TEP</td>
<td>61</td>
</tr>
<tr>
<td>3.5 Packing Density Upper-Bound</td>
<td>64</td>
</tr>
<tr>
<td>3.5.1 Linear Programming Formulation of PDUB</td>
<td>66</td>
</tr>
<tr>
<td>3.5.2 Packing Configurations</td>
<td>67</td>
</tr>
<tr>
<td>3.5.3 Numerical Results For the Packing Density Upper-Bounds</td>
<td>68</td>
</tr>
<tr>
<td>3.5.4 Non-linear Fourier-analytic Formulation of some Combinatorial Problems in Geometry</td>
<td>71</td>
</tr>
<tr>
<td>3.6 Concluding Remarks</td>
<td>76</td>
</tr>
<tr>
<td>4 Summary</td>
<td>80</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Key Terms and Notations Used in the Definition of the Features and the Basis Functions ................................................. 20
2.2 Features of the Approximated Markovian Model for the Telescope-environment System .................................................. 24
2.3 Basis Functions of the Feature-based Scheduler, the Building Blocks of the Decision Function ........................................... 26
2.4 Feasibility of Field-Filter \((i, f)\) for a Visit at \(t^{n+1}\) .................................................. 30
2.5 The Median and Variance of the Co-added Depth Distribution on a Finely Discretized Sky .................................................. 40
2.6 Comparison of the Schedulers in a Section of the LSST Main Survey Area; WFD .................................................. 44
3.1 Numerical Upper-Bounds for the Turán Extremal Problem .................. 63
3.2 Turán Extremal Problem Computations ............................................... 63
3.3 Numerical results for the Packing Density Upper-Bound .................. 71
3.4 Packing Density Upper-Bound Computations ............................................... 73
List of Figures

2.1 Regions of the sky with different requirements and constraints for scheduling: (1) Galactic Plane (GP), (2) Universal or Wide Fast Deep (WFD), (3) South Celestial Pole (SCP), (4) North Ecliptic Spur (NES), and (5) Deep Drilling Fields (DD). ........................................... 21

2.2 One-dimensional slices of two objective functions, $U_1$ and $U_2$, defined in Equations (2.12) and (2.13), respectively. The variation of the objective functions, especially in the midrange slices (solid line), suggests that the LSST is controllable with the Feature-based Scheduler. . . . . . . . 28

2.3 Reinforcement learning of the scheduler’s parameters, $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. All of the parameters are initialized at the midrange value, 5. During the simulation, after each decision step there is a reward associated with the outcome of the decision that implies a small adjustment on each of the five design parameters. Then, the next decision will be taken with a slightly different set of parameters. This procedure continues until the adjustments on the variables are negligible. . . . . . . . 32

2.4 Progress of the black-box objective function $U_{\pi_0}$ over the iterations of the eDE algorithm. $U_{\pi_0}$, for this simulation is the total number of the observations for 10 nights starting from 2021 January 1. . . . . . . . 34
2.5 Sky coverage in each of the six filters \([u, g, r, i, z, y]\), measured by co-added depth. According to the mission’s objective, the scheduler has to provide a uniform coverage of the visible sky within each region and in each filter. The left panels show the opsim V4 simulation results, while the Modified Feature-based scheduler is on the right. Even without a given observation proposal, the Modified Feature-based scheduler can closely match the large-scale footprint of the official survey. (The individual yellow dots with high co-added depth are fields in the Deep Drilling survey, and the NES region is not visited in the \(u\), and \(y\) bands.)

2.6 Co-added depth coverage in a zoomed-in area of the sky map, around the border between the WFD (green) and SCP (blue) regions, in the \(r\) band. The smooth coverage of the Modified Feature-based scheduler (right) vs. the granular pattern of opsim V4 (left) further respects the uniformity of the coverage, which is one of the most fundamental objectives of a survey instrument. The middle panel is the coverage of opsim V4, with dithering of the same sequence of observations that fundamentally cannot become as smooth as the right panel, because, unlike Feature-based scheduler, the scheme of opsim V4 does not easily allow for decision-making with arbitrarily fine tessellations of the sky.

2.7 Each plot compares the distributions of the co-added depth coverage in one of the six filters. A dithering scheme in the Modified Feature-based scheduler in addition to a finer tessellation of the sky smoothens the density of the coverage where the fields overlap.
2.8  Ratio of the pairs (in the \( g, r, \) and \( i \) filters) to the total number of the observations on the sky map. For the areas to which the pair constraint is applied, the ratio is desired to be 1. For the Modified Feature-based scheduler (right), the pair constraint is applied to all of the regions, whereas for opsip V4 (left), it is applied to the WFD and NES regions only.

2.9  Distribution of the pair ratio to the total number of the observations, in the \( g, r, \) and \( i \) filters. The Feature-based scheduler covers a large area of the sky by a successful pair visit (higher median), whereas opsip V4, maintains a uniform ratio of pair visit for a larger area of the sky (lower variance).

2.10 Each plot is the distribution of the visits on an altitude-azimuth sky map in one of the six filters. The two left columns belong to opsip V4, and the two right columns belong to the Modified Feature-based scheduler. The higher concentration on the meridian (vertical axis) for the Modified Feature-based scheduler shows a more desirable behavior. Moreover, consistent concentration of the visits on the east wing can potentially provide a better success rate in pair observations.

2.11 Distribution of the visits on an altitude-azimuth sky map for three schedulers. On the left, opsip V3 schedules most observations at high air-mass and very few on the meridian. In the middle, opsip V4 schedules many observations on the meridian but still executes deep drilling fields and a smattering of other observations at high air-mass. On the right, the Modified Feature-based scheduler concentrates observations around the meridian, including deep drilling observation.
3.1 Optimal solutions of TEP for two known cases. The two top plots are the solutions for $K = [-1, 1]$, and the two bottom ones are the solutions when $K$ is $[-1, 1]^2$. Note that the solutions are symmetric about the origin, but the two-dimensional figures show one quarter of the solutions for clarity.

3.2 Computational efficiency gain for two-dimensional TEP. The left plot shows the growth of the number of constraints and variables with the increments of discretization resolution. It is clear that the intrinsic size of the optimization problem is larger in the sparse versions of the problem. The right plot compares the computational time of the dense and the sparse versions of the problem with the growth of the variables.

3.3 Numerical solution with arbitrary choices of truncation and discretization.

3.4 Convergence of the numerical solutions to the known solution of TEP in dimension one. With the proposed choices of discretization, the deviation decreases as the resolution of the discretization increases.

3.5 Two experiments for the solution of TEP relaxations with $K = [-1, 1]^2$. The two top plots are experiments with coarser discretization compare to the two bottom plots.

3.6 Convergence of the upper-bound with the growth of the discretization resolution for TEP in different dimensions and with different convex bodies.
3.7 Packing of $[-5.5, 5.5]^2$ with non-overlapping unit squares. This (admittedly trivial) packing problem is solved through the Fourier-analytic Linear Programming formulation where the model and the solver are both completely oblivious to the combinatorial origin of the problem. The optimal configuration, marked by black crosses, is given by where the optimal solution of Problem 3.17 vanishes.

3.8 Arbitrary convex bodies. Our framework can provide density upper-bound estimations without exploiting the special structure of a convex body.

3.9 Convergence of the density upper-bound estimations as the size of the "time" domain, $B$, and consequently the discretization resolution grow.

B.1 The moving circular mark is where the telescope is pointed to at any given moment. Those parts of the sky that are covered during the same night are star-like colored marks, and the color reflects the filter with which the observation is performed. The gray areas are where the sky is bright and generally no observation is allowed, for instance, the halo around the moon. The white areas show where the clouds are based on a realistic model of the cloud covers at the LSST site in Cerro Pachon. Finally, the light-orange areas, which do not change as the time goes by, indicate the four main sky regions (Galactic Plane Region, Universal or Wide Fast Deep, South Celestial Pole, North Ecliptic Spur).
Chapter 1

Introduction

Modern-day astronomy is dramatically changing with the influx of big data that is collected by a new generation of the telescopes. The fast operational features of these instruments allow for the collection of an enormous amount of data by quickly and repeatedly observing the sky. The algorithm that makes the sequential decisions of which filter to use, and which direction to point the telescope to is called the scheduler. The chosen filter determines at which wavelength the observation has to be performed.

In the first part we present a framework to design schedulers of modern astronomical instruments. Telescope schedulers are expected to respond to competing scientific priorities and stochastic variations in the weather in a timely manner. They also have to maximize the scientific outcome of the telescope during its limited period of operation.

Older instruments such as the Keck (Nelson et al., 1985), and ALMA (Wootten 2003) telescopes are scheduled based on chunks of pre-scripted sequences of astronomical targets called proposals. These proposals are manually created by astronomers according to their individual scientific needs. They are then gathered for execution
which often requires the examination of the feasibility and priority of the proposals by a human operator.

The fast operational features of modern telescopes provide the possibility of executing multiple proposals simultaneously. For instance, by quickly redirecting the telescope, to and from, a proposal’s target while waiting for the next target of another proposal to become available. Moreover, on a partially cloudy night, fast telescopes can operate by chasing the holes in the clouds, whereas in the same situation the telescopes with pre-scripted schedules fail to operate.

These advantages motivate a scheduling policy that makes online decisions at a single observation level. Large Synoptic Survey Telescope (LSST) for instance, is capable of 1000 observations each clear night over a period of 10 years, amounting to about 2.5 million inherently dependent decisions. Handcrafting or brute-forcing an optimal sequence of such a large number of the decisions is obviously out of the question.

To address the stochastic elements of the decision process, we proposed a Markovian model of the instrument-environment. One of the main challenges is to design a realistic and computationally tractable state-space which at the same time respects the theoretical assumptions of the Markovian model. This dissertation presents a model which satisfied those theoretical and applied requirements. To incorporate the state of the model into the decision-making process we proposed a class of parametrized policies. These policies take the current state of the telescope-environment system and decide the next state transition.

Another challenge is to formalize the performance of the scheduling as a concrete objective function in order to find a set of optimal parameters for the decision policy. However, there are various interpretations of the mission, and the mission itself could be adjusted from time to time. To provide a versatile optimization framework we
proposed a reinforcement learning and an evolutionary algorithm for different classes of possible objective functions.

The former optimization method is automated by design, however, the latter requires the users to tune its meta-parameters. Such an expectation from users outside of the optimization community is unrealistic. To overcome this difficulty, we adopted Entropic Differential Evolution (eDE) introduced by Naghib and Nobakhti (2016). eDE is an adaptive evolutionary algorithm which does not require the user to tune any meta-parameters. This algorithm constructs a supervisory subroutine which uses a notion of entropy to measure the diversity of candidate optimal solutions of the evolutionary part of the algorithm. Based on this measure, the subroutine adaptively tunes the meta-parameters which determine the tendency of the evolutionary algorithm to explore or exploit the search space.

***

Standard optimization technologies tend to suffer from the combinatorial nature of many fundamental problems in optimization. The focus of second part is on building a framework to address this issue by exploiting the special structure of some problems in discrete domains. In particular, we present a class of infinite-dimensional optimization problems with Fourier-analytic structures which provides an organic link to bridge the continuous nature of interior-point solvers with combinatorial optimization. This is because of an elegant property of the Fourier transform which states that dispositions in the domain of $x$ (the original search domain) are equivalent to multiplications in the domain of $\hat{f}(.)$ (FCO domain). More specifically, let $f : R^N \rightarrow C$ be an $L^1$ function, and $\mathcal{F}$ denote the Fourier operator, then $\hat{f} := \mathcal{F}(f(\cdot))$ is the Fourier transform of $f(\cdot)$, with the following property:

$$\mathcal{F}(f(\cdot + \Delta x))(\xi) = e^{-2\pi i \Delta x \xi} \hat{f}(\xi).$$

(1.1)
This property states that geometric translations in the domain of $x$ are equivalent to multiplications in the domain of $\hat{f}(\cdot)$. Building on this property, some problems that involve discrete geometrical translations can take equivalent or relaxed continuous form in the Fourier domain. The FCO formulation decouples the variables of this type of problems from their highly nonlinear objective functions, and translates them into well-behaved trigonometric functions for which there are efficient computational technologies, tailored from software to hardware levels. Moreover, having the explicit decomposition of the objective functions in Fourier basis, one can adjust accuracy in return for computational cost by omitting the negligible terms in the Fourier expansion. Hence this approach provides a versatile trade-off practices.

We present an efficient computational technology to provide concrete solutions to a large class of combinatorial optimizations with infinite-dimensional Fourier-analytic structures. The break-up of classical duality theory due to an uncountable number of variables and constraints, the requirement from numerical solutions to respect the Fourier structure, and the need for efficient sampling schemes are amongst the challenges we face. To address the curse-of-dimensionality, we designed a Fast Fourier transform algorithm for high geometrical dimensions. To ensure that our problem respects its Fourier-analytic structure we derived a specialized discretization scheme. To speed up the computation we used problem-specific initializations and added valid constraints which considerably narrowed down the search space. Combining these methods we were able to approximate the optimal solutions of some instances that, to the best of our knowledge, have never previously been computed.

The instances that we focused on are nontrivial FCO formulations for some fundamental open problems in discrete mathematics: Packing Density Upper-Bound (PDUB) with convex bodies in discrete geometry \cite{Coln2003}, and Turan’s Extremal Problem (TEP) \cite{Revesz2011} which has applications in number the-

\footnote{Compact convex sets with non-empty interiors}
ory. Latest theoretical advances on Sphere Packing includes the remarkable proof of
the densest packing in 8-dimensional Euclidean space by Viazovska (2017), and later,
building on her proof, for 24-dimensional Euclidean space by Cohn et al. (2017). Both
of these proofs rely on the FCO formulation that this chapter has revolved around.

The numerical efforts to estimate the density of packing is an active area of re-
search conducted by Frank Valentine et al. (e.g., the results by Dostert et al. (2017)),
and Torquato et al. (2009), respectively holding the best upper-bounds and lower-
bounds of packing density with some convex bodies. The focus of these computational
efforts have been mainly on exploiting the special features of the convex body of in-
terest, such as symmetry, which have resulted in significant improvements of the
approximations. Our computational tool can be applied to packing with arbitrary
convex bodies, for which no density upper-bounds are known. Moreover, our technol-
ogy is scalable to higher geometrical dimensions where the alternative methods are
likely to fail due to the curse-of-dimensionality. Finally we explain the generalization
and applications of this computational technology in some other contexts such as
continuous facility allocation.

Contributions

The following contributions are addressed in the first part.

- Establishing the necessity of a regime change in the scheduling of the modern
  astronomical instruments.

- Creating a framework to connect the astronomical expertise and operations re-
  search for decision-making in the context of optimal astronomical observations.

- Developing a holistic scheduling software package with flexibility to re-optimize
  for variety of astronomical instruments and missions. This software package
does not require expertise in optimization and operations research, and is designed to be used by the astronomy community.

- Designing a scheduler methodology for the Large Synoptic Survey Telescope, which will be used to operate this telescope starting in 2021.

And in the second part, the following contributions are discussed.

- Creating a generalizable computational model for the optimization of Fourier Constrained Optimization (FCO) problems.

- Developing a computational tool for linear FCO problems.

- Providing numerical estimations for the density of packing with convex bodies which are instances of FCO problems.

- Providing understanding for the packing with convex-bodies that are not well-understood in theoretical studies.

- Improving rigorous upper-bounds for the optimal solutions of the Turán Problem, another instance of the FCO problems.
Chapter 2

Scheduling of Modern-day Astronomical Instruments in Fast-paced, Uncertain Environments

2.1 Introduction

The algorithm that makes the sequential decisions of which filter to use and which direction to point the instrument toward is called the scheduler. A scheduler has to maximize the scientific outcome of the instrument during its limited period of operation.

The first generation of schedulers for astronomical instruments was developed for space missions mainly to automate their operation. The ROSAT mission’s scheduler (Nowakowski et al. 1999), Spike (Johnston et al. 1994), the Hubble Space Telescope’s scheduler, and Heuristic Scheduling Testbed System (Muscettola et al. 1995)
pioneered many of the developments in algorithmic scheduling of observations for the space missions.

Despite the similarity of the science objectives for space- and ground-based telescopes, the determining factors for the purpose of scheduling are fundamentally different. While space telescopes are required to respect kinematical and dynamical constraints, weather is the main challenge in the scheduling of ground-based telescopes. The former is predictable and efficiently computable, while the latter involves both inherent uncertainties and uncertainties due to computational limitations.

Earlier algorithmic approaches to the scheduling of ground-based telescopes are heavily based on observation proposals. Proposals are handcrafted sequences of scripted astronomical observations. They are generally tested only for feasibility (e.g., that a set of fields were visible, or lie within a specified air-mass range, or within a window in time), but not necessarily for optimality. For instance, the operation of Keck Telescope (Nelson et al., 1985), is 100% based on proposals, and the Hobby-Eberly Telescope (Shetrone et al., 2007), has a semimanual scheduling scheme.

More recently, the development of more expensive ground-based instruments with complex missions made it impossible to rely solely on handcrafted proposals. The need for more efficient use of the instrument’s time led to the development of decision-making algorithms to optimize their science output.

A common algorithmic practice is called queue scheduling, where many proposals are grouped together and priority is assigned to them based on the sky conditions and proposal requirements. For instance, the scheduler of the Liverpool robotic telescope was designed in 1997 to automatically allocate time slots to chunks of scripted observations. This time allocation strategy was preferred to scheduling at the single-visit level. The scheduling at the single-visit level is referred to as optimal scheduling by Steele, and Carter (1997), and it is stated that the optimal scheduling requires reeval-
uating the future sequence of observations once it is interrupted, but the necessary extra computation is neither affordable nor fast enough. However, in this study we show that the scheduling in the single-visit level, optimal scheduling, can be quickly recovered after an interruption, if a memoryless framework is used. Thus, the optimality does not necessarily need to be sacrificed because of the limited computational resources.

Another example is the Las Cumbres Observatory, Global Telescope Network (LCOGT), with one of the most advanced telescope scheduling algorithms (Boroson et al., 2014). LCOGT uses an integer linear programming (ILP) model to optimize the scheduling of observations over a global network of telescopes (Lampoudi et al., 2015). Due to the success of this approach, the Zwicky Transient Facility at Palomar Observatory (Belm, 2014) has also adopted a simillar scheduler. The ILP scheduling model performs well for observatories where slew time overheads are small compared to exposure execution time. In contrast, for faster instruments with larger cameras and short exposure execution time, scheduling algorithm must explicitly minimize the slew times between successive observations. With a fundamental adjustment however, ILP scheduling approach could be used for faster instruments as well. First one needs to acquire large scripted blocks of observations such that the slew time is minimized within each script. The blocks could be set to follow a path that only includes short slews (this is similar to the strategy taken in the scheduler developed by Rothchild et al. (2019)). The disadvantage of this approach and any other scripted schedule is principally the lack of recoverability from unpredictable interruptions, such as inability to dodge the clouds.

Even given the reliability of fully automatic scheduling technologies, there remain a number of modern telescopes, such as SALT (Brink et al., 2008), and ALMA (Wootsen, 2003), that are being operated based only on traditional handcrafted proposals.

1GitHub repository: https://github.com/ZwickyTransientFacility/ztf_sim
ALMA, in particular, requires a highly regulated structure for proposals that potentially leads to suboptimality as demonstrated by Alexander et al. (2017). They also suggest a number of corrections for the scheduling regulations to provide adaptivity to time-sensitive observations.

The Large Synoptic Survey Telescope (LSST) project, first developed a proposal-based scheduler, created by Delgado and Reuter (2016). It also supported the design and implementation of proposal-free decision algorithms, such as the scheduler proposed in this thesis, and a semi-scripted cadence by by Rothchild et al. (2019). The LSST is a ground-based survey telescope and one of the pioneers of the fast modern generation of astronomical instruments. In this thesis along with the application of our scheduling framework, we discuss its unique properties and requirements in depth.

This chapter is organized as follows. In Section 2.2, first we explain the choice of Markovian framework for the proposed scheduler. We refer to our scheduler as Feature-based scheduler. In Sections 2.2.1 and 2.2.2 we provide the mathematical details of the scheduler model in that framework. Section 2.3 presents two approaches for the optimization of the model’s parameters. Sections 2.4 demonstrates the application of the Feature-based scheduler on the Large Synoptic Survey Telescope (LSST) which is then followed by a comparison between a modified version of the Feature-based scheduler and LSST’s proposal-based scheduler in Section 2.5. Finally, Section 2.6 presents our concluding remarks.

## 2.2 Scheduling Framework

To run a ground-based telescope with multiple science objectives, such as the LSST, the scheduler has to offer controllability, adjustability, and recoverability.

- **Controllability**: A telescope is controllable by a given scheduler if changes of the scheduler’s design parameters are visibly manifested in the behavior and
performance of the telescope. Controllability is determined by two factors: the information that is fed to the scheduler and the structure of the scheduler. Feeding the scheduler with information that is not sufficiently relevant to the high-level mission objectives leads to an output schedule that is irrelevant to the performance of the telescope. In addition, a scheduler with a flexible structure is needed to cover a large class of decision strategies, each determined by a set of its design parameters. Controllability is necessary for a scheduler to be optimizable by searching within the space of its design parameters.

- **Adjustability**: A scheduler must be adjustable according to new conditions, environment, and scientific desirables. For a complex and multiobjective mission it is common that the scientific goals are required to be modified in the middle of the operational period. Another example is aging that causes changes in the mechanical characteristics of the telescope. Regardless of the reason, a scheduler is adjustable if the changes can be accommodated with reasonable computational cost, and preferably no or minimal human-expert intervention. Hand-tuned scheduling strategies and policies that are written in forms of instructions for human operators, or are based on observation proposals, are not fully adjustable.

- **Recoverability**: The presence of unpredictable factors in the operation of ground-based instrument is due to the natural stochastic processes (such as the weather) and the complexity of the mechanical facility. Unscheduled downtime and instrument failures are examples of the many unpredictable survey interruptions. In addition, there are inherently predictable interruptions, such as maintenance downtimes and cable winding, that, due to the complexity of the mechanical system, are not computationally affordable and/or valuable to keep track of. Therefore, they are considered to be stochastic variables as well. A
scheduler is required to be able to make alternative decisions once a previously unpredictable event occurred. Moreover, it has to return to its optimal behavior shortly after the interrupting event is over. Such a scheduler is called recoverable, and its response to the interruptions is ideally as quick as the length of a single observation. The response time is usually limited by computational complexity of the scheduling algorithm. For instance, strategies that need to look back at the history of observations or look forward through possible sequences of observations are not fast recoverables owing to their computational

The Feature-based scheduler is verifiably controllable. The Markovian framework provides a well-defined and tractable set of design parameters and a well-defined measure of the performance for any given choice of the design parameters; therefore, controllability is empirically verifiable. It is adjustable because the derivation of the design parameters is automatic and computationally tractable, once a new high-level mission objective is given or the telescope-environment system changes. Finally, it is recoverable owing to the inherent memorylessness of the Markovian Decision Process (MDP), which, for a decision at any time, only requires the current state of the system.

2.2.1 Markovian Representation

**Definition 1.** Let \( X(\cdot) \) be a stochastic process for which \( X_i \) represents the state of the system at \( t_i \), and let \( \mathcal{S} \) be the set of all possible states that the system can take. Let \( P(X_i) \), be the probability distribution of \( X_i \) on \( \mathcal{S} \). Then, \( X(\cdot) \) is a Markovian process if and only if it satisfies the following *memorylessness* property,

\[
\forall i \quad P(X_{i+1}|X_i) = P(X_{i+1}|X_i, X_{i-1}, \ldots, X_0),
\]
where \( P(X_{i+1}|X_i) \) is the conditional probability distribution of the system’s state at \( t_{i+1} \) given its state at \( t_i \), and \( P(X_{i+1}|X_i, X_{i-1}, \ldots, X_0) \) is the conditional probability distribution of the system’s state at \( t_{i+1} \), given all of the states that the system has been in until \( t_i \).

The memorylessness property asserts that the system’s next state depends only on its current state and is independent of its earlier history. This property is the main reason for choosing a Markovian framework for the scheduler.

**Definition 2.** Let \( < S, A, P_a(\ldots), R_a(\ldots), \gamma > \), be a Markovian Decision Process (MDP), where \( A \) is the set of actions and \( P_a(x,y) \) is the transition probability from state \( x \) to \( y \) which is equal to \( P(X_{i+1} = y|X_i = x, a) \), the conditional probability of transition from state \( x \) to state \( y \) given action \( a \in A \). Finally, the transition reward is denoted by \( R_a(x,y) \), and \( \gamma \in (0, 1] \) is the discount factor.

**Definition 3.** Action \( a_i \in A \) is admissible for \( < S, A, P_a(\ldots), R_a(\ldots), \gamma > \), if it is feasible, (i.e., it is possible to be taken at \( t_i \)), and progressively measurable, (i.e., depends only on the current state of the system, \( X_i \), not on the future states).

To schedule a telescope is to take an admissible action (e.g., to determine the next observation) at all decision steps \( t_i \). Notice that the decision steps \( t_i \), are not uniformly spaced. Decision steps are determined by the time that each observation takes. Also note that the scheduling is a finite time procedure from \( t_0 \) until the operation of the telescope is over, \( T \). Our framework is general and covers all possible discretizations of the time in both finite and infinite horizon scheduling tasks.

**Definition 4.** A deterministic policy \( \pi : S \rightarrow A_i \), is a mapping from \( S \) to the set of all admissible actions at \( t_i \), denoted by \( A_i \).

A policy provides a time-invariant law that for all possible \( x_i \in S \) suggests an admissible action. The policy is the heart of the scheduler, which takes the nec-
necessary information encoded in the current state and makes a decision for the next observation. The design of an optimal scheduler is mainly to find an optimal policy.

**Definition 5.** A deterministic optimal policy $\pi$ is a solution to the following optimization problem:

\[
\max_{\pi} E_{\pi}\left[\sum_{i=0}^{N} \gamma^i R_{\pi}(X_{i-1}, X_i)|x_0]\right],
\]  

where $x_0$ is a given initial state.

With this definition, we take the optimal policy to be a policy that maximizes the expected discounted sum of the rewards. The discount factor, $0 < \gamma \leq 1$, determines the priority of the overall gain (after an episode of observation) versus instant gains (after a single observation). Larger discount factors prioritize overall gains over instant gains. The choice of $\gamma$ depends on the application. It is usually empirically tuned and remains constant throughout the optimization.

**Proposition 2.2.1.** For the Markov decision process of $< S, A, P_a(., .), R_a(., .), \gamma >$, there exists a deterministic optimal policy, and it can be written as follows:

\[
\pi^* = \arg\min_{a_i \in A_i} E[\Phi(X_{i+1})|a_i],
\]  

where $\Phi : S \rightarrow \mathbb{R}$ is a function of the following form

\[
\Phi(x_i) = -R_{\pi^*}(x_{i-1}, x_i) + \gamma E_{\pi^*}[\Phi(X_{i+1})|x_i].
\]  

**Proof.** See Appendix A.

For the telescope scheduler we require the policy to be deterministic, because the simulations have to be repeatable for comparison and evaluation purposes. However, it can be shown that the deterministic optimal policy is optimal not only among
deterministic policies but also among stochastic policies. Therefore, the choice of
deterministic policy does not harm the optimality of the scheduler.

As a result of Proposition (2.2.1), search for the optimal policy of problem (2.1)
can be reduced from a search over the set of policies (all possible mappings from
the state space to the action space) to a search over $\Phi$ functions, without loss of
generality. This is a significant reduction made possible by the choice of Markovian
Decision framework.

2.2.2 Markovian Approximation

For a decision that is inherently time dependent, such as scheduling an observation,
only a maximal definition of the system’s state yields a perfect Markovian system.
The maximal definition of state space includes all of the possible decision sequences.
In particular, LSST requires a sequence of about 1000 decisions at each night. There-
fore, storing all possible scenarios requires a state space of size $N_f^{1000}$, where $N_f$
the number of tessellation centers on the visible sky. For any $N_f$, $N_f^{1000}$ number of
scenarios is neither tractable nor storable in a realistic memory. In order to over-
come the curse of dimensionality, we have designed a set of features to summarize the
state of the system with only the most determining information. Thus, the telescope-
environment is only an approximated Markovian system once its state space is reduced
to a feature space.

On the other hand, Proposition (2.2.1) shows that the search for an optimal
scheduler lies within the set of functions instead of a much larger set of mappings
from the state space to the action space. Despite this reduction, problem (2.1) is still
an infinite dimensional optimization problem, because its variable is a function.

To be able to numerically compute the $\Phi$ function, we assume that it can be
expressed as a linear weighted summation of some basis functions. With this structure
of $\Phi$, our optimization would be further reduced to find the optimal values of the
weights:

\[ \tilde{\Phi}_\theta(x_i) := \sum_{j=1}^{m} \theta_j \Phi_j(x_i), \]

where \( \theta \) is the vector of variables which fully characterizes \( \tilde{\Phi}(.) \), and \( \Phi_i(x_i) \)'s are the building blocks of \( \tilde{\Phi} \). We refer to \( \Phi_i(x_i) \)'s as basis functions which are handcrafted functions of the features and are designed in such a way that the domain knowledge of astronomical observation is incorporated into the decision-making strategy. For instance, if we define \( \Phi_1 \) to be the slew time from the current target to the desired target, the \( \Phi \) could be interpreted as the cost of this operation. However, \( \tilde{\Phi}_\theta \) does not necessarily have to carry a well-defined interpretation.

With this approximation, the search space is reduced from the space of functions to a finite-dimensional vector space. This approximation replaces the original optimal policy (2.2) with the following approximate policy:

\[
\tilde{\pi}^*_\theta(x_i) = \arg\min_{a_i} E[\tilde{\Phi}^*_\theta(X_{i+1})|a_i] = \arg\min_{a_i} \sum_{j=1}^{k} \theta^*_j E[\Phi_j(X_{i+1})|a_i],
\]

where \( \theta^* \) is a solution to the following optimization problem, in which policy \( \pi \) is fully determined by \( \theta \).

\[
\max_{\theta} E_{\pi^*_\theta} \sum_{i=0}^{N} \gamma^i R_{\pi^*_\theta(X_{i-1})}(X_{i-1}, X_i)|x_0].
\]

Note that the computational time to find the optimal policy could be relatively long, but it is an offline task and can be done before the telescope starts to operate. However, evaluating the action, given a policy at all \( t_i \) must be at least as fast as the length of the shortest observation. A linear policy not only demands relatively small computational resources to be optimized but also is very quick to be evaluated in real time.
2.3 Scheduler Optimization

In this section we introduce two different approaches to solve problem (2.5). The solution of this problem is an optimal set of weights, \( \theta^* \), for a given set of basis functions. The first optimization approach is faster but requires the high-level mission objectives to belong to a certain class of functions. The second optimizing approach is applicable to all types of the high-level mission objectives but requires more computational resources.

2.3.1 Reinforcement Learning

Assume that there exists a well-defined notion of an instant reward for each state transition; then, \( \Phi(x_i) \) by the definition given in Equation (2.3) is

\[
\Phi(x_i) = -R_{\pi^*}(x_{i-1}, x_i) + \gamma E_{\pi^*}[\Phi(x_{i+1})|x_i].
\]  

(2.6)

Accordingly, for the parameterized \( \Phi \) function, we require the following:

\[
\tilde{\Phi}_{\theta^*}(x_i) = -R_{\pi^*}(x_{i-1}, x_i) + \gamma E_{\pi^*}[	ilde{\Phi}_{\theta^*}(x_{i+1})|x_i].
\]  

(2.7)

The main idea behind reinforcement learning is to optimize \( \tilde{\Phi}_{\theta^*}(.) \) while the decisions are being made in a simulated environment. First, the policy is initialized by an arbitrary set of variables, \( \theta^0 \). Then, the first decision is made by the policy associated with \( \theta^0 \). Then, based on the outcome reward, we update the initial set of variables to get \( \theta^1 \). Next, a decision is made with a policy associated with the new set of variables \( \theta^1 \), which yields a reward. By repeating this process, we gradually update the variables in each decision step until they converge to an optimal value. The optimal \( \Phi \) respects Equation (2.7) for all \( i \in \{j : t_0 \leq t^j \leq T\} \).
Note that at $t_i$, after the transition from $x_{i-1}$ to $x_i$, we have the value of $R_{\pi_{\theta_i}}(x_{i-1}, x_i)$ already evaluated for making the decision. On the other hand, $\Phi_{\theta_i}(x_i)$ can be approximated by $E_{\pi_{\theta_i}}[\Phi_{\theta_i}(x_i)]$ which is also evaluated in the decision-making process, where $\theta_i$ is the last version of the optimization variables at $t_i$. Using the target value given in Equation (2.7), the update rule is as follows:

$$\tilde{\Phi}_{\theta_{i+1}}(x_i) = (1 - \alpha)\tilde{\Phi}_{\theta_i}(x_i) + \alpha(\gamma E_{\pi_{\theta_i}}[\tilde{\Phi}_{\theta_i}(x_{i+1})|\pi_{\theta_i}(x_i)] - R_{\pi_{\theta_i}}(x_{i-1}, x_i)),$$

in which $0 < \alpha < 1$ is the learning rate. The first term on the right-hand side of the equation is the most recent approximated value of $\tilde{\Phi}$ associated with $\theta_i$. This term has $(1 - \alpha)$ amount of contribution to $\tilde{\Phi}$ of the next iteration. The second term is the value of the target $\tilde{\Phi}$ (according to Equation (2.7)), which has $\alpha$ amount of contribution to $\tilde{\Phi}$ of the next iteration. Clearly, with smaller $\alpha$ values this update imposes smaller adjustments from one decision to the next. The choice of $\alpha$ depends on the application. Higher $\alpha$ values are computationally preferred because they speed up the optimization; however, depending on the natural dynamic of the system and how much it changes from one decision to another, higher $\alpha$ values could make the process of optimization unstable and diverging. Accordingly, updates of the variables $\theta_j$, $j = 1, \ldots, k$ can be expressed as follows:

$$\theta_{j}^{i+1} = \theta_{j}^{i} + \left( \tilde{\Phi}_{\theta_{i+1}}(x_i) - \tilde{\Phi}_{\theta_{i}}(x_i) \right) \Phi_j(x_i)$$

$$= \theta_{j}^{i} + \alpha \left( - \Phi_{\theta_{i}}(x_i) + \gamma E_{\pi_{\theta_i}}[\tilde{\Phi}_{\theta_i}(X_{i+1})|x_i] - R_{\pi}(x_{i-1}, x_i) \right) \Phi_j(x_i).$$

This variant of reinforcement learning is called Temporal-Difference (TD) learning with function approximation [Tsitsiklis and Van Roy, 1997]. Variants of this approach have been successfully applied to real-life problems such as training of a backgammon player [Tesauro, 1995].
Recall that in order to be able to use the TD reinforcement learning method, it is necessary to have a well-behaved notion of a reward that reflects the instant gain of all possible decisions at all of the decision steps. Moreover, the discounted sum of the instant rewards has to reflect the objective of the mission according to Equation (2.1). For instance, in the LSST scheduling problem, after each visit, the negative of the slew time is a well-defined instant reward that reflects how time-efficiently the telescope is being operated. This, however, does not reflect all aspects of the high-level mission’s objective such as the need to reobserve a field within a valid time window (explained in Section 2.4), for which there is no equivalent instant reward. For this reason, we also introduced a black-box function optimizer in the following section. We use the black-box approach when there is no well-defined notion of an instant reward with which we can build the high-level mission’s objective function.

2.3.2 Global Optimization

In the absence of a well-defined instant reward, instead of solving problem (2.5), the following problem can be solved:

$$\max_{\theta} U_{\pi_{\theta}}(x_i, x_{i+1}, \ldots, x_j), \tag{2.10}$$

where $U_{\pi_{\theta}}(x_i, x_{i+1}, \ldots, x_j)$ is a utility function that measures the performance of the scheduler on a simulated episode of the operation from $t_i$ to $t_j$. Unlike the previous method, the policy, $\pi_{\theta}$, is fixed during an episode of simulation in this approach. The simulation episode is usually a small fraction of the total time of the telescope’s operation, $T - t_0$. The idea is to test the performance of a set of candidate policies in a parallel manner, within a short episode. Then, we infer a better set of candidates and repeat until the performance cannot be improved. In general, $U(.)$, cannot be explicitly expressed as a function of $\theta$, therefore, a global optimizer that
can maximize a black-box function is required. Evolutionary optimizers have successfully been applied to numerous real-life problems involving black-box function optimization, and specifically astronomical mission planning such as the scheduling of the Exoplanet Characterization Observatory \cite{Garcia-Piquer2015}. We used the eDE evolutionary optimizer \cite{Naghib2016} which is an adaptive version of the Differential Evolution (DE) algorithm \cite{Storn1997}. DE is generally one of the most efficient evolutionary algorithms, and the eDE variant uses a notion of entropy to automatically preserve the diversity of the candidate solutions. As a result, in contrast with DE, it does not require the user to choose any tuning parameters for the algorithm, which is the most time-consuming task in using an evolutionary optimizer. In addition, eDE, similar to any other evolutionary algorithms, is highly parallelizable, and the computational time can be almost linearly decreased with respect to the number of computational cores.

### 2.4 Problem of Scheduling for the LSST

The LSST’s mission is to uniformly scan the visible sky within five different regions shown in Figure\ref{fig:lsst_regions}. Each region, also referred to as a survey, has certain science-driven goals and constraints, defined and precisely described in \cite{Ivezic2008}.

The notion of the features enables the scheduler to systematically fetch all of the various bits of information and turn them into comparable quantities for the purpose of the decision-making. The proposed feature space of the LSST contains seven features (see Table\ref{tab:features}), each of which can be evaluated given a field \(i\), a filter \(f\), and a time \(t\). The fields discretize the visible sky through a fixed sky tessellation. There are six choices of filter, \([u, g, r, i, z, y]\), for each visit. Finally, the time is discretized by the natural timing of the process. In other words, the time interval between two consecutive decision steps is the time that it takes to execute the observation.
Table 2.1: Key Terms and Notations Used in the Definition of the Features and the Basis Functions

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fields</td>
<td>Point configuration on the sky</td>
</tr>
<tr>
<td>$N_f$</td>
<td>Total number of the fields</td>
</tr>
<tr>
<td>$t$</td>
<td>Coordinated Universal Time (UTC)</td>
</tr>
<tr>
<td>$\tau_n(t)$</td>
<td>Beginning (end) of the night that $t$ lies within</td>
</tr>
<tr>
<td>$\tau_{\text{rise(set)}}(i, f, t)$</td>
<td>Rising (setting) time of field-filter $(i, f)$ before $\tau_n(t)$</td>
</tr>
<tr>
<td>$id(t)$</td>
<td>ID number of the field that is visited at $t$</td>
</tr>
<tr>
<td>$ft(t)$</td>
<td>Camera’s filter at $t$</td>
</tr>
<tr>
<td>$n(i, f, t)$</td>
<td>Total number of the visits of field-filter $(i, f)$ before $t$</td>
</tr>
<tr>
<td>$\text{slew}(i, j)$</td>
<td>Slew time from field $i$ to field $j$ in seconds</td>
</tr>
<tr>
<td>$\text{settling}(i, j)$</td>
<td>Mechanical settling time after slewing from field $i$ to field $j$</td>
</tr>
<tr>
<td>$\Delta t_f$</td>
<td>Time needed to change filter, a constant value about 2 minutes</td>
</tr>
<tr>
<td>$t_{\text{dome}}(i)$</td>
<td>Time needed to move the dome to make field $i$ visible to the telescope</td>
</tr>
<tr>
<td>$ha(i, t)$</td>
<td>Hour angle of the center of field $i$ at $t$ in hours, $-12 \leq ha(i, t) \leq 12$</td>
</tr>
<tr>
<td>$am(i, t)$</td>
<td>Air-mass of the center of field $i$ at $t$</td>
</tr>
<tr>
<td>$br(i, t)$</td>
<td>Brightness of the sky at the center of field $i$ at $t$</td>
</tr>
<tr>
<td>$\sigma(i, t)$</td>
<td>Seeing of the sky at the center of field $i$ at $t$</td>
</tr>
<tr>
<td>$K(i, f, t)$</td>
<td>Atmospheric extinction coefficient at the center of field $i$ at $t$</td>
</tr>
<tr>
<td>$W_1, W_2$</td>
<td>Given constant time window within which a revisit is valid</td>
</tr>
</tbody>
</table>

In between the two decisions. Given that a consecutive visit of the same field-filter is not allowed in the main survey, there is a slew time between any two decisions, and therefore $t_j - t_{j-1} > 0$. On the other hand, the operation is over a limited time horizon, $T$; thus the number of the decision time steps is finite. In conclusion, a finitely discretized sky, a finite number of filters and a finite number of time steps pose a finite feature space, denoted by $\{(f_1(i, f, t_j) \ldots f_7(i, f, t_j)) : i = 1 \ldots n_f, f \in \{u, g, r, i, z, y\}, j = 0, \ldots N\}$. With this feature space the implication of the policy, stated in Equation (2.4), is as follows:

$$\tilde{\pi}^* (x_j) = \arg\min_{(i, f) \in A_j} \sum_{k=1}^{5} \theta_k E \pi_k [\Phi_k (X_{j+1}) | x_j],$$

(2.11)

where $x_j = [f_1, \ldots, \tau(id(t_j), ft(t_j), t_j)]$ is a seven-dimensional state at $t_j$, and $(i, f)$ is an admissible field-filter pair. Section 2.4.3 introduces the constraints under which a
Figure 2.1: Regions of the sky with different requirements and constraints for scheduling: (1) Galactic Plane (GP), (2) Universal or Wide Fast Deep (WFD), (3) South Celestial Pole (SCP), (4) North Ecliptic Spur (NES), and (5) Deep Drilling Fields (DD).

field-filter pair is admissible at $t_j$. Accordingly, $\mathcal{A}_j$ is the set of all field-filter pairs that are admissible at $t_j$.

In the implementation of the scheduling software we took a modular approach. The expected values of the basis functions, $E_{\pi}[\Phi_k(x_{j+1})|x_j]$ for $k = 1, \ldots, 5$, are evaluated in separate software routines. Then, they are delivered to the scheduler at the stage of the decision. The basis functions that address the environmental parameters are developed by the LSST community. For example, see Gressler et al. (2014) for the parameters that capture the status of the LSST site, Sebag et al. (2008) and Sebag et al. (2007) for cloud cover measurements that were used to develop a predictive cloud model, and see Yoachim et al. (2016) for the sky brightness model.

Generally speaking, making a decision for a visit at $t$ for the LSST scheduling problem is mainly determined by the following factors:
1. The amount of the time it takes to redirect the telescope and the dome to move from one target to the next target (slew time).

2. The short-term science-driven requirements, such as the same-night revisit of a field.

3. The long-term mission-driven requirements, such as maintaining a uniform coverage of all field-filter pairs within each region.

4. The observational quality of a field-filter pair at the time of decision, such as the expected depth of the resulting image.

5. The general preference for observing the fields around the meridian.

Accordingly, the basis functions of the LSST scheduler are designed to formalize the above factors. For the full definition of the basis functions of the Feature-based scheduler for LSST refer to Section 2.4.2.

In what follows, first we precisely define the features and basis functions of the LSST scheduler, and then we show how the two optimization procedures, described in Section 2.3, are applied. The optimization results are associated with two sample objective functions. However, the LSST community, and principally any individual, can design their own mission objective function and repeat the optimization procedure to obtain a scheduler that optimizes the probabilistic expectation of their objective function.

2.4.1 Features of the Telescope Scheduler

For designing the features, it is important to avoid redundancy in the information that features contain. It is also critical to hold a modular approach in the delivery of the information to the decision stage. For instance, consider the amount of time, $\Delta t$, it takes for a telescope to move from one visit to another. In the LSST problem, $\Delta t$
mainly depends on the slew time, the mechanical settling time, the dome placement
time, and the time it takes to change the filter. All of these timings are available
through a precise simulation of the LSST model (Delgado et al., 2014). A modular
design would be to bring the summation of the operational timings to the stage of
the decision instead of bringing them separately as different features. This approach
makes the implementation significantly simpler and more readable. Conceptually,
the modular approach makes it possible to track the effect of the operational costs
of a similar nature in the overall performance of the scheduling. Particularly, the
operational cost is independent of the amount to which each cause contributes to the
overall $\Delta t$. Hence, bringing the timing of each procedure separately in the decision
making level adds unnecessary complications to the design.

Using the notations in Table 2.1, we propose seven features for the LSST scheduler
in Table 2.2. Features are carefully designed to efficiently carry the determining
information with a modular approach. Each feature is denoted by $f_k(i, f, t)$ for $k =
1..7$ and indexed by the triplet of $(i, f, t)$, field, filter, and time. To make a decision
at $t$, the scheduler computes seven features for all of the $(i, f)$ pairs.

There are some features that do not change in every time step, for instance, if $i$
is not visited at $t^j$, then $f_5(i, f, t^j) = f_5(i, f, t^{j+1})$. For such cases, the implementation
has a categorized updating routines to avoid redundant computations. For features
such as the cloud coverage that are inherently random variables, we have separate
predictive modules that evaluate the expectation of their values at the time of the
next observation.

The main computational time during the real-time scheduling is spent on the
evaluation of the features. Some features such as the slew times are essentially look-
up tables. Some features such as the co-added depth, are stored in the memory and
indexed by a pair of field-filter, which the scheduler updates for only one field-filter
after every decision step. The rest are continuous variables such as the location of
Table 2.2: Features of the Approximated Markovian Model for the Telescope-environment System

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(i, f, t)$</td>
<td>$(\text{slew}(id(t), i) + \text{settling}(id(t), i) + \Delta t_f I_{f(t) \neq f}) \vee t_{\text{dome}(i)}$: either the time required to point the telescope to $i$, and change the filter to $f$, or the time required to relocate the dome to make $i$ visible, whichever that is larger</td>
</tr>
<tr>
<td>$f_2(i, f, t)$</td>
<td>Total number of the same-night visits of field-filter $(i, f)$ until $t$</td>
</tr>
<tr>
<td>$f_3(i, f, t)$</td>
<td>$(t - \tau_n(i, f, t)) I_{{\theta(i, f, t) &gt; \tau_i(i)}}$, time since the last same-night visit of $(f, i)$</td>
</tr>
<tr>
<td>$f_4(i, f, t)$</td>
<td>Remaining time for field-filter $(i, f)$ to become invisible, either by passing the air-mass or the moon-separation limit or by being covered by temporary objects such as clouds, as projected at $t$</td>
</tr>
<tr>
<td>$f_5(i, f, t)$</td>
<td>Co-added depth, a measure of cumulative quality of past visits of field-filter $(i, f)$ until $t$</td>
</tr>
<tr>
<td>$f_6(i, f, t)$</td>
<td>$5\sigma$-depth, a measure for quality of visiting field-filter $(i, f)$ at $t$, depending on seeing, sky brightness, and air-mass; $f_6(i, f, t) = C_m + 2.5 \log(\frac{0.7}{\sigma(i,t)}) + 0.50(br(i, t) - 21) - K(i, f)am(i, t)$, where $C_m$ is a scaling coefficient</td>
</tr>
<tr>
<td>$f_7(i, t)$</td>
<td>Hour angle of field $i$ at $t$</td>
</tr>
</tbody>
</table>

the clouds, which can be interpolated from their past values, even if the exact value at a given time is not available.

2.4.2 Basis Functions of the Telescope Scheduler

Basis functions, $\Phi_k$ for $k = 1 \ldots 5$, are fully determined by the value of the features. Hence, they are indexed in the same way that features are, by a triplet of $(i, f, t)$, field, filter, and time. Similar to the update of the features, all five basis functions should be evaluated for all pairs of $(i, f)$ for a decision at time $t$, except that in this case admissibility of a field-filter can be evaluated. (see Section 2.4.3 for the list of constraints that ensure admissibility). Therefore, while it is required to evaluate the features for all possible pairs of $(i, f)$ at all decision steps, the number of the basis function evaluations is on average a factor of three times less than the number of the feature computations.

Common basis functions are shared amongst all of the regions of the sky. They are designed to reflect the five general decision factors described in Section 2.4.
The LSST’s mission poses different requirements on different regions of the sky, shown in Figure B. First, we modify $\Phi_2$, for the Wide Fast Deep (WFD) and North Ecliptic Spur (NES) regions. Because they require the telescope to observe a field twice on the same night\(^2\) within a valid time window, $[W_1, W_2]$. The following modification prioritizes the fields that have received a first visit but not a second visit on the same night. There are two regimes to prioritize these fields, encoded in $\mathbb{1}_{(f_i \geq W_2)}$. This function is zero if the expected remaining time for a field to become invisible, $f_4$, is less than $W_2$; hence, the associated cost is minimal ($=0$) to ensure that this field will receive its second visit before it becomes invisible. If $f_4$ is larger than or equal

\(^2\)Later, during the development of the scheduler software it was decided that this constraint should be applied to all of the regions.
to $W_2$, it means that the expected remaining time for the field to become invisible is longer than the revisit deadline; therefore, $1_{(f_4 \geq W_2)}$ is one and the cost is evaluated by a different regime defined by $\Phi^\text{pair}$ in the following:

$$
\Phi^{WFD}_2(f_2, f_3, f_4) = \begin{cases} 
\Phi^\text{pair}(f_3)1_{(f_4 \geq W_2)}, & \text{if } \sum_f f_2 = 1, \\
\Phi_2(f_2), & \text{else},
\end{cases}
$$

where, $\Phi^\text{pair}(f_3) = \exp(-\frac{\text{min}_\phi f_3(i, \phi, t)}{W_2})$. Fields that have received their first visit of the night are distinguished via $\sum_f f_2 = 1$, the sum of the number of visits in all filters. Finally, $\Phi^\text{pair}$ prioritizes the fields that received their first visit earlier, because they are more likely to miss the valid time window of the revisit.

The Deep Drilling Field (DDF) region contains a very small fraction of the visible sky’s area (it is about 10 individual fields). Hence, it is unnecessary to adjust the basis functions that yield a separate generic policy for such fields. Instead, we treat the observation of DDFs as interruptions to the scheduler’s regular operation, with each interruption comprising a sequence of DDF observations. This scheme is computationally more efficient and reduces the structural complexity of the scheduler. The recoverability attribute of the Feature-based scheduler enables the scheme of the interruptions to be a part of the optimal scheduling.

**Controllability of the scheduler**

As discussed in Section 2.2, the scheduler is optimizable if the telescope is controllable. In this section we discuss the empirical controllability of the LSST given the Feature-based scheduler. What we observe are the variations of two sample objective functions with respect to the variations of the design parameters, $\theta$. If there are no meaningful variations, then the telescope is not controllable with this scheduler. As a result, the scheduler is not amenable to any form of optimization. On the other hand, if
the objective function is extremely variable with respect to the changes in the design parameters, the solution of the optimization is not fully reliable, because the objective is not a well-behaved function of the optimization variables.

In order to observe the variability of the objective functions with respect to changes in the scheduler’s design parameters, first we defined a sequence of equidistant values for \( \theta_i \in \{0, 0.5, \ldots, 8.5, 9\} \) and kept the other \( \theta \) values fixed at 2 or 5 or 8 in separate experiments. Then, we scheduled a simulated episode of observation for \( t_n - t_0 = 4.8 \, hr \), for each \( \theta_i \) separately. Finally, we evaluated the objective functions \( U_1 \) and \( U_2 \) of each schedule, where,

\[
U_1(x_0, x_1, \ldots, x_n) = \sum_{\{i: t_0 < t^i < t_n\}} -slew(id(t_i), id(t_i)) - 10am(id(t_i)), \quad (2.12)
\]

\[
U_2(x_0, x_1, \ldots, x_n) = n. \quad (2.13)
\]

The first objective function reflects the slew time, \( slew() \), and air-mass, \( am() \), averaged, and \( U_2 \) reflects the time efficiency of the operation by counting the total number of observations. Note that the second objective function belongs to the class of black-box functions, and the first one belongs to the other class where there exists a decomposition based on the discounted sum of the rewards.

Figure 2.2 contains slices of the 5-dimensional \( U_1 \) and \( U_2 \). Both of the simple objective functions reasonably respond to the changes in all five dimensions of the variable \( \theta \), which is the evidence of controllability. Moreover, the smaller variations for slices closer to the boundaries of the search space suggest that the design and scaling of the basis functions provide a desirable behavior within the proposed search space.
Figure 2.2: One-dimensional slices of two objective functions, $U_1$ and $U_2$, defined in Equations (2.12) and (2.13), respectively. The variation of the objective functions, especially in the midrange slices (solid line), suggests that the LSST is controllable with the Feature-based Scheduler.

### 2.4.3 Survey-specific Constraints

The scheduler’s decision at each time step is an admissible (feasible and measurable) pair of field-filter $(i, f)$. Feasibility of a candidate $(i, f)$ is driven by the following factors:

- **Visibility**: the candidate field-filter has to be visible.
- **Quality**: the expected observational quality of a field-filter, such as the expected depth of the resulting image, has to be better than a given threshold.
Table 2.4: Feasibility of Field-Filter \((i, f)\) for a Visit at \(t^{n+1}\)

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Description</th>
<th>region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (\frac{\tau_{\text{rise}}(i, f, t^{n+1})}{\tau_{\text{set}}(i, f, t^{n+1})}) (\leq) (t^{n+1}) (\leq) (\frac{\tau_{\text{rise}}(i, f, t^{n+1})}{\tau_{\text{set}}(i, f, t^{n+1})})</td>
<td>field-filter ((i, f)) has to be above the acceptable airmass horizon at (t^{n+1}).</td>
<td>All regions</td>
</tr>
<tr>
<td>2 (E[f_1(i, f, t^{n+1})] \neq 0)</td>
<td>field-filter ((i, f)) is not temporarily masked (e.g., by the moon) at (t^{n+1}).</td>
<td>All regions</td>
</tr>
<tr>
<td>3 (\sum f_2(i, f, t^{n}) &lt; N_{\text{region}})</td>
<td>(N_{\text{region}}) is a region dependent upper-bound on the number of the visits for each field in that region. (N_{\text{WFD}} = N_{\text{NES}} = 3), and (N_{\text{GPR}} = N_{\text{SCP}} = 1).</td>
<td>All regions</td>
</tr>
<tr>
<td>4 (E[f_6(i, f, t^{n+1})] &lt; \sigma_{\text{(region, f)}})</td>
<td>the expected quality of visiting field-filter ((i, f)) at (t^{n+1}) has to be better than the given threshold, (\sigma_{\text{(,)}}), that depends on the region and the filter.</td>
<td>All regions</td>
</tr>
<tr>
<td>5 (f \neq \text{id}(t^{n}))</td>
<td>consecutive visit of a same field is not allowed.</td>
<td>All except DD</td>
</tr>
<tr>
<td>6 if (\sum f_2(i, f, t^{n}) = 0) then (\max_{\phi} f_4(i, \phi, t^{n}) &gt; \frac{W_1 + W_2}{2})</td>
<td>the first visit of field (f) has to occur (\frac{W_1 + W_2}{2}) time before it becomes invisible, so that the second visit of (f) can be scheduled in the valid time window.</td>
<td>WFD and NES</td>
</tr>
<tr>
<td>7 if (\max_{\phi} \theta(i, \phi, t^{n}) &gt; \tau_s(t^{n})) then (W_1 \leq \min_{\phi} f_2(i, \phi, t^{n}) \leq W_2)</td>
<td>if there has been a same-night visit of field (f) until (t^{n}), then the next same-night visit has to occur in the valid time window.</td>
<td>WFD and NES</td>
</tr>
<tr>
<td>8 if (\max_{\phi} \theta(i, \phi, t^{n}) &gt; \tau_s(t^{n})) then (f \notin {y, u})</td>
<td>if there is a same-night visit of field (f) until (t^{n}), then the next same-night visit cannot be with either of (u) or (y) filters.</td>
<td>WFD</td>
</tr>
<tr>
<td>9 (f \notin {y, u})</td>
<td>visits with (u) filter and (y) filter is not allowed.</td>
<td>NES</td>
</tr>
</tbody>
</table>

*We use the probabilistic expectation, \(E[\cdot]\), of the stochastic values, as evaluated at \(t^n\). *Later, during the development of the scheduler software, it was decided that the 6\(^t\)\(h\) and the 7\(^t\)\(h\) constraint should be applied to all of the regions.

- Survey’s timing: the science-driven revisit constraints has to be respected.

To ensure measurability, the above criteria must be evaluated based only on the information that is encoded in the feature space. Exact expressions of the proposed constraints for the LSST scheduler are presented in Table 2.4.
2.4.4 Scheduler Optimization

In this section, we present two simple choices for the high-level mission objective to demonstrate the application of the proposed optimization approaches, discussed in Section 2.3. (More sophisticated mission objective functions can be defined based on the LSST performance studies such as Grav et al. (2016), Graham et al. (2018), Jacklin et al. (2017), and Oluseyi et al. (2012)). The choice of optimization algorithm depends on the nature of the mission objective. The first experimental mission objective function in this section can be expressed as the discounted sum of instant rewards $R(s_{i-1}, s_i)$, thus, the reinforcement learning is applied to find the scheduler’s parameters $\theta$. The second objective function cannot be decomposed as a discounted sum of instant rewards; thus, we used the global optimizer approach. From the computational point of view, the first approach is preferred. For the following experiment, the reinforcement learning is about 10 times faster than the global optimization and requires 50 times less memory. From the practical point of view, however, for some missions it is impossible to define an objective function that can be expressed as a discounted sum of a well-defined instant reward. In that case, the mission objective can be optimized only via a global optimizer.

In the following experiments, for both of the optimizations we used a simulated model of the telescope (Connolly, 2014) and the environment which are developed based on the measurements at the LSST site. In particular we use the models that are developed to predict the brightness of the sky and the coverage of the clouds.

Reinforcement Learning for the First Choice of Mission Objective

Let the instant reward, $R(i-1, i)$, be $-slew(id(t_{i-1}), id(t_i)) - am(id(t_i))$. It is defined as a linear combination of the slew time to point the telescope from the $(i - 1)$th field to the $i$th field and the air-mass of the destination. Since both factors have a negative effect on the overhead and quality of the observation, the reward would be measured
by the negative of each. Then, the mission objective function can be simply defined as \( \sum_{i=0}^{N} \gamma^i R(i - 1, i) \).

The simulation for the reinforcement learning starts at \( t_0 = 2462867.5 \ mjd \) (2021 January 1), with \( \theta^0 = (5, 5, 5, 5, 5) \), initialized at the midrange values, and continues until \( \theta \) converges. Figure 2.3 is the training curve for all of the variables over a course of 3000 decisions. The discount rate \( \gamma = 0.9 \), and learning rate \( \frac{0.01}{\log^3(i)} \) are chosen empirically.

Figure 2.3: Reinforcement learning of the scheduler’s parameters, \((\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)\). All of the parameters are initialized at the midrange value, 5. During the simulation, after each decision step there is a reward associated with the outcome of the decision that implies a small adjustment on each of the five design parameters. Then, the next decision will be taken with a slightly different set of parameters. This procedure continues until the adjustments on the variables are negligible.

For the above choices of reward, learning rate, discount factor, and initialization, \( \theta \) converges to \( \theta^* = (8.18, 1.04, 3.26, 7.59, 1.13) \). With a personal computer, each

---

\(^3\)Processor: 1.6 GHz, Memory 1600 MHz DDR3
decision and its associated update takes about 0.8 sec thus, the time of the convergence for the simulation presented in this section is $3000 \times 0.8 \text{ sec} = 40 \text{ minutes}$. For this experiment we stop the experiments when the change in the $\theta$ values are less than a small number. Note that the optimization time is linear with respect to the number of decision steps.

**Global Optimization for the Second Choice of Mission Objective**

One of the important simple objective functions that cannot be expressed as the discounted sum of the rewards is the total number of the observations from a given $t_i$ to a given $t_j$, which can be expressed as $U_{\pi_0}(x_i, x_{i+1}, \ldots, x_j) = j - i$. To find a set of parameters, $\theta$, that optimizes this objective function, we applied the global optimization approach, explained in Section 2.3. To decrease the computational time, we used the following regulatory constraints:

- $\theta \geq 0$: Positive coefficients for the basis functions are assumed in the design of the basis functions, because in the context of the telescope scheduling, it is more natural to create the basis functions to reflect the cost of the operation.

- $\theta_1 = \theta_0$: Without loss of generality, we fix the value of the first element of $\theta$ to reduce the dimension of the optimization problem by one. Because, homogeneity of the policy implies that if $\theta^*$ yields an optimal scheduler, then $\alpha \theta^*$ for $\alpha > 0$ yields an optimal scheduler as well.

We defined the above objective function, $U_{\pi_0}$, over a period of 10 days, from $t_i = 2462867.5 \text{ mjd} \ (2021 \text{ January 1})$ to $t_j = 2462877.5 \text{ mjd} \ (2021 \text{ January 11})$. Figure 2.4 shows the value of this objective function at each iteration of the eDE algorithm. The solution $\theta^* = (1.00, 0.84, 0.99, 1.34, 3.04)$, yields the best $U_{\pi_0}$ after 50 iterations.

This reflects the efficiency of the first scheduler’s prototype. The time of each decision simulation for the current efficient software is about 0.036 s.
Figure 2.4: Progress of the black-box objective function $U_{\pi_0}$ over the iterations of the $e$DE algorithm. $U_{\pi_0}$, for this simulation is the total number of the observations for 10 nights starting from 2021 January 1.

$e$DE is a population-based metaheuristic algorithm. For our experiment, shown in Figure 2.4, the number of populations $N_P$ is set to be 50. Each function evaluation is in fact the simulated operation of the telescope for 10 nights, with a candidate scheduler which takes about 8 minutes. Therefore, each iteration of the algorithm $e$DE takes $N_P \times 8$ minutes. The optimization can be manually terminated if the result is satisfactory, or it can be continued until a full convergence is achieved. In $e$DE (and all genetic algorithms in general), function evaluation for each individual is independent from other individuals; therefore, the parallel implementation of the same algorithm can be faster up to a factor of $N_p$. 
2.5 Performance of a Modified Feature-Based Scheduler for LSST

In this section, the LSST Metric Analysis Framework \cite{Jones2014} is used, to compare the performance of a modified version of the Feature-based scheduler with opsim V4 and opsim V3, the most recent producers of the baseline schedules of the LSST. (We use two specific sequences produced by each of these scheduler. Namely, astro-lsst-01-2013 and minion-1016 produced respectively by opsim V4 and opsim V3.)

The following results are based on the simulations that use a computer model of the LSST-environment system. The resulting schedule, however, will not be used in practice, because it is made by some realization of the random processes such as the cloud coverage. For the real-life operation of LSST, we use the same approach to optimize the scheduler; however, the decisions are made on the fly, using the real-time values of the features.

Recall that the optimization part consumes the main portion of the total computational time. Fortunately, the scheduler’s optimization is an offline process and can be done before the telescope is ready to operate. On the other hand, the real-time decision-making, with an already optimal scheduler is a very fast process. It only includes evaluation of the features and basis functions. The current version of the Feature-based scheduler can schedule 28 observations per second, while the length of each observation in real-time is at least 30 s. Hence, the decision-making speed is roughly 900 times faster than what is required for the real-time scheduling. However,

\footnote{The sky background models and weather downtime used to benchmark the algorithms are not exactly identical because of the practical difficulties in the separation of the environment and the opsim scheduler. However, for the purpose of the comparisons in this section, the behavior of our sky and observatory model is sufficiently close to the official model. See \cite{Delgado2016}, and \cite{Reuter2016} for the official operation simulator.}
we are still working to increase the time efficiency of the software, because it directly effects the computational time of the offline optimization procedure.

The Modified Feature-based scheduler is under active development[^1] and it addresses the observational details of the LSST’s mission through the adjustment of the constraints and the basis functions. It is designed to produce a software that can be used in practice.

Unlike the default sky tessellation for LSST, we do not require the tessellation centers to be determined based on the telescope’s field of view. The default sky tessellation adopted in the baseline scheduler results in 23% of the sky being covered by more than one field, which causes serious nonuniformity in the final coverage of the sky. In the modified Feature-based scheduler, we adopt a 50 times finer discretization of the sky using Hierarchical Equal Area isoLatitude Pixelization (HEALpix) (Górski et al., 2005). The fact that the decision-making is not computationally expensive makes it possible to use a much finer discretization of the feature-space resulted by the finer tessellation of the sky. This approach allows the scheduler to handle the fields’ overlaps which cause inhomogeneity of the final sky coverage.

In addition to adopting a finer discretization, we use a spatial dithering scheme to randomize the final pointing of the telescope by a small amount around the tessellation centers to further assist the homogeneity of the coverage. Adopting the dithering scheme, the median number of observations at a typical point in the sky increases by $\sim 15\%$. Dithering is also essential for removing systematic effects for science cases such as measuring galaxy counts (see Awan (2016) for more details). Moreover, the modified Feature-Based scheduler uses separate routines to decide whether an observation will need to be observed in a pair and to decide whether a Deep Drilling sequence should be executed by interrupting the normal operation of the telescope.

[^1]: GitHub repository: [https://github.com/lsst/sims_featureScheduler](https://github.com/lsst/sims_featureScheduler)
2.5.1 Sky Coverage Uniformity

For a survey telescope, such as the LSST, the density of the co-added depth over the visible sky should ideally be uniform in each filter and within each of the five survey regions. Figure 2.5 compares the values of the co-added depth on a discretized full sky map. Figure 2.6 demonstrates the smoothness of the coverage by zooming in to a smaller area of the sky map. It shows the co-added depth values of opsim V4, with and without dithering, and compares them with that of the Modified Feature-based scheduler in the r band. The zoomed-in area is around the boundary of WFD and GP regions, which have different target co-added depths. The smoother coverage that the Modified Feature-based scheduler offers is due to the fine discretization of the sky in the decision-making stage, in addition to the dithering that takes place after the decision is made.

Figure 2.7 compares the distribution of the co-added depth on a (finely) discretized sky. The Modified Feature-based scheduler has paved the rightmost peak that appears in the distribution of opsim V4. This peak is the result of the field’s undesirable overlaps. Table 2.5 contains the median and variance of the co-added depth for both of the schedules in each of the sky regions and in each filter. The Modified Feature-based scheduler provides deeper (higher median) and more uniform (lower variance) coverage in most of the cases.

2.5.2 Pairs

In addition to uniformity of the coverage, the LSST mission calls for pairs of visits within a valid time window on the same night. The main reason is to detect the transient and moving objects. Because the moving objects usually belong to the solar system, the pair constraint was initially imposed only on the WFD and NES regions. However, there are interesting solar system objects, such as interstellar asteroids, that can be observed in any direction of the sky. In addition, identification of the other
Figure 2.5: Sky coverage in each of the six filters \([u, g, r, i, z, y]\), measured by co-added depth. According to the mission’s objective, the scheduler has to provide a uniform coverage of the visible sky within each region and in each filter. The left panels show the opsim V4 simulation results, while the Modified Feature-based scheduler is on the right. Even without a given observation proposal, the Modified Feature-based scheduler can closely match the large-scale footprint of the official survey. (The individual yellow dots with high co-added depth are fields in the Deep Drilling survey, and the NES region is not visited in the \(u\), and \(y\) bands.)

Varying objects, such as supernovae, can benefit from a follow-up visit, especially if the second visit is with a different filter. Thus, in the Modified Feature-based scheduler we made the pair constraint a universal constraint for all of the regions. The downside of this extension is the fact that it constrains the scheduler even more and the performance can be potentially less than it could be.

Note that the structure of the Feature-based scheduler allows for extension or restriction of the constraints down to the level of an individual field-filter. These extensions and restrictions do not contradict any of the Markovian assumptions, and do not break the structure of the implementation.
Figure 2.6: Co-added depth coverage in a zoomed-in area of the sky map, around the border between the WFD (green) and SCP (blue) regions, in the $r$ band. The smooth coverage of the Modified Feature-based scheduler (right) vs. the granular pattern of opsim V4 (left) further respects the uniformity of the coverage, which is one of the most fundamental objectives of a survey instrument. The middle panel is the coverage of opsim V4, with dithering of the same sequence of observations that fundamentally cannot become as smooth as the right panel, because, unlike Feature-based scheduler, the scheme of opsim V4 does not easily allow for decision-making with arbitrarily fine tessellations of the sky.

Figure 2.8 demonstrates the distribution of observations in pairs (in the $g$, $r$, and $i$ filters) to the total number of the observations. For the regions to which the pair constraint is applied, this ratio can be interpreted as the success rate of the schedulers in satisfying the pair constraint. This success rate ranges from 0 to 1, and the higher values indicate the more successful pair visits.

Figure 2.9 compares the distribution of the pair ratio of the Modified Feature-based scheduler and opsim V4 on a (finely) discretized sky. Note that the peak of the distribution for the Modified Feature-based scheduler is closer to 1, which means that a larger area of the sky is covered by a successful pair visit; however, opsim V4 offers a sharper concentration of the values that can be interpreted as a more homogenous pair visit, which agrees more with the LSST’s mission to survey the sky in a uniform manner.
Figure 2.7: Each plot compares the distributions of the co-added depth coverage in one of the six filters. A dithering scheme in the Modified Feature-based scheduler in addition to a finer tessellation of the sky smoothens the density of the coverage where the fields overlap.

2.5.3 AltAz and Air-mass Distributions

Air-mass is one of the major obstacles for ground-based instruments. Zenith observations have the minimum air-mass; however, off-zenith observations cannot be avoided, due to time-efficiency and design limitations. In this case, observations around the meridian provide high-quality images and consequently result in more efficient operation of the instrument. Figure 2.10 compares the number density of the visits on an altitude-azimuth sky map in each of the six filters \([u, g, r, i, z, y]\). Clearly, in all of the filters the Modified Feature-based scheduler schedules more visits around the meridian zone. In addition, it offers a consistent concentration peak on the east wings, which is essential for a higher success rate of the pair constraint. If the first visit of the night occurs when the field is on the east side of the sky, the scheduler has a longer opportunity to schedule the second visit of the night, and hence it is more likely to perform a successful pair visit. Figure 2.11 demonstrates the density
<table>
<thead>
<tr>
<th>f</th>
<th>WFD</th>
<th>GP</th>
<th>SCP</th>
<th>NES</th>
<th>WFD</th>
<th>GP</th>
<th>SCP</th>
<th>NES</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>25.63, 0.04</td>
<td>25.12, 0.11</td>
<td>24.91, 0.12</td>
<td>-</td>
<td>25.68, 0.01</td>
<td>25.32, 0.05</td>
<td>25.10, 0.04</td>
<td>-</td>
</tr>
<tr>
<td>g</td>
<td>27.13, 0.04</td>
<td>26.41, 0.09</td>
<td>26.32, 0.12</td>
<td>26.30, 0.13</td>
<td>27.18, 0.01</td>
<td>26.69, 0.04</td>
<td>26.56, 0.04</td>
<td>26.47, 0.09</td>
</tr>
<tr>
<td>r</td>
<td>27.19, 0.04</td>
<td>26.01, 0.16</td>
<td>25.84, 0.24</td>
<td>26.38, 0.12</td>
<td>27.14, 0.01</td>
<td>26.21, 0.07</td>
<td>26.08, 0.05</td>
<td>26.43, 0.09</td>
</tr>
<tr>
<td>i</td>
<td>26.60, 0.04</td>
<td>25.44, 0.15</td>
<td>25.28, 0.22</td>
<td>25.82, 0.12</td>
<td>26.56, 0.01</td>
<td>25.68, 0.08</td>
<td>25.43, 0.07</td>
<td>25.88, 0.09</td>
</tr>
<tr>
<td>z</td>
<td>25.73, 0.04</td>
<td>24.62, 0.17</td>
<td>24.57, 0.21</td>
<td>24.90, 0.14</td>
<td>25.87, 0.01</td>
<td>25.03, 0.09</td>
<td>24.81, 0.05</td>
<td>25.16, 0.10</td>
</tr>
<tr>
<td>y</td>
<td>24.92, 0.04</td>
<td>23.81, 0.16</td>
<td>23.72, 0.21</td>
<td>-</td>
<td>24.92, 0.02</td>
<td>24.01, 0.09</td>
<td>23.88, 0.06</td>
<td>-</td>
</tr>
</tbody>
</table>

The Modified Feature-based scheduler closely matches the footprint of the official survey, and in addition outperforms opsim V4 in terms of the uniformity of the coverage, with lower variances, specially in WFD and SCP regions.

Figure 2.8: Ratio of the pairs (in the $g$, $r$, and $i$ filters) to the total number of the observations on the sky map. For the areas to which the pair constraint is applied, the ratio is desired to be 1. For the Modified Feature-based scheduler (right), the pair constraint is applied to all of the regions, whereas for opsim V4 (left), it is applied to the WFD and NES regions only.

of visits collectively in all filters for opsim V3, opsim V4, and the Modified Feature-based scheduler. Note that the adjustability of the Feature-based scheduler allows for a significant change in the behavior of the telescope; in this case, to encourage observations around the meridian we defined a new basis function in such a way that
Figure 2.9: Distribution of the pair ratio to the total number of the observations, in the $g$, $r$, and $i$ filters. The Feature-based scheduler covers a large area of the sky by a successful pair visit (higher median), whereas opsim V4, maintains a uniform ratio of pair visit for a larger area of the sky (lower variance).

the Modified Feature-based scheduler prefers to observe a contiguous set of fields that is then reobserved later in the same order.

### 2.5.4 Signal-to-noise Ratio

For a multiobjective survey telescope, such as LSST, comparing the overall performance of the different schedules is a difficult task, particularly because of the large number of competing factors that are involved in the evaluation of their performance. In some cases these criteria are not even objective or well-defined, such as the importance of an area of astronomy compared to the rest. Nevertheless, we conclude this section with a general comparison of the performance using the overall signal-to-noise ratio of the surveys. Table 2.6 reflects the value of median throughput for three different schedulers in $r$ and $g$ bands. The modified Feature-based scheduler significantly outperforms the other two.
Figure 2.10: Each plot is the distribution of the visits on an altitude-azimuth sky map in one of the six filters. The two left columns belong to opsim V4, and the two right columns belong to the Modified Feature-based scheduler. The higher concentration on the meridian (vertical axis) for the Modified Feature-based scheduler shows a more desirable behavior. Moreover, consistent concentration of the visits on the east wing can potentially provide a better success rate in pair observations.

The throughput is mainly determined by the combination of open shutter fraction (OSF) and air-mass. The OSF is the total time that the telescope camera shutter was open divided by the maximal time it could have been open. This reveals how time-efficiently the observations have been scheduled. The median air-mass reflects the overall quality of the collected data. As mentioned before, observations in lower air-mass allow for higher data quality. Comparing the values of the OSF and air-mass for both of the baseline schedules, opsim V3 and opsim V4, shows that there is
Table 2.6: Comparison of the Schedulers in a Section of the LSST Main Survey Area; WFD

<table>
<thead>
<tr>
<th>Survey</th>
<th>median throughput $r$ (%)</th>
<th>median throughput $g$ (%)</th>
<th>OSF</th>
<th>median Airmass</th>
<th>dithered</th>
</tr>
</thead>
<tbody>
<tr>
<td>modified Feature-Based</td>
<td>63.7</td>
<td>47.0</td>
<td>0.705</td>
<td>1.1</td>
<td>yes</td>
</tr>
<tr>
<td>opsim V3</td>
<td>55.3</td>
<td>40.0</td>
<td>0.736</td>
<td>1.2</td>
<td>no</td>
</tr>
<tr>
<td>opsim V4</td>
<td>54.4</td>
<td>40.8</td>
<td>0.715</td>
<td>1.1</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 2.11: Distribution of the visits on an altitude-azimuth sky map for three schedulers. On the left, opsim V3 schedules most observations at high air-mass and very few on the meridian. In the middle, opsim V4 schedules many observations on the meridian but still executes deep drilling fields and a smattering of other observations at high air-mass. On the right, the Modified Feature-based scheduler concentrates observations around the meridian, including deep drilling observation.

a trade-off between the two values. While opsim V4 offers better median air-mass, its OSF is worse. However, its median throughput is very close to that of opsim V3. This comparison reveals that the change of meta-parameters, as well as tweaking the structures of proposal-based schedulers such as opsim V3 and opsim V4, only changes the balance of trade-off between OSF and air-mass, but not the actual performance of the scheduling.

44
2.6 Discussion

This study demonstrated that a Markovian scheduler with expert-designed features and a parameterized linear decision-making policy can be successfully applied to multi-mission, ground-based instruments such as LSST. Unlike the mainstream telescope schedulers, the Feature-based scheduler does not rely on handcrafted observation proposals. Instead, bringing the decision-making process to the individual observation level improves the efficiency of the telescope’s operation.

In particular, while proposals are designed for feasibility, our approach is designed for optimality in addition to feasibility. In general, automatic schedulers such as the Feature-based scheduler are fundamentally less prone to suboptimality compared to the schedulers that heavily rely on human interactions. The source of such suboptimality is the manual tailoring that is performed based on the inspections of the instances. Moreover, adjusting the behavior of human-dependent schedulers is inconvenient and time-consuming in practice.

Furthermore, being modeled as an MDP, the Feature-based scheduler is a systematic approach to operating the instruments under uncertainties and interruptions.

On the other hand, the building blocks of the Feature-based scheduler are designed modularly in an intuitive way for the astronomy community. This property allows for expert interventions if needed. However, the robust structure and implementation of the scheduler allow regulated interventions only. In particular, valid interventions include defining new features, new basis functions, and adjusting the design parameters, none of which break the measurability, linearity, and memorylessness of the scheduler. As a result, all of the desirable properties, such as simplicity, optimality conditions, and modularity of the design, remain valid.

In addition, due to the coherent structure, from training to online decision-making, the Feature-based scheduler is easy to understand, implement, and troubleshoot. Simplicity of the design and implementation also provides a user-friendly environment
for a wide range of programming expertise to define a custom mission objective, train a scheduler, and examine the behavior of the scheduler with various mission objectives. Similarly, in a particular project, when a change in the mission’s objective is necessary, deriving a new scheduler that optimizes the new objective is principally automated.

Moreover, for the mission-planning stage of a future instrument, a scheduler with an adjustable objective can be extremely helpful, because it can answer the high-level trade-off questions, such as the time efficiency of the different proposed strategies.

Computationally, the optimization part of the scheduler is the most demanding task. Once the optimal values of the design parameters are known, one night of observation can be simulated in 2-3 minutes. Equivalently, in real-time operations, decisions are made on the fly about 900 times faster than what is needed. For the optimization part, however, the necessary computational time and resources vary. If many different objective functions are being tested for planning a mission, then a quick eDE optimization, which takes a few hours, can find a sufficiently good scheduler for each mission. Even a quick manual hand tuning that reflects the intuitive importance of each basis function is possible, because they carry an astronomical meaning. On the other hand, if the objective is known and the scheduler is being trained for real-time decision-making, then one might even categorize the observation nights based on their main differences, such as the moon phase, seasonal variations, and weather patterns, and then train a scheduler specifically for each category to further increase the efficiency of the operation.
Chapter 3

Fourier-analytic Optimization

3.1 Introduction and Preliminaries

3.1.1 An Overview

Optimization problems with Fourier-analytic structures naturally appear in many engineering problems such as signal processing, antenna array design, and optical instrumentation. They also appear in the formulation of some fundamental problems in pure mathematics such as the Turán Extremal Problem (TEP), the Packing Density Upper-Bound (PDUB), and the Beurling-Selberg box minorant problem.

Optimization problems that contain both a function and its Fourier transform in their variables are referred to as Fourier Constrained Optimization (FCO); because the definition of the Fourier transform appears as a set of constraints that relates the variables of optimization.

Traditionally, numerical computation of the FCO problems takes place in engineering applications. However, with the advancement of technology in the computational power and storage, it is possible to numerically compute the solution of some FCO problems that have been only the subject of analytical approaches. Especially for
problems that despite the analytical efforts are remained unsolved, numerical solutions could provide bounds, approximations, evidences, and insights.

We propose a framework to create a computable model of linear FCO problems. We demonstrate the application of our framework on the Turán Extremal Problem (TEP) and Packing Density Upper-Bound (PDUB). TEP and PDUB are linear FCO problems that are motivated by two different mathematical subjects. However, from the optimization point of view the similarity between the two calls for similar computational approaches.

While there are numerous published results on approximating the density of packing, numerical investigations of TEP have not been been very extensive. On the other hand, there is a broader theoretical understanding of TEP. In this dissertation, we will take advantage of the recent advances on each problem to further understand the other problem.

TEP was inspired by a problem in radar engineering at the Jet Propulsion Laboratory in the early seventies [Révész, 2011]. Turán proposed a conjecture for the solution of this problem which we refer to as Turán Conjecture. This conjecture is proven for some special cases of TEP. Section 3.2 presents the formulation of TEP, the Turán Conjecture, and the cases for which the Turán Conjecture is proven. Then we propose a computable model of TEP in Section 3.3 which is followed by numerical results in 3.4.

The problem of packing with convex bodies has long been studied with a special attention to the densest packing of Euclidean space with spheres. This important special case of the problem is formally referred to as sphere packing which asks for a configuration of congruent spheres that fills the highest fraction of the space. The spheres cannot overlap except at their boundaries. in Section 3.5, we provide more background on the origin, current status, and applications of sphere packing as well as its generalized version; packing with convex bodies. Then we present the formulation
of PDUB, which provides upper-bounds on the density of packing convex bodies into Euclidean space. In Section 3.5.3, we provide a computable formulation of PDUB and our numerical results.

A solver called LOQO is used to obtain the numerical results of this chapter. Computer codes that are used for computations are expressed in AMPL modeling language developed by Fourer et al. (1993). Our AMPL models are provided in Appendix C.

3.1.2 Definitions

In what follows, we state the definitions that are used in this chapter. First, we take the definition of the Fourier transform to be as follows:

**Definition 3.1.1.** Let \( f : \mathbb{R}^N \to \mathbb{R} \) be an \( L^1 \) function. Then,

\[
\hat{f}(\xi) := \int_{\mathbb{R}^N} f(x)e^{-2\pi i(x,\xi)} \, dx,
\]

(3.1)

is the Fourier transform of \( f \), where \((x,\xi)\) denotes the inner product of \( x \) and \( \xi \), and \( i = \sqrt{-1} \).

The inverse Fourier transform is given by;

**Definition 3.1.2.**

\[
f(x) = \int_{\mathbb{R}^N} \hat{f}(\xi)e^{2\pi i(x,\xi)} \, dx.
\]

(3.2)

For convenience, we will refer to the Fourier transform of \( f \) by \( \mathcal{F}(f) \), and to the inverse Fourier transform of \( \hat{f} \) by \( \mathcal{F}^{-1}(\hat{f}) \). We also refer to the domain of function as “time” domain and the domain of the its Fourier transform as “frequency” domain.

**Definition 3.1.3.** A convex body is a compact convex set in \( \mathbb{R}^N \) with non-empty interior.
In this manuscript, all convex-bodies will also be assumed to be symmetric with respect to the origin, unless otherwise is stated.

**Definition 3.1.4.** Convolution of two functions \( f : \mathbb{R}^N \rightarrow \mathbb{C} \) and \( g : \mathbb{R}^N \rightarrow \mathbb{C} \) is defined as follows,

\[
(f * g)(x) = \int_{\mathbb{R}^N} f(y)g(x - y)dy.
\]  \hfill (3.3)

**Definition 3.1.5.** Discrete Fourier Transform (DFT) in \( N \) dimensional space is defined as follows;

\[
a_{k_1,k_2,...,k_N} = \sum_{n_N=0}^{P_N-1} \ldots \sum_{n_2=0}^{P_2-1} \sum_{n_1=0}^{P_1-1} z_{n_1,n_2,...,n_N} e^{-2\pi i (k_1,n_2,...,n_N)} e^{\left(\frac{2\pi}{P_1} n_1 \ldots \frac{2\pi}{P_N} n_N\right)},
\]  \hfill (3.4)

where \( z_{n_1,n_2,...,n_N} \) is the “time” domain tensor, and \( a_{k_1,k_2,...,k_N} \) is the “frequency domain” tensor. In general, \( n_j = 1,2,...,P_j \) and \( k_j = 1,2,...,Q_j \). However, usually the discretization is uniform and the domain is origin-symmetric; hence, \( P_1 = P_2 = ... = P_N \) and \( Q_1 = Q_2 = ... = Q_N \).

**Definition 3.1.6.** We refer to the following \( N \)-dimensional set as “\( p \)-unit ball”;

\[
\{(x_1,x_2,...,x_N \in \mathbb{R}^N : |x_1|^p + |x_2|^p + ... + |x_N|^p \leq 1}\}. \hfill (3.5)
\]

### 3.2 Turan’s Extremal Problem

Let \( K \subset \mathbb{R}^N \) be a symmetric convex-body. A symmetric convex body poses a symmetric problem, therefore we restrict our attention to real-valued functions which have real-valued Fourier transforms. In this setting, Turan’s Extremal Problem can
be formulated as follows:

\[
\sup_{f: \mathbb{R}^N \to \mathbb{R}} \int_{\mathbb{R}^N} f(x) dx, \\
\text{subject to } f(0) = 1, \\
\hat{f}(\xi) \geq 0, \text{ for } \xi \in \mathbb{R}^N, \\
f(x) = 0, \text{ for } x \notin K.
\] (3.6)

This problem searches for “extremal” positive-definite functions that are supported on a convex body. The first constraint is a scaling restriction. The second condition simply imposes non-negativity on the Fourier transform which guarantees the positive-definiteness of the feasible functions. Finally, the last condition directly restricts the support of \(f\) to be a convex body. Note that \(\hat{f}(0)\) can be replaced for the objective function, because by the definition of the Fourier transform, 
\[
\int_{\mathbb{R}^N} f(x) dx = \hat{f}(0).
\]

Conjecture 3.2.1 (Turán conjecture). The optimal value of Problem (3.6), denoted by \(\gamma_N(K)\), is

\[
\gamma_N(K) = \frac{\text{vol}(K)}{2^N},
\]

where, \(\text{vol}(K)\) is the volume of convex body \(K\).

Furthermore, the following function is the optimal solution of TEP,

\[
f^* = \frac{1}{2} \mathcal{I}_{K/2} * \mathcal{I}_{K/2},
\]

where \(K/2\) is the convex body resulting from downscaling of \(K\) by a factor of 2, and \(\mathcal{I}_{K/2}\) is the characteristic function of \(K/2\). Characteristic function of a set, \(\mathcal{I}_K(x)\), is equal to one for \(x \in K\), and zero otherwise. Finally * denotes the convolution operator.
The Turán Conjecture is proven for a set of convex bodies, including hyperellipsoids. It is also shown to be true for convex bodies that tile the space by translation and separately for spectral bodies. Whether the two categories of bodies are equivalent is an open question, which is conjectured to be true by Fuglede (1974).

Figure 3.1 demonstrates the optimal solution of TEP for two known cases of $K; [-1, 1]$ and $[-1, 1]^2$. The optimal Fourier transform function for $K = [-1, 1]$ is $\text{sinc}^2(x)$ and for $K = [-1, 1]^2$ is $\text{sinc}^2(x)\text{sinc}^2(y)$, where $\text{sinc}(x) = \frac{\sin(x)}{x}$.

The support of the optimal functions are compact, but the support of the Fourier transforms are not. For clarity in Figure 3.1, we chose an arbitrary truncation of the domains in the plots.

![Figure 3.1: Optimal solutions of TEP for two known cases. The two top plots are the solutions for $K = [-1, 1]$, and the two bottom ones are the solutions when $K$ is $[-1, 1]^2$. Note that the solutions are symmetric about the origin, but the two-dimensional figures show one quarter of the solutions for clarity.](image)

**Theorem 3.2.2.** The conjectured optimal value of TEP is a lower-bound on the optimal value of TEP;

$$\gamma_N(K) \geq \frac{\text{vol}(K)}{2^N}.$$  

We provide a proof of the above theorem in Appendix A.2.1.
3.3 An Approximate Formulation of TEP

In the presence of a theoretical lower-bound, which is conjectured to be sharp, our numerical approach addresses a rigorous upper-bound approximation. In this section, we present a formulation to compute these upper-bounds. Then a mathematically equivalent formulation is proposed which computationally is more efficient than using the direct expression of the problem. Finally, we add an extra constraint to TEP in order to narrow down the search space. While the first two methods can be applied to all Fourier Constrained Optimization problems, the application of the last method is limited to problems with certain structures, such as those with compactly supported variables. In Section 3.4, we provide numerical results for some instances of TEP in order to demonstrate the performance of our framework in practice.

3.3.1 TEP Relaxation

Problem (3.6) is an infinite-dimensional Linear Program. Its variables are functions, $f$ and $\hat{f}$, which are indexed by $x \in K$ and $\xi \in \mathbb{R}^N$ respectively. The first step to create a computable formulation of TEP is to discretize the variables and the search space.

In what follows we show that without loss of generality one can assume that $\hat{f}$ is periodic. Periodicity of $\hat{f}$ in addition to the constraint that $f$ is compactly supported implies that $f$ is a vector.

Let $T$ be an integer, $\mathbb{Z}^N$ be the integer lattice, and $T\mathbb{Z}^N$ a scaled integer lattice by $T$. Define $\hat{f}_T(\xi) := \sum_{y \in T\mathbb{Z}^N} \hat{f}(\xi + y)$ to make a periodic Fourier transform. Then
consider the following optimization problem;

\[
\max_{f_T: \mathbb{R}^N \to \mathbb{R}} \widehat{f}_T(0),
\]

subject to \( f_T(0) = 1, \)

\[
\widehat{f}_T(\xi) \geq 0, \quad \text{for } \xi \in \mathbb{R}^N,
\]

\[
f_T(j) = 0, \quad \text{for } j \notin K,
\]

Note that in the case where \( \widehat{f}_T \) is periodic, \( f_T \) is a series of real-valued numbers indexed by \( j \in \mathbb{Z}^N_T. \)

The above periodization scheme poses a relaxation of TEP for which the value of \( f \) samples at the discretization points are equal to \( f_T \) at the same discretization points. The following lemma establishes that the computed values of \( f_T \) via above formulation, coincide with the samples of \( f \) on the discretized domain.

**Lemma 3.3.1.** Provided that \( \widehat{f} \) decays fast enough, \( f(x) \) and \( f_T(j) \), are equal at \( x = j. \)

*Proof.* See Appendix A.2.2.

An application of the above lemma, is to approximate \( f \) by interpolation of the computed samples.

The following proposition establishes an approach to evaluate upper-bounds for the optimal value of TEP in all dimensions and for all origin-symmetric convex bodies.

**Proposition 3.3.1.** Let \( K_u \) be the smallest polytope with vertices in \( \mathbb{Z}^N_T \) that contains \( K \). Substitute \( K \) with \( K_u \) in Problem (3.7). Then;

\[
\gamma^u_N(K) \geq \gamma_N(K),
\]

where \( \gamma^u_N(K) \) is the optimal value of Problem (3.7).

*Proof.* See Appendix A.2.3.
3.3.2 Computational Efficiency

Fourier transform appears as a set of linear equality constraints in the FCO problems. For many instances, the set of Fourier constraints is a major computational burden. In particular, for problems where the Fourier transform operator is defined on higher geometrical dimensions, the complexity of the computations grow exponentially with respect to the geometrical dimension. This is due to the discretization scheme which exponentially increases the number of variables with respect to the geometrical dimension of the problem.

There has been extensive studies on the advancement of the Fast Fourier Transform (FFT) algorithms and related computational tools. An FFT algorithm takes a vector (or a tensor in multi-dimensional settings), $z_n$, as input and efficiently computes its Discrete Fourier Transform (DFT), $a_n$. Direct computation of $a_n$ values via the definition of DFT is an operation of complexity $O(n^N)$, however FFT algorithms use a clever segmentation of the intervals, on which the input data is defined, to reduce the complexity of the transform. Two-dimensional transforms, for instance, are of the complexity of $O(n \log(n))$.

However, to efficiently solve the Fourier Constrained Optimization problems, FFT algorithms cannot be used for the following reasons.

First, while adding a peripheral algorithm to an optimization model is not impossible, it brings complications to the implementation, integrity and compatibility of the software. But more importantly, it defeats the original purpose of substituting the explicit DFT expression by FFT to gain efficiency; because calling an external routine at every iteration of the optimizer is very inefficient.

Secondly, there are edge cases that should be handled differently, such as computing the Fourier transform of the Delta function or its approximations. Using an FFT algorithm that is able to compute a valid DFT for some edge cases might bring hidden non-convexity or discontinuity to the resulting optimization model.
Thirdly, for standard FFT algorithms to work properly, there are restrictions on the number of discretization intervals. For instance, the size of the input, $a_n$, and output, $z_n$, vectors have to be a multiplication of certain numbers, and both are assumed to be equidistance samples of their respective functions in “time” and “frequency” domains. Obviously these algorithm-specific restrictions are irrelevant to the original problem, and may result in suboptimality of the solution, or inefficiency of the computation.

Finally, for problems that take place in higher geometrical dimensions, standard FFT algorithms are not easily generalizable. In other words, for each dimension one has to use a particular FFT algorithm that is suitable and customized for that particular dimension.

Therefore, instead of adopting an external FFT algorithm, we use the approach introduced in Vanderbei (2012) in order to overcome the curse-of-dimensionality brought to the problem by the direct expression of DFT.

First, rearrange the formulation of DFT to obtain separate sums;

$$a_{k_1,\ldots,k_d} = \sum_{n_N=0}^{P_N-1} e^{-2\pi i k_N \frac{n_N}{P_N}} \cdots \sum_{n_1=0}^{P_1-1} e^{-2\pi i k_1 \frac{n_1}{P_1}} z_{n_1,\ldots,n_N}, \quad (3.8)$$

then define the following kernel:

$$K_i(\cdot) := \sum_{n_i=0}^{P_i-1} e^{-2\pi i k_1 \frac{n_1}{P_1}} (\cdot), \quad \text{for } i = 1,\ldots,N. \quad (3.9)$$

With the above notation one has;

$$a_{k_1,\ldots,k_d} = K_N(K_{N-1}(\cdots(K_1(z_{n_1,\ldots,n_N})))), \quad (3.10)$$

Note that $K_i$ is a linear kernel; therefore it can be written as matrix multiplication with properly rearranging $a_{k_1,\ldots,k_d}$ and $z_{n_1,\ldots,n_N}$. The matrix associated with kernel $K_i$
is denoted by \( \tilde{K}_i \) in the following:

\[
\tilde{K} = \tilde{K}_N \tilde{K}_{N-1} \cdots \tilde{K}_1,
\]

(3.11)

where \( \tilde{K} \) is the matrix associated with the N-dimensional DFT. Note that \( K \) is a dense matrix, however each \( \tilde{K}_i \) is a sparse matrix. This decomposition is critical to gain computational efficiency using both Simplex Method and Interior Point solvers. For solvers based on Simplex Method there has been extensive studies to have the algorithm take a small number of simplex pivots for sparse problems (Vanderbei et al., 2016). Similarly, Interior Point solvers which can be interpreted as iterations of solving a System of Linear Equations (SLE) are much more efficient for sparse problems; because there has also been extensive studies to optimize the SLE solvers.

With the above notations, the set of Fourier equality constraints that relates the function and its Fourier transform in the optimization problem can be expressed as \( \hat{F}_D = \tilde{K} F_D \), where \( \hat{F}_D \) and \( F_D \) are the properly rearranged discretized versions of the function and its Fourier transform.

Eventually, what we use in the modeling and the computations is the following:

\[
\begin{align*}
G_1 &= \tilde{K}_1 F_D, \\
G_2 &= \tilde{K}_2 G_1, \\
&\vdots \\
\hat{F}_D &= \tilde{K}_N G_{N-1}
\end{align*}
\]

(3.12)

Note that in the new formulation, which is mathematically equivalent to the original DFT, we have introduced a set of new variables; \( G_1, G_2, \ldots, G_{N-1} \). This formulation increases the intrinsic size of the optimization problem, but the sparse structure yields a much more efficient computational tool. Figure 3.2 compares the computational efficiency of solving the dense and the sparse versions of TEP with convex body 1-unit.
ball in dimension two. The gain in computational efficiency grows as the geometrical
dimension of the problem grows.

Figure 3.2: Computational efficiency gain for two-dimensional TEP. The left plot shows the growth of the number of constraints and variables with the increments of discretization resolution. It is clear that the intrinsic size of the optimization problem is larger in the sparse versions of the problem. The right plot compares the computational time of the dense and the sparse versions of the problem with the growth of the variables.

In addition to adopting the sparse formulation of a Fourier Constrained Optimization problem, we believe that developing a solver with embedded Fourier transform can further improve the efficiency of the problem. Appendix C.3 presents a simple Matlab implementation of such solver for two-dimensional FCO problems. The experiments that are shown in Figure 3.2 are performed with this solver.

3.4 Computational Turán Extremal Problem

The following formulation is used to evaluate the numerical results of this section. In this formulation \( f_D \) and \( \hat{f}_D \) denote the vectors that are substituted for \( f_T \) and \( \hat{f}_T \) in Problem (3.7).
\[
\max_{f_D : B \cap \Delta x \mathbb{Z}^N \to \mathbb{R}} \quad \hat{f}_D(0)
\]

subject to
\[
\begin{align*}
&f_D(0) = 1, \\
&\hat{f}_D(\xi) \geq 0, \quad \text{for } \xi \in C \cap \Delta \xi \mathbb{Z}^N, \\
&f_D(x) = 0, \quad \text{for } x \notin K \cap \Delta x \mathbb{Z}^N, \\
&f_D(x) = \sum_{\xi \in C \cap \Delta \xi \mathbb{Z}^N} \hat{f}_D(\xi) e^{2\pi i x \xi} \Delta \xi, \quad \text{for } x \in B \cap \Delta x \mathbb{Z}^N,
\end{align*}
\]

where \( B \) and \( C \) are truncated “time” and “frequency” domains respectively, and \( \Delta x \) and \( \Delta \xi \) are the “time” and “frequency” resolutions. The last set of constraints are the explicit definition of the discrete inverse Fourier transform.

### 3.4.1 Discretization Resolution

The choices of truncation of the “time” and the “frequency” domains, \( B \) and \( C \), along with the resolutions of the discretization, \( \Delta x \) and \( \Delta \xi \), are not arbitrary. Figure 3.3 shows the numerical solution of a one-dimensional TEP for \( K = [-1, 1] \) with arbitrary choices of discretization and truncation. It demonstrates that the numerical results are not nearly close to the solution of TEP, which is known in dimension one.

To prevent the solver from returning wrong numerical solutions, such as the one shown in Figure 3.3, we use the following rules;

**Proposition 3.4.1.** For Fourier constrained optimization problems, the resolution of the “time” domain, \( \Delta x \), and the truncated “frequency” domain, \( C \) must satisfy the following rule;

\[
\Delta x = \frac{1}{\text{vol}(C)},
\]

where \( \text{vol}(C) \) is the volume of \( C \).
Similarly, the resolution of the “frequency” domain, $\Delta \xi$, and the truncated “time” domain, $B$ must satisfy the following rule:

$$\Delta \xi = \frac{1}{\text{vol}(B)},$$

(3.15)

where $\text{vol}(B)$ is the volume of $B$.

We provide a heuristic proof of the above rule in Appendix A.2.4. There is an information-theoretic intuition behind this rule. More specifically, this rule asserts that the information encoded in the function must be the same as the information encoded in its Fourier transform, otherwise the analytic optimization problem is infeasible and the numerical approximations of the problem yield irrelevant results; usually with very large numbers.

Figure 3.4 demonstrates the trend of convergence to the known solution of TEP in dimension one with the above choice of truncation and discretization, whereas the arbitrary choices yield worse results as the discretization becomes finer. We measure the deviation of our solution with the sum of squares of the differences between numerical results and the true values. Then, we divide that some by the number of samples.
3.4.2 Extra Constraints

Looking at the original formulation of TEP, where the integral of $f$ over $K$ is being maximized, one expects $f$ to remain non-negative. Provided that the third constraint requires $f$ to vanish outside of $K$, it is possible that $f$ would vanish on the boundary of $K$ as well. Therefore, adding the following extra constraint does not seem to create major conflict with the analytical structure of TEP;

$$f_D(x) = 0, \text{ for } x \in (\partial K + ball(\Delta x)).$$

(3.16)

In the above equation $(\partial K + ball(\Delta x))$ is the augmented boundary of $K$ and $ball(\Delta x)$ represents a small ball of size $\Delta x$. In other words, we force the values of $f_D$ that are close to the boundary of $K$ to remain zero. In practice, adding this constraint significantly speeds up the computations.

3.4.3 Numerical Results for TEP

For the following numerical results we used the discretization scheme introduced in Section 3.4.1 and added the extra constraint discussed in Section 3.4.2. The computer
model is implemented in AMPL modeling language and it can be found in Appendix C.1.

Figure 3.5 demonstrates two experiments for $K = [-1, 1]^2$. One of which uses a coarse discretization, and the other one with a finer discretization. In the same figure we also compare the empirical solutions with the optimal TEP solutions which are known for this choice of $K$.

![Figure 3.5: Two experiments for the solution of TEP relaxations with $K = [-1, 1]^2$. The two top plots are experiments with coarser discretization compare to the two bottom plots.](image)

For the examples presented in Table 3.1 we take $B$ to be the unit box, $[-1, 1]^N$, and scale $K$ so that it is contained in the unit box. Note that for a positive real number, $\alpha$, that scales $K$ into $\alpha K$, one has; $\gamma_N(\alpha K) = \frac{\gamma_N(K)}{\alpha^N}$.  

62
Table 3.1: Numerical Upper-Bounds for the Turán Extremal Problem

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$C$</th>
<th>Conjecture/Known Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1-unit ball</td>
<td>$[-30,30]^2$</td>
<td>0.5000*</td>
</tr>
<tr>
<td>2</td>
<td>2-unit ball</td>
<td>$[-30,30]^2$</td>
<td>0.7854*</td>
</tr>
<tr>
<td>2</td>
<td>3-unit ball</td>
<td>$[-30,30]^2$</td>
<td>0.8833</td>
</tr>
<tr>
<td>2</td>
<td>4-unit ball</td>
<td>$[-30,30]^2$</td>
<td>0.9270</td>
</tr>
<tr>
<td>2</td>
<td>5-unit ball</td>
<td>$[-30,30]^2$</td>
<td>0.9501</td>
</tr>
<tr>
<td>2</td>
<td>$[-1,1]^2$</td>
<td>$[-31,31]^2$</td>
<td>1.0000*</td>
</tr>
<tr>
<td>3</td>
<td>1-unit ball</td>
<td>$[-9,0,9,0]^3$</td>
<td>0.1667</td>
</tr>
<tr>
<td>3</td>
<td>2-unit ball</td>
<td>$[-5,0,5,0]^3$</td>
<td>0.5236*</td>
</tr>
<tr>
<td>3</td>
<td>3-unit ball</td>
<td>$[-5,5,5,5]^3$</td>
<td>0.7121</td>
</tr>
<tr>
<td>3</td>
<td>4-unit ball</td>
<td>$[-7,5,7,5]^3$</td>
<td>0.8102</td>
</tr>
<tr>
<td>3</td>
<td>5-unit ball</td>
<td>$[-8,5,8,5]^3$</td>
<td>0.8663</td>
</tr>
<tr>
<td>4</td>
<td>1-unit ball</td>
<td>$[-7,0,7,0]^4$</td>
<td>0.04167</td>
</tr>
<tr>
<td>4</td>
<td>2-unit ball</td>
<td>$[-6,0,6,0]^4$</td>
<td>0.3084*</td>
</tr>
<tr>
<td>4</td>
<td>3-unit ball</td>
<td>$[-6,0,6,0]^4$</td>
<td>0.5341</td>
</tr>
<tr>
<td>4</td>
<td>4-unit ball</td>
<td>$[-6,0,6,0]^4$</td>
<td>0.6750</td>
</tr>
<tr>
<td>4</td>
<td>5-unit ball</td>
<td>$[-5,0,5,0]^4$</td>
<td>0.7631</td>
</tr>
</tbody>
</table>

*Known values

Figure 3.6 demonstrates the convergence of the upper-bounds for some convex bodies as the size of $C$ and along with it the resolution of the discretization in the “frequency” domain increases.

The computational parameters including the number of variables, number of constraints and the timing is presented in Table 3.2. The last column that is labeled with $\sigma_K$ is a measure of computational difficulty of solving the problem for convex body $K$. We define $\sigma_K$ to be $\frac{\text{time(sec.)}}{\text{number of variables}+\text{number of constraints}}$ normalized by its value for 1-unit ball of their corresponding dimension.

### 3.5 Packing Density Upper-Bound

In this section, we discuss a Fourier-analytic Linear Programming that provides an upper-bound for the maximal density of packing with convex bodies. The goal of the packing problem is to find a configuration of congruent convex bodies in the Euclidean
Figure 3.6: Convergence of the upper-bound with the growth of the discretization resolution for TEP in different dimensions and with different convex bodies.

space such that the density is maximized. We define the density of packing to be the fraction of the space that is covered by the convex bodies.

Formalization of the packing problem dates back to 17\textsuperscript{th} century when Kepler conjectured that a densest packing of congruent spheres is obtained by the Face
Table 3.2: Turán Extremal Problem Computations

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>Number of variables</th>
<th>Number of constraints</th>
<th>Time (sec.)</th>
<th>$\sigma_K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1-unit ball</td>
<td>206643</td>
<td>204814</td>
<td>5.1e4</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>2-unit ball</td>
<td>206643</td>
<td>203765</td>
<td>2.2e4</td>
<td>4.32</td>
</tr>
<tr>
<td>2</td>
<td>3-unit ball</td>
<td>206643</td>
<td>203418</td>
<td>3.4e4</td>
<td>6.69</td>
</tr>
<tr>
<td>2</td>
<td>4-unit ball</td>
<td>206643</td>
<td>203267</td>
<td>4.9e4</td>
<td>9.64</td>
</tr>
<tr>
<td>2</td>
<td>5-unit ball</td>
<td>206643</td>
<td>203189</td>
<td>4.1e4</td>
<td>8.07</td>
</tr>
<tr>
<td>2</td>
<td>$[-1, 1]^2$</td>
<td>220599</td>
<td>216756</td>
<td>5.2e3</td>
<td>0.96</td>
</tr>
<tr>
<td>3</td>
<td>1-unit ball</td>
<td>4901260</td>
<td>4900063</td>
<td>5.8e4</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>2-unit ball</td>
<td>857004</td>
<td>856366</td>
<td>1.1e3</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>3-unit ball</td>
<td>1136128</td>
<td>1135047</td>
<td>5.4e3</td>
<td>0.41</td>
</tr>
<tr>
<td>3</td>
<td>4-unit ball</td>
<td>2850304</td>
<td>2847311</td>
<td>8.2e4</td>
<td>2.44</td>
</tr>
<tr>
<td>3</td>
<td>5-unit ball</td>
<td>4134880</td>
<td>4130361</td>
<td>3.9e5</td>
<td>8.00</td>
</tr>
<tr>
<td>4</td>
<td>1-unit ball</td>
<td>1838705</td>
<td>1837706</td>
<td>3.6e5</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>2-unit ball</td>
<td>782861</td>
<td>777352</td>
<td>3.2e5</td>
<td>2.10</td>
</tr>
<tr>
<td>4</td>
<td>3-unit ball</td>
<td>782861</td>
<td>772594</td>
<td>4.3e5</td>
<td>2.84</td>
</tr>
<tr>
<td>4</td>
<td>4-unit ball</td>
<td>782861</td>
<td>769573</td>
<td>4.8e5</td>
<td>3.19</td>
</tr>
<tr>
<td>4</td>
<td>5-unit ball</td>
<td>392305</td>
<td>385334</td>
<td>1.4e5</td>
<td>1.86</td>
</tr>
</tbody>
</table>

*Known values

Centered Cubic (FCC) configuration, the most intuitive configuration of identical spheres. The problem remained unsolved until [Hales (1998)](1998), provided a computer-aided proof of the Kepler’s Conjecture. Hale’s controversial proof was originally written in about 300 pages, but eventually with much efforts in order to summarize the proof and to provide more intuitive sketches of the proof (see [Hales and Ferguson (1998)](1998), [Hales (1998)](1998), [Hales (2003)](2003), [Hales et al. (2009)](2009)), the community gradually confirmed that the sphere packing problem has been solved, see [Sloane (1998)](1998) for instance.

The packing problem is difficult even for congruent circles. Lagrange showed that the Hexagonal Packing is the densest lattice packing, Gauss showed that the same configuration is the densest periodic packing, and eventually Thue in 1890 proved that the Hexagonal Packing is the optimal configuration in general. After 1998, one of the most significant developments on the packing problem was provided by [Cohn and Elkies (2003)](2003).
Based on the earlier works of Levenshtein et al., Cohn and Elkies (2003) provided a Linear Programming formulation to calculate an upper-bound on the density of packing with convex bodies. Moreover, they conjectured that the bound is sharp for sphere packing in eight and twenty-four dimensions, with which one could solve the problem in those two charming dimensions.

Later, Viazovska (2017) solved the problem in 8-dimensional space, using the very same Linear Programming. She showed that the optimal configuration is the root lattice. A few month later, Viazovska in collaboration with Cohn et al. (2017) solved the 24-dimensional problem and showed that the optimal configuration is the Leech lattice (see Conway and Sloane (1989) for more details on these packings). The problem of sphere packing is still open for dimensions other than one, two, three, eight, and twenty-four.

In this section, we use the aforementioned Linear Programming formulation to provide numerical approximations for the Packing Density Upper-Bound (PDUB) with congruent convex bodies. We also discuss the insights that this computational approach can provide about packing configurations.

### 3.5.1 Linear Programming Formulation of PDUB

Cohn and Elkies (2003) showed that the optimal solution of the following Linear Programming provides an upper-bound for the density of packing with convex body $K \in \mathbb{R}^N$.

\[
\inf_{f : \mathbb{R}^N \to \mathbb{R}} \hat{f}(0) \quad \text{subject to} \quad \begin{align*}
\hat{f}(0) &= 1 \\
 f(x) &\geq 0, \text{ for } x \in \mathbb{R}^N \\
 \hat{f}(\xi) &\leq 0, \text{ for } \xi \notin K.
\end{align*}
\]  

(3.17)
Let the optimal value of the above optimization problem be $\delta_N(K)$, and the density of packing with $K$ in dimension $N$ be $\Delta_N(K)$ then;

$$\Delta_N(K) \leq \delta_N(K) \frac{\text{vol}(K)}{2^N}.$$ 

Notice that in order to find an upper-bound, finding any feasible function for Problem (3.17) suffices, but obviously the optimal value provides the best LP-bound. For convenience, we refer to the upper-bounds that are provided by the above formulation as LP-bound.

The combinatorial nature of this problem stems from the constraint that the convex bodies cannot overlap. To illustrate, consider the problem of packing with only two bodies. This constraint poses no penalty for infinitely many configurations where the two bodies are sufficiently far apart, but when the boundaries of the bodies overlap, any arbitrary small change in the distance of the bodies can suddenly make the problem infeasible. The Fourier-analytic version of this problem however, uses a clever approach to express this combinatorial problem as a continuous Linear Programming. In Section 3.5.4, we show how to exploit the properties of Fourier decomposition to formulate some combinatorial optimization problems in geometry in order to obtain a continuous relaxations of such problems.

### 3.5.2 Packing Configurations

It is discussed by Cohn and Elkies (2003), that the optimal configuration can be extracted by the optimal solution of Problem (3.17), denoted by $f^*$. This is possible only if $\Delta_N(K) = \delta_N(K) \frac{\text{vol}(K)}{2^N}$. This condition has been shown to be true for sphere packing in one, eight, and twenty-four dimensions.

For those cases of the packing problem that the above condition holds, the next step is to extract the points at which $f(x)$ vanishes. Zeros of $f$ are the optimal
configuration. In what follows we present an illustrative example for the (admittedly trivial) problem of packing with squares. We chose this problem because the LP-bound is sharp; hence a configuration can be inferred by $f^\ast$.

For a computationally tractable example, consider a large square in lieu of 2-dimensional Euclidean space. The goal is to find a feasible configuration of congruent copies of a smaller square that maximally packs the large square. We numerically solved this problem via proper discretization and truncation of the original LP, which led to the optimal density, 1.

The zeros of our optimal solution is shown in Figure 3.7. In order to provide a better understanding of how the solution looks like we plot all of the zeros before smoothing and interpolating. However, retrieving the optimal function via interpolation of the samples leaves only some of the initial zeros to remain zero (with some tolerance). We marked the final zeros with black crosses in Figure 3.7 which is exactly the optimal configuration for square packing.

The above result, even in such a simple setup, is promising. First, because it is extracted from a continuous linear programming where the model and the solver are both completely oblivious to the combinatorial and geometrical origin of the problem. Secondly, because the perfect configuration is extracted from a truncated and discretized version of an infinite-dimensional LP.

### 3.5.3 Numerical Results For the Packing Density Upper-Bounds

It is discussed earlier that any feasible solution of Problem (3.17) can provide an upper-bound on the density of packing with convex body $K$. In this section we present our numerical results for the density of packing with some convex bodies. Amongst them are circle and sphere for which the density is known, but provide a benchmark
Figure 3.7: Packing of $[-5.5,5.5]^2$ with non-overlapping unit squares. This (admittedly trivial) packing problem is solved through the Fourier-analytic Linear Programming formulation where the model and the solver are both completely oblivious to the combinatorial origin of the problem. The optimal configuration, marked by black crosses, is given by where the optimal solution of Problem 3.17 vanishes.

for our method. We also provide the numerical results of different experiments with different discretization resolution.

The following formulation is used to obtain the numerical results, where $f_D$ and $\hat{f}_D$ are the discretized and truncated versions of $f_D$ and $\hat{f}_D$ respectively.

\[
\begin{align*}
\min_{f_D: B \cap \Delta x \mathbb{Z}^N \to \mathbb{R}} & \quad \hat{f}_D(0) \\
\text{subject to} & \quad f_D(0) = 1, \\
& \quad f_D(x) \geq 0, \quad \text{for } x \in B \cap \Delta x \mathbb{Z}^N, \\
& \quad \hat{f}_D(\xi) \leq 0, \quad \text{for } \xi \notin K \cap \Delta \xi \mathbb{Z}^N, \\
& \quad \hat{f}_D(\xi) = \sum_{x \in B \cap \Delta x \mathbb{Z}^N} f_D(x) e^{-2\pi i x \xi} \Delta x, \quad \text{for } \xi \in C \cap \Delta \xi \mathbb{Z}^N,
\end{align*}
\]
where $B$ and $C$ are truncated “time” and “frequency” domains respectively, and $\Delta x$ and $\Delta \xi$ are the “time” and “frequency” domain resolutions. The last constraint is the definition of Fourier transform that appears as a finite set of linear equality constraints. Note that in our computer model, we used the sparse version of the Fourier transform in order to gain computational efficiency. We discussed the details of this approach in Section 3.3.2.

The choice of discretization to solve Problem (3.18) follows the rule that is proposed in Proposition 3.4.1. We also add the following boundary constraint to gain computational efficiency so that one could compute the upper-bound with higher resolutions. This is similar to the extra constraint that we added to TEP;

$$\hat{f}_D(\xi) = 0, \text{ for } \xi \in (\partial K + \text{ball}(\Delta \xi)),$$

where $\text{ball}(\Delta \xi)$ is a small ball of size $\Delta \xi$ and $(\partial K + \text{ball}(\Delta \xi))$ is the augmented boundary of $K$. This constraint forces the samples of $\hat{f}$ that are closest to the Boundary of $K$ to be zero. We argue that this constraint is a natural choice, because $\hat{f}$ is being maximized at the origin, and it must become negative for $\xi$ values outside of $K$, therefore this constraint agrees with the natural behavior of the solution.

Note that the above condition further constrains the problem which does not change the fact that we are still evaluating an upper bound; because the resulting optimal solution will still be feasible for the original problem.

The numerical results that are presented in what follows, are obtained with the above settings and the AMPL model that is provided in Appendix C.

The numerical results presented in Table 3.3 are not guaranteed to be lower-bounds on the analytical version of PDUB. We use these results as an estimation of the upper-bound. However, comparing the first two rows that we used to benchmark our framework, shows that the error between the results and the theoretical upper-bounds is not significant. These results are promising, especially when it comes to
Table 3.3: Numerical results for the Packing Density Upper-Bound

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$B$</th>
<th>$C$</th>
<th>Best Known LP-bound*</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2-unit ball</td>
<td>$[-16.5, 16.5]^2$</td>
<td>$[-5.0, 5.0]^2$</td>
<td>0.907 (Cohn and Elkies 2003)</td>
<td>0.902</td>
</tr>
<tr>
<td>3</td>
<td>2-unit ball</td>
<td>$[-4.0, 4.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>0.778 (Cohn and Elkies 2003)</td>
<td>0.770</td>
</tr>
<tr>
<td>2</td>
<td>3-unit ball</td>
<td>$[-15, 15]^2$</td>
<td>$[-5.0, 5.0]^2$</td>
<td>-</td>
<td>0.920</td>
</tr>
<tr>
<td>2</td>
<td>4-unit ball</td>
<td>$[-10, 10]^2$</td>
<td>$[-5.0, 5.0]^2$</td>
<td>-</td>
<td>0.901</td>
</tr>
<tr>
<td>2</td>
<td>5-unit ball</td>
<td>$[-12, 12]^2$</td>
<td>$[-5.0, 5.0]^2$</td>
<td>-</td>
<td>0.962</td>
</tr>
<tr>
<td>3</td>
<td>1-unit ball</td>
<td>$[-4.0, 4.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>0.973 (Dostert et al. 2017)</td>
<td>0.951</td>
</tr>
<tr>
<td>3</td>
<td>3-unit ball</td>
<td>$[-3.0, 3.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>0.824 (Dostert et al. 2017)</td>
<td>0.766</td>
</tr>
<tr>
<td>3</td>
<td>4-unit ball</td>
<td>$[-4.0, 4.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>0.874 (Dostert et al. 2017)</td>
<td>0.841</td>
</tr>
<tr>
<td>3</td>
<td>5-unit ball</td>
<td>$[-4.0, 4.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>0.922 (Dostert et al. 2017)</td>
<td>0.868</td>
</tr>
<tr>
<td>2</td>
<td>Fig. 3.8 [left]</td>
<td>$[-10, 10]^2$</td>
<td>$[-7.5, 7.5]^2$</td>
<td>-</td>
<td>0.886</td>
</tr>
<tr>
<td>2</td>
<td>Fig. 3.8 [right]</td>
<td>$[-10, 10]^2$</td>
<td>$[-7.5, 7.5]^2$</td>
<td>-</td>
<td>0.864</td>
</tr>
<tr>
<td>3</td>
<td>Molecule**</td>
<td>$[-4.0, 4.0]^3$</td>
<td>$[-3.5, 3.5]^3$</td>
<td>-</td>
<td>0.654</td>
</tr>
</tbody>
</table>

*To the best of our knowledge.

**Rotation of Figure 3.8 [right] around its horizontal axis.

estimation of the density upper-bounds for arbitrary convex bodies, for which there are no theoretical results.

Also, notice that providing computational lower-bounds on the density of packing with arbitrary convex bodies is a more straightforward task; because any feasible configuration can provide a lower-bound. Such configurations can be obtained via computer simulations, or even practical experiments with physical objects.

Figure 3.8: Arbitrary convex bodies. Our framework can provide density upper-bound estimations without exploiting the special structure of a convex body.
Figure 3.9: Convergence of the density upper-bound estimations as the size of the “time” domain, $B$, and consequently the discretization resolution grow.

Figure 3.9 demonstrates the improvement of our LP-bound estimations as $B$ grows, and consequently as the resolution of the “frequency domain” increases (see Proposition 3.4.1).
Table 3.4: Packing Density Upper-Bound Computations

<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>Number of variables</th>
<th>Number of constraints</th>
<th>time (sec.)</th>
<th>$\sigma_K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2-unit ball</td>
<td>192063</td>
<td>191180</td>
<td>$5.0e^5$</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>2-unit ball</td>
<td>351740</td>
<td>351515</td>
<td>$2.7e^5$</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>3-unit ball</td>
<td>158853</td>
<td>158036</td>
<td>$2.4e^6$</td>
<td>0.58</td>
</tr>
<tr>
<td>2</td>
<td>4-unit ball</td>
<td>79609</td>
<td>79166</td>
<td>$3.1e^4$</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>5-unit ball</td>
<td>101883</td>
<td>101325</td>
<td>$7.3e^4$</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>1-unit ball</td>
<td>351740</td>
<td>351676</td>
<td>$3.6e^5$</td>
<td>1.33</td>
</tr>
<tr>
<td>3</td>
<td>3-unit ball</td>
<td>151645</td>
<td>151509</td>
<td>$3.1e^4$</td>
<td>0.27</td>
</tr>
<tr>
<td>3</td>
<td>4-unit ball</td>
<td>351740</td>
<td>351380</td>
<td>$2.9e^5$</td>
<td>1.07</td>
</tr>
<tr>
<td>3</td>
<td>5-unit ball</td>
<td>97556</td>
<td>97215</td>
<td>$3.6e^4$</td>
<td>0.48</td>
</tr>
<tr>
<td>2</td>
<td>Figure 3.8 [left]</td>
<td>68403</td>
<td>68231</td>
<td>$1.7e^4$</td>
<td>0.10</td>
</tr>
<tr>
<td>2</td>
<td>Figure 3.8 [right]</td>
<td>68403</td>
<td>68332</td>
<td>$2.4e^4$</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>Molecule-like*</td>
<td>351740</td>
<td>351594</td>
<td>$2.6e^5$</td>
<td>0.96</td>
</tr>
</tbody>
</table>

*Rotation of Figure 3.8 [right] around its horizontal axis.

Finally, Table 3.4 shows the size of the optimization problem in terms of the number of variables and the number of constraints in addition to the computational timing of the examples we presented in this section. To reflect the difficulty of the computations for each convex body, we also evaluate $\sigma_K$, that is defined as

$$\frac{\text{time (sec.)}}{\text{number of variables + number of constraints}}$$

ormalized by its value for 2-unit ball of the dimension of $K$.

### 3.5.4 Non-linear Fourier-analytic Formulation of some Combinatorial Problems in Geometry

In this chapter we introduced a framework for solving linear Fourier-analytic optimization problems. In this section we discuss how some combinatorial problems can be formulated as non-linear Fourier-analytic problems. We also explain the benefits of such formulation in order to provide more motivations for the non-linear generalization of our framework in future studies.

Consider the problem of packing. Here we provide a direct non-linear Fourier-analytic formulation for finding an optimal configuration. Recall that the linear
programming formulation yields an optimal configuration only if the resulting bound is known to be sharp.

Once again, consider a compact set $B \subset \mathbb{R}^N$. The objective is to find a feasible configuration of size $m$ that maximally packs $B$ with congruent copies a convex body, denoted by $K \subset \mathbb{R}^N$. The combinatorial constraint is that the copies do not overlap. Define $I_K(x)$ to be the characteristic function of $K$. Then the packing coverage of a given configuration, $(y_1, \ldots, y_m)$, is defined as $F(x) := \sum_{j=0}^{m} I_K(x + y_j)$. Note that any given $x \in B$ is in at most one of the copies of $K$, therefore requiring a non-overlapping coverage amounts to the constraint of $F(x) \in \{0, 1\}$, which is equivalent to $F(x) \leq 1$ in this setup. This leads to the following optimization problem:

$$\begin{align*}
\text{maximize} \quad & \int_B F(x) dx = \int_B \sum_{j=0}^{m} I_K(x + y_j) dx \\
\text{subject to} \quad & F(x) \leq 1 \iff \sum_{j=0}^{m} I_K(x + y_j) \leq 1.
\end{align*}$$

The objective is highly nonlinear, and is notoriously difficult to solve by standard optimization technologies. Using the following property of Fourier transform:

$$\mathcal{F}(f(\cdot + \Delta x))(\xi) = e^{-2\pi i \Delta x \xi} \hat{f}(\xi), \quad (3.20)$$

the same problem in its Fourier Constrained form can be expressed as followe:

$$\begin{align*}
\text{maximize} \quad & \int_{\mathbb{R}^N} \sum_{j=0}^{m} e^{-2\pi iy_j \xi} \hat{I}_K(\xi) d\xi \\
\text{subject to} \quad & \sum_{j=0}^{m} e^{-2\pi iy_j \xi} \hat{I}_K(\xi) \leq 1.
\end{align*} \quad (3.21)$$

With this formulation, variables, $y_j$, of the problem are decoupled from a highly non-linear function, $I_K$, and now appear as arguments of tractable trigonometric functions. Moreover, the strong non-linearity of $F$ in the $x$ domain is decomposed into
a smooth trigonometric basis, for which the computational technology is advanced and optimized. In addition, This formulation provides an organic way to relax the problem for the interest of computational timing. For instance, eliminating the terms that represent higher “frequencies” yields an easier problem that optimizes a smooth approximation of the original function.

To demonstrate the application of Fourier-analytic formulation of highly non-linear problems in practice, consider the problem of efficient utility allocation which is one of the longest studied problems in Operations Research. Theoretically the combinatorial nature of such problems have caused most of the computational challenges. Furthermore, the need to oversimplify the problem so that it is amenable to theoretical analysis has alienated it from realistic solutions. Fourier-analytic formulation of this problem can be used to overcome some of the difficulties for creating high-fidelity models, and efficient approximations of their solutions.

More specifically, let the total number of providers (hospitals, nurses, Taxi hubs, computational nodes, power plants, etc.) be $N$, and their coverage function, $f(x)$, be the amount of resources provided to location $x$ if the individual provider is located at the origin. Then the cumulative coverage for a given configuration of providers, $(y_1, \ldots, y_N)$, is $F(x) = \sum_{i=0}^{N} f(x + y_i)$. An efficient distribution of resources can be expressed as a given ideal coverage $F_{\text{ideal}}(x)$. For example, in urban planning, $F_{\text{ideal}}$ can be the population density to ensure equal resource-per-capita. The objective is to find a configuration that minimizes $\int (F_{\text{ideal}}(x) - \sum_{i=0}^{N} f(x + y_i))^2 dx$. There could also be context-dependent constraints on the coverage. For instance, when the city does not allow taxi hubs near bus stations, the coverage is constrained; $[F(x) = 0$, for $x \in$ forbidden area]. In human resource time allocation, where $x$ stands for time and $F(x)$ for service provided at time $x$; $[F(x) \leq$ available staff, for $x \in$ holidays].

Such problems involve context-dependent non-linearities in the objectives and constraints. Solving them with standard optimization technologies bears the curse-of-
dimensionality, and implementation difficulties, and results in solutions with poorly-understood global optimality.

On the other hand, an equivalent Fourier-analytic problem is to minimize
\[
\int (\hat{F}_{\text{ideal}}(\xi) - \sum_{i=0}^{N} e^{-2\pi j y_i \xi} \hat{f}(\xi))^2 d\xi,
\]
where the translation of the individual providers is manifested as a multiplicative weights of the Fourier transform of the individual coverage, \( \hat{f} \). The FCO formulation decouples the variables of the problem from the arbitrary non-linearity of \( f(x) \), originated from a real-life form of coverage, and translates them into well-behaved trigonometric non-linearity.

Moreover, having the explicit decomposition of \( f(x) \) in Fourier basis, one can omit the negligible terms determined by the computational power limitations. Hence this approach provides a versatile framework for theoretical investigations as well as the development of efficient computational tools for a broad range of applications.

### 3.6 Concluding Remarks

In this chapter we discussed a class of optimization problems with Fourier-analytic structure. Variables of these problems are functions and the constraints involve the Fourier transform of the variable, hence we refer to them as Fourier Constrained Optimization (FCO) problems. Some important problems in pure mathematics are within this class of optimization problems, including the Turán Extremal Problem (TEP) and The Packing Density Upper-Bound (PDUB) which we discussed thoroughly in this chapter.

FCO problems which contain functions amongst their variables are infinite-dimensional optimization problems. Approximation and estimation of such problems require careful discretization and truncation of the variables and the search space. Particularly, in the Fourier-analytic problems, the discretization and truncation of the “time” domain and the “frequency domain” are not independent. In order to
pose a correct approximation of such problems, information-theoretic aspects must be taken into account. More specifically, the information encoded in the variable and its Fourier transform must be proportional.

From the computational point of view, evaluating the Fourier transform with well-established fast algorithms is not practically efficient, because it requires adding extra peripheral routines to the package of optimization model-solver. To avoid this inefficiency, we introduced a modeling method that evaluates the Fourier transform within the optimization model. With this approach both function and its Fourier transform simultaneously evolve through the iterations of the solver until an optimal value is reached. In addition, we proposed that it would be even more efficient if specialized solvers with embedded transformation modules are developed. In this case the Fourier transform is evaluated at the solver layer rather than the model layer.

Moreover, we presented upper-bound approximations for instances of TEP, and estimations for PDUB. Our results cover two categories of instances. First, instances that the optimal value is theoretically known, and secondly the instances that the optimal value is unknown. The first category is used to compare our results with what is known through other approaches.

Due to the applications of packing in a variety of fields including geometry, communication, cryptography, material sciences, signal processing and optics there has been extensive studies on different instances of the packing problem in different geometrical dimensions. Despite the large volume of the previous studies, what we know about the problem of packing compare to what has remained unknown is infinitesimal. Currently one can find many results that are being published frequently on the optimal configurations of different bodies, their maximal packing densities, and approximations.
One of the most frequent approaches in the literature has been to exploit the special structure of a particular body or a class of bodies. For instance, the best known upper-bounds on the density of sphere packing is evaluated through the exploitation of radial symmetry. Another example is the density of packing with congruent tetrahedra which was previously upper-bounded by $1 - 2.6e^{-25}$ (1 is the trivial density upper-bound) by Gravel et al. (2010). To obtain this upper-bound the authors incorporated clever geometrical intuitions and cuts which is only applicable to tetrahedra. Later, this bound was improved by Dostert et al. (2017), by exploiting a more general structure that is applicable to 3-dimensional bodies with tetrahedral symmetry. Other approaches include, adopting random placing processes, and using practical experiments with physical objects of different shapes.

All of the above-mentioned approaches are limited to the specific properties of the packing bodies. In this dissertation we proposed a new approach and a computational tool to estimate the packing density upper-bounds of all convex bodies in all dimensions. Note that the computations in higher dimensions run into curse of dimensionality. To address this issue, first note that amongst all of the existing tools that we know of, only our framework is based on a linear optimization problem which provides one of the most efficient computational bounds compare to the current tools. Secondly, while the experiments might be limited by computational resources, the framework is general; hence in the future when more powerful computational resources are available, our framework can generate new results.

Furthermore, there has been a significant growth in the number of studies in discrete geometry that explore the area in between the computational and theoretical approaches. In this dissertation we have contributed to such efforts in order to prepare for the emergence of powerful computational resources in the near future which would actually close the existing gap between computations and theory.
In the present, we discussed how our tool can provide accurate estimations, intuitions, and insights to the solutions of TEP and PDUB. In addition, comparing the different difficulties of the computations associated with different bodies shows that evaluating the solutions for some bodies are computationally less demanding than the others. While we have not completely understood how to characterize the computationally demanding bodies, we observed that the bodies with more “round” boundaries require more computational efforts, possibly due to the difficulty of expressing such objects on a discretized domain. On the other hand, evaluating the solutions through theoretical approaches are less difficult for bodies with special symmetries such as ellipsoids. Therefore, in order to obtain a richer understanding of these problems, each of the computational and analytical approaches can provide solutions to the instances that the other one cannot.

Finally, we proposed a non-linear FCO formulation for geometrical optimization problems. Some fundamental problems in theory such as surface covering, and in practice such as efficient facility allocation are under the umbrella of non-linear FCO problems. Based on the proposed framework for linear Fourier-analytic optimization problems in this dissertation, it is possible to develop a computational framework to solve non-linear FCO problems which can be a rich area of investigations for future studies.
Chapter 4

Summary

In the first part of this dissertation we establish a performance evaluation scheme to argue that the scheduling of the modern ground-based astronomical instruments is not efficient with traditional methods. This is mainly due to the fast mechanical features of such instruments. We proposed a Markovian Decision Process to model the complex system of instrument-environment. Based on the proposed model we designed a decision-making framework for the efficient operation of the ground-based astronomical instruments. In addition to our theoretical work, we developed a scheduler for the Large Synoptic Survey Telescope. This scheduler is an algorithm that automatically decides where on the sky the telescope should be pointed to and which filter to use for a particular observation.

In the second part of this dissertation we provided a computational scheme to model and solve Fourier Constrained Optimization (FCO) problems. This class of problems involve the Fourier Transform of their variables in their constraints. Some fundamental problems in geometry and number theory, namely packing density upper-bound and Turán extremal problem are linear FCO’s. In a nutshell, the former problem searches for configurations of congruent non-overlapping convex bodies in Euclidean space, such that the fraction of the space that is covered by the bodies
is maximized. The latter problem searches for extremal functions that are positive-definite. Based on our theoretical results, we developed a computational tool with which estimations of the density upper-bounds and approximations of Turán extremal functions are evaluated. Unlike most of the current approaches, our numerical framework is not analytically limited to the geometrical dimension of the problem neither is it limited to the special geometric symmetries of the solutions.
Appendix A

Mathematical Proofs

A.1 Proofs of Results in Chapter 2

A.1.1 Proof of Proposition 2.2.1

Proof. Consider a function $C_\pi: S \to \mathbb{R}$, defined as follows,

$$C_\pi(x_i) = -\mathbb{E}_\pi\left[ \sum_{i \leq j} \gamma^{i-j} R_{\pi(x)}(X_j, X_{j+1}) \right] x_i$$

$$= -\mathbb{E}_\pi[R_{\pi(x)}(x_i, X_{i+1})|x_i] - \gamma \mathbb{E}_\pi\left[ \sum_{i+1 \leq j} \gamma^{j-(i+1)} R_{\pi(x)}(X_j, X_{j+1}) \right] x_i.$$  

Where $\mathbb{E}_\pi[.]$ is the expectation of the argument inside the brackets, if policy $\pi$ is being used. Then by applying the law of total expectation on the second term,

$$C_\pi(x_i) = -\mathbb{E}_\pi[R_{\pi(x)}(x_i, X_{i+1})|x_i] - \gamma \mathbb{E}_\pi[\mathbb{E}_\pi\left[ \sum_{i+1 \leq j} \gamma^{j-(i+1)} R_{\pi(x)}(X_j, X_{j+1}) \right] X_{i+1}|x_i]$$

$$= -\mathbb{E}_\pi[R_{\pi(x)}(x_i, X_{i+1})|x_i] - \gamma \mathbb{E}_\pi[C_\pi(X_{i+1})|x_i].$$
By assuming a finite state space one can expand the expectation as a finite sum,

$$C_\pi(x_i) = -E_\pi[R_\pi(x_i)(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} \mathbb{P}(x_{i+1}|\pi(x_i), x_i, x_{i-1}, \ldots, x_0)C_\pi(x_{i+1}).$$

Then by the Markov property, probabilities that are conditioned on the whole previous sequence of states can be replaced by probabilities that are conditioned only on the previous state,

$$C_\pi(x_i) = -E_\pi[R_\pi(x_i)(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1})C_\pi(x_{i+1}),$$

where, $P_\pi(x_i)(x_i, x_{i+1})$ is the transition probability from $x_i$ to $x_{i+1}$, under the outcome action of policy $\pi$. Now, let $C^*(x_i) = \min_\pi C_\pi(x_i)$ then,

$$C^*(x_i) = \min_\pi (-E_\pi[R_\pi(x_i)(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1})C_\pi(x_{i+1}))$$

$$= \min_{\pi(x_i), \pi(x_{i+1}), \ldots} (-E_\pi[R_\pi(x_i)(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1})C_\pi(x_{i+1}))$$

$$= \min_{\pi(x_i)} (-E_\pi[R_\pi(x_i)(x_i, X_{i+1})|x_i] - \min_{\pi(x_{i+1}), \pi(x_{i+2}), \ldots} \gamma \sum_{x_{i+1} \in S} P_\pi(x_i)(x_{i+1}, x_i)C_\pi(x_{i+1})).$$

(A.1)

For the next time-step, one can construct a function, $\hat{C}$, such that, $\hat{C}(x_{i+1}) = \min_{\pi(x_{i+1})} \min_\pi C_\pi(x_{i+1})$, then,

$$\sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \ldots\}} C_\pi(x_{i+1}) = \sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1})\hat{C}(x_{i+1})$$

$$\geq \min_\pi \sum_{x_{i+1} \in S} P_\pi(x_i)(x_{i+1}, x_i)C_\pi(x_{i+1}).$$

On the other hand,

$$\sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \ldots\}} C_\pi(x_{i+1}) \leq \min_\pi \sum_{x_{i+1} \in S} P_\pi(x_i)(x_i, x_{i+1})C^*(x_{i+1}).$$
Therefore,

\[ \min_{\pi} \sum_{x_{i+1} \in S} P_{\pi(x_i)}(x_i, x_{i+1})C_\pi(x_{i+1}) = \sum_{x_{i+1} \in S} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\{\pi(x_{i+1}), \pi(x_{i+2}), \ldots\}} C_\pi(x_{i+1}). \]

By substituting the second term of the right hand side of Equation (A.1) with the right hand side of the above equation,

\[ C^*(x_i) = \min_{\pi(x_i)} (-E_\pi[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_{\pi(x_i)}(x_i, x_{i+1}) \min_{\pi} C_\pi(x_{i+1})) \]

\[ = \min_{\pi(x_i)} (-E_\pi[R_{\pi(x_i)}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_{\pi(x_i)}(x_i, x_{i+1})C^*(x_{i+1})). \]

The last equality follows from the definition of \( C^* \), and is in the form of Optimal Bellman Equation, for which a solution exists [Bellman 1957]. Moreover \( C^*(x_0) = \min C_\pi(x_0) \) attains the optimal value by the construction of \( C_\pi(x_0) \) which is equal to 

\[-E[\sum_{\pi(x_i)}(x_i, X_{i+1})|x_0]. \]

Now, given a \( C^* \), an optimal policy can be simply evaluated,

\[ \pi^*(x_i) = \arg\min_{a_i \in A_i} (-E_\pi[R_{a_i}(x_i, X_{i+1})|x_i] - \gamma \sum_{x_{i+1} \in S} P_{a_i}(x_i, x_{i+1})C^*(x_{i+1})) \]

\[ = \arg\min_{a_i \in A_i} (-E_\pi[R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1})|x_i]). \]  

(A.2)

Finally define,

\[ \Phi(X_{i+1}) := R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1}), \]  

(A.3)

and substitute \( R_{a_i}(x_i, X_{i+1}) - \gamma C^*(X_{i+1}) \) with \( \Phi(X_{i+1}) \) in Equation (A.2) to complete the proof.
A.2 Proofs of Results in Chapter 3

A.2.1 Proof of Theorem 3.2.2

Proof. In this proof, first, we restrict the problem with more constraints. Then, we calculate the optimal solution and the optimal value of the restricted maximization problem, $\bar{\gamma}_N(K)$, which provides a lower-bound on the optimal solution of TEP. Next, we show that the calculated lower-bound is, in fact, equal to the Turán Conjecture.

Let $\Omega$ be the feasible set of the TEP;

$$\sup_{f: \mathbb{R}^N \to \mathbb{R}} \int_{\mathbb{R}^N} f(x) dx,$$

subject to

$$f(0) = 1,$$

$$\hat{f}(\xi) \geq 0, \text{ for } \xi \in \mathbb{R}^N$$

$$f(x) = 0, \text{ for } x \notin K,$$

and $\bar{\Omega}$ be the feasible set of the following optimization problem, which we refer to as TEP relaxation,

$$\sup_{f: \mathbb{R}^N \to \mathbb{R}} \int_{\mathbb{R}^N} f(x) dx,$$

subject to

$$f(0) = 1,$$

$$\hat{f}(\xi) = \hat{g}^2(\xi), \text{ for } \xi \in \mathbb{R}^N$$

$$f(x) = 0, \text{ for } x \notin K,$$

$$g(x) = 0, \text{ for } x \notin K/2,$$

where $\hat{g}(x)$ is a real-valued, $L^2$ function, and $\mathcal{F}^{-1}(\hat{g}) = g$. Also, $K/2$ is the convex body resulting from downscaling of convex body $K$ by a factor of 2; then $\bar{\Omega} \subset \Omega$. In what follows we further simplify the TEP relaxation problem.
First note that the second constraint of Problem (A.4) implies the second constraint of TEP. Secondly, by one of the properties of the Fourier transform;

\[ \hat{f}(\xi) = \hat{g}^2(\xi) \iff f(x) = g(x) * g(x), \]

and with the definition of convolution, provided that \( \text{supp}(g) = K/2; \)

\[ \text{supp}(g(x) * g(x)) = \text{supp}(\int_{K/2} g(x)g(\lambda - x)d\lambda) = K. \]

On the other hand;

\[ \int_{K/2} g(x)g(\lambda - x)d\lambda = \int_{\mathbb{R}^N} g(x)g(\lambda - x)d\lambda := f(x). \]

Hence, the third constraint of Problem (A.4) is redundant. Finally, note that by the definition of the Fourier transform the objective function can be replaced by \( \hat{f}(0) \) which is equal to \( \hat{g}^2(0). \)

Now we can write the relaxation of TEP in terms of \( g; \)

\[ \sup_{g: \mathbb{R}^N \to \mathbb{R}} \left( \int_{\mathbb{R}^N} g(x)dx \right)^2, \]

subject to 

\[ \int_{\mathbb{R}^N} g^2(x)dx = 1, \]

\[ g(x) = 0, \text{ for } x \notin K/2, \]

To obtain the objective function use the definition of the Fourier transform. For the first constraint, apply the convolution operator on \( f \) at 0; provided that \( g(x) = g(-x), \) by the definition of convolution \( f(0) = \int_{\mathbb{R}^N} g^2(x)dx. \) Given the compact support of \( g \) and the fact that the optimal \( g \) is positive based on the structure of the
problem, We can further simplify the relaxation of TEP as follows:

\[
\sup_{g: \mathbb{R}^N \to \mathbb{R}} \left( \int_{\mathcal{K}/2} g(x) dx \right),
\]

subject to \( \int_{\mathcal{K}/2} g^2(x) dx = 1, \)

The optimal solution of the above problem is analytically known to be a constant function, we take the value of this function to be \( c \), then;

\[
\int_{\mathcal{K}/2} g^2(x) dx = 1 \Rightarrow c^2 \text{vol}(\mathcal{K}/2) = 1 \Rightarrow c^2 = \frac{1}{\text{vol}(\mathcal{K}/2)}.
\]

Where \( \text{vol}(\mathcal{K}/2) \) is the volume of \( \mathcal{K}/2 \). Therefore the optimal value, \( \bar{\gamma}_N(\mathcal{K}) \), is;

\[
\bar{\gamma}_N(\mathcal{K}) = c^2 \text{vol}^2(\mathcal{K}/2) = \text{vol}(\mathcal{K}/2).
\]

But, \( \text{vol}(\mathcal{K}/2) = \frac{\text{vol}(\mathcal{K})}{2^N} \), which completes the proof.

\[\square\]

### A.2.2 Proof of Lemma 3.3.1

**Proof.** Consider the following definition of inverse Fourier transform for periodic functions,

\[
f_T(j) := \int_{[-T,T]^N} \hat{f}_T(\xi)e^{2\pi i \xi j} d\xi \quad j \in \frac{1}{T} \mathbb{Z}^N.
\]

Substitute the definition of \( \hat{f}_T \);

\[
f_T(j) = \int_{[-T,T]^N} \sum_{y \in T\mathbb{Z}^N} \hat{f}(\xi + y)e^{2\pi i \xi j} d\xi = \int_{-T}^{T} \sum_{y \in T\mathbb{Z}^N} \hat{f}(\xi + y)e^{2\pi i (j+y)} d\xi = \int_{\mathbb{R}^N} \hat{f}(\xi)e^{2\pi i \xi j} d\xi =: f(j),
\]

where the second equality is true because \( y \) is an integer.

\[\square\]
A.2.3 Proof of Proposition 3.3.1

Proof. Recall the periodic formulation of TEP;

\[
\max_{f_T: \mathbb{R}^N \to \mathbb{R}} \hat{f}_T(0),
\]
subject to \( f_T(0) = 1, \)
\[
\hat{f}_T(\xi) \geq 0, \text{ for } \xi \in \mathbb{R}^N, \]
\[
f_T(j) = 0, \text{ for } j \notin K, \]

and let \( \Omega_T \) be its feasible set. First, we show that the feasible set of TEP, \( \Omega \) is contained in \( \Omega_T \). Then we substitute \( K \) by \( K_u \), and denote the feasible set of the new problem with \( \Omega_u \). Provided that \( K \subset K_u \) by definition; \( \Omega_T \subset \Omega_u \). This implies \( \Omega \subset \Omega_u \).

Let \( g \) and \( \hat{g} \) be a feasible solution of TEP, and define \( \hat{g}_T(\xi) := \sum_{y \in T \mathbb{Z}^N} \hat{g}(\xi + y) \).

We show that \( g_T \) is a feasible solution of Problem (A.5).

- \( g(0) = 1 \Rightarrow g_T(0) = 1; \) by applying Lemma 3.3.1 at \( j = 0 \).
- \( g(j) = 0 \Rightarrow g_T(0) = 0; \) by applying Lemma 3.3.1 for \( j \notin K_u \).
- \( \hat{g}_T(\xi) \geq 0 \) is implied by the following:

Under the assumption that \( g \) admits the Poisson Summation one has;

\[
\sum_{j \in \mathbb{Z}^N_T} g(j)e^{-2\pi i \xi j} = \sum_{\eta \in T \mathbb{Z}^N} \hat{g}(\eta - \xi).
\]

Note that the right hand side is positive by feasibility of \( \hat{g} \). Next, use the definition of the Fourier transform for periodic functions along with Lemma 3.3.1 to achieve the
following:

\[ \hat{g}_T(\xi) := \sum_{j \in \mathbb{Z}^N} g_T(j) e^{-2\pi i \xi j} \]

\[ = \sum_{j \in \mathbb{Z}^N} g(j) e^{-2\pi i \xi j} \geq 0. \]

Now we use the definition of the periodic Fourier transform, and the Poisson Summation once again to complete the proof by the following:

\[ \hat{g}_T(0) := \sum_{j \in \mathbb{Z}^N} g_T(j) \]

\[ = \sum_{\eta \in T \mathbb{Z}^N} \hat{g}_T(\eta) \]

\[ = \hat{g}(0) + \sum_{\eta \in T \mathbb{Z}^N / 0} \hat{g}(\eta) \geq \hat{g}(0) =: \gamma_N(K), \]

where the last inequality is implied by the non-negativity of \( \hat{g} \) everywhere. \( \square \)

A.2.4 Proof of Proposition 3.4.1

Proof. To obtain the computational versions of FCO, we substitute the set of Fourier transform constraints, with the following set of equalities:

\[ f_D(x) = \sum_{\xi \in C \cap \Delta \mathbb{Z}^N} \hat{f}_D(\xi) e^{2\pi i x \xi} \Delta \xi, \quad \text{for } x \in B \cap \Delta x \mathbb{Z}^N. \]

Consider one of the above equalities at \( x = 0; \)

\[ f_D(0) = \sum_{\xi \in C \cap \Delta \mathbb{Z}^N} \hat{f}_D(\xi) \Delta \xi, \]

and substitute \( \hat{f}_D(\xi) \) by its definition in terms of \( f_D(x); \)

\[ f_D(0) = \sum_{\xi \in C \cap \Delta \mathbb{Z}^N} \sum_{x \in B \cap \Delta x \mathbb{Z}^N} f_D(x) e^{2\pi i x \xi} \Delta x \Delta \xi. \]
Now, isolate $f_D(0)$ from the right hand side of the equality;

\[
f_D(0) = |C \cap \Delta \xi \mathbb{Z}^N| \Delta \xi \Delta x f_D(0) + \sum_{\xi \in C \cap \Delta \xi \mathbb{Z}^N} \sum_{x \in B \cap \Delta x \mathbb{Z}^N \setminus 0} f_D(x) e^{2\pi i x \xi} \Delta x \Delta \xi,
\]

where $|C \cap \Delta \xi \mathbb{Z}^N|$ is the cardinality of set $C \cap \Delta \xi \mathbb{Z}^N$. With a simple rearrangement, and approximating $|C \cap \Delta \xi \mathbb{Z}^N| \Delta \xi$ with $\text{vol}(C)$, one has;

\[
f_D(0)(1 - \text{vol}(C) \Delta x) = \sum_{\xi \in C \cap \Delta \xi \mathbb{Z}^N} \sum_{x \in B \cap \Delta x \mathbb{Z}^N \setminus 0} f_D(x) e^{2\pi i x \xi} \Delta x \Delta \xi. \tag{A.6}
\]

Now we show that the right hand side of the above equality is zero. Define $g_D(x)$ to be a new discretized function, such that;

\[
g_D(x) := \begin{cases} 
0 & \text{if } x = 0, \\
 f_D(x), & \text{else}.
\end{cases}
\]

Then;

\[
\sum_{\xi \in C \cap \Delta \xi \mathbb{Z}^N} \sum_{x \in B \cap \Delta x \mathbb{Z}^N \setminus 0} f_D(x) e^{2\pi i x \xi} \Delta x \Delta \xi = \sum_{\xi \in C \cap \Delta \xi \mathbb{Z}^N} \sum_{x \in B \cap \Delta x \mathbb{Z}^N} g_D(x) e^{2\pi i x \xi} \Delta x \Delta \xi \\
=: g_D(0) \\
= 0.
\]

Therefore;

\[
f_D(0)(1 - \text{vol}(C) \Delta x) = 0. \tag{A.7}
\]

However, $f_D(0)$ cannot be zero or unbounded, because that would yield trivial solutions in extremal problems. For instance in TEP, we force $f_D(0) = 1$ to prevent trivial solutions. Hence, $(1 - \text{vol}(C) \Delta x) = 0$, implies the following:
\[ \Delta x = \frac{1}{\text{vol}(C)}, \]  
(A.8)

which completes the proof of the first part. The second part can be proven exactly with the same approach. \qed
Appendix B

Scheduling Simulation

With the online version of this dissertation we provide an instance of the Feature-based scheduling. Figure B.1 shows the environment of this simulation.
Figure B.1: The moving circular mark is where the telescope is pointed to at any given moment. Those parts of the sky that are covered during the same night are star-like colored marks, and the color reflects the filter with which the observation is performed. The gray areas are where the sky is bright and generally no observation is allowed, for instance, the halo around the moon. The white areas show where the clouds are based on a realistic model of the cloud covers at the LSST site in Cerro Pachon. Finally, the light-orange areas, which do not change as the time goes by, indicate the four main sky regions (Galactic Plane Region, Universal or Wide Fast Deep, South Celestial Pole, North Ecliptic Spur).
Appendix C

Computer Codes

C.1 AMPL model for Turán Extremal Problem

The following code evaluates the upper- and lower-bound approximations of Turán Extremal Problem for the geometrical dimension of 2, \( N = 2 \), and some unit \( L_p \) balls. The code can easily be adjusted for other geometrical dimensions and other convex bodies.

```
# Main Model

param pi := 4*atan(1);

param N := 2; /* Geometrical dimension */
param B_s; /* size of the truncated frequency domain */
param m; /* resolution in time domain */
param n;/* resolution in frequency domain */

param dx;
param dxi;
```
set Xs;
set Xis;

set B  dimen 2; /* frequency lattice */
set C  dimen 2; /* time lattice */
set C1B1  dimen 2; /* intermediate lattice*/

set K dimen 2; /* convex body approximation */
set K_boundary dimen 2; /* boundary of approximation */
param K_vol; /* convex body volume */
param p;

var f {C}; /* extremal function */
var g11 {C1B1}; /* intermediate function, for fast Fourier transform */
var fh {B} >= 0; /* extremal Fourier transform */

maximize gamma2:
    fh[0,0];

subject to f_zero_one:
    f[0,0] = 1;

subject to out_of_K_zero {(x1,x2) in C: (x1,x2) not in K}:
    f[x1,x2] = 0;

subject to g11_def {(x1,xi2) in C1B1}:
g11[x1,xi2] = 2*sum {xi1 in Xis} fh[xi1,xi2]*cos(2*pi*x1*xi1) *dxi
- sum {xi1 in Xis: xi1 = 0 || xi1 = B_s} fh[xi1,xi2]*cos(2*pi*x1*xi1) *dxi;

subject to f_def {(x1,x2) in C}:
    f[x1,x2] = 2*sum {xi2 in Xis} g11[x1, xi2]*cos(2*pi*x2*xi2) *dxi
    - sum {xi2 in Xis: xi2 = 0 || xi2 = B_s} g11[x1, xi2]*cos(2*pi*x2*xi2) *dxi;

subject to f_at_boundry {(x1,x2) in K_boundary}:  # valid constraint
    f[x1,x2] = 0;

option solver loqo;
option loqo_options "verbose=2 timing=1 epsdiag=1.0e-6 sigfig=5";

############################ instances ################################

let B_s := 10;
let m := 2*B_s;
let n := 2*B_s;

let dx := 1/m;
let dxi := B_s/n;
let \( X_s := \{ k \in \{0..m\} \} \cdot k \cdot dx; \)

let \( X_{is} := \{ j \in \{0..n\} \} \cdot j \cdot dx_i; \)

let \( B := \{ (xi_1, xi_2) \in X_{is} \times X_{is} \} \)

let \( C := \{ (x_1, x_2) \in X_s \times X_s \} \)

let \( C_{1B1} := \{ (x_1, xi_2) \in X_s \times X_{is} \} \)

### different unit lp-balls upper and lower approximations

### l2

let \( p := 2; \)

let \( K_{vol} := \pi; \) /* l2 vol */

/* K is l2 ball (lower approximation)*/

let \( K := \{ (x_1, x_2) \in C: (x_1 + dx)^p + (x_2 + dx)^p \leq 1 \} \)

let \( K_{boundary} := \{ (x_1, x_2) \in C: x_2 = (\max \{ (x_1, y_2) \in K \} y_2) \} \)

solve;

/* K is l2 ball (upper approximation)*/

let \( K := \{ (x_1, x_2) \in C: (x_1 - dx)^p + (x_2 - dx)^p \leq 1 \} \)

let \( K_{boundary} := \{ (x_1, x_2) \in C: x_2 = (\max \{ (x_1, y_2) \in K \} y_2) \} \)

solve;
### l3
let p := 3;
let K_vol := 3.53327; /* l3 vol */

/* K is l3 ball (lower approximation)*/
let K := setof {(x1,x2) in C: (x1+dx)^p + (x2+dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C: x2 = (max {(x1,y2) in K} y2)} (x1,x2);
solve;

/* K is l3 ball (upper approximation)*/
let K := setof {(x1,x2) in C: (x1-dx)^p + (x2-dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C: x2 = (max {(x1,y2) in K} y2)} (x1,x2);
solve;

### l4
let p := 4;
let K_vol := 3.7081493; /* l4 vol */

/* K is l4 ball (lower approximation)*/
let K := setof {(x1,x2) in C: (x1+dx)^p + (x2+dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C: x2 = (max {(x1,y2) in K} y2)} (x1,x2);
solve;

/* K is l4 ball (upper approximation)*/
let K := setof {(x1,x2) in C: (x1-dx)^p + (x2-dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C: x2 = (max {(x1,y2) in K} y2)} (x1,x2);
solve;
let K := setof {(x1,x2) in C:
(x1-dx)^p + (x2-dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C:
x2 = (max {(x1,y2) in K} y2}) (x1,x2);

### l5
let p := 5;
let K_vol := 3.800600; /* l5 vol */

/* K is l5 ball (lower approximation)*/
let K := setof {(x1,x2) in C:
(x1+dx)^p + (x2+dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C:
x2 = (max {(x1,y2) in K} y2}) (x1,x2);
solve;

/* K is l5 ball (upper approximation)*/
let K := setof {(x1,x2) in C:
(x1-dx)^p + (x2-dx)^p <= 1} (x1,x2);
let K_boundary:= setof {(x1,x2) in C:
x2 = (max {(x1,y2) in K} y2}) (x1,x2);
solve;
C.2 AMPL model for Packing Density Upper-Bound

The following code, approximates the Packing Density Upper-Bounds with some 3-dimensional $L_p$ balls, and an arbitrary convex body.

```
############ Main Model ################################

param pi := 4*atan(1);

param N := 3; /* Geometrical dimension */
param C_s; /* size of the truncated frequency space*/
param B_s; /* size of the truncated time space */
param m; /* number of pixels in frequency space */
param n; /* C_s >= 1 otherwise; undersampling */

param K_s; /* size of the convex body*/

param dx;
param dxi;

param x_pix;
param xi_pix;

set Xs;
set Xis;

set B dimen 3; /* truncated time space */
set C2B1 dimen 3; /* intermediate space, for fast fourier transform */
```
set C1B2 dimen 3;
set C dimen 3; /* truncated frequency space */

set K dimen 3;
set K_boundary dimen 3;
param K_vol;
set K_2 dimen 3;
set K_2_boundary dimen 3;
param p;
param obj_par;

param fh_conj{C};
param fh_deviation;

var f {B} >= 0;
var g12 {C1B2};
var g21 {C2B1};
var fh {C};

minimize delta3:
    fh[0,0,0];

subject to f_zero_one:
    f[0,0,0] >= 1;

subject to out_of_K_nonpos {(xi1,xi2,xi3) in C: (xi1,xi2,xi3) not in K}: fh[xi1,xi2,xi3] <= 0;
subject to g21_def {(x1,xi2,xi3) in C2B1}:
  g21[x1,xi2,xi3] = 2*sum {xi1 in Xis} fh[xi1,xi2,xi3]*cos(2*pi*x1*xi1) *dxi
  - sum {xi1 in Xis: xi1=0 || xi1=C_s} fh[xi1,xi2,xi3]*cos(2*pi*x1*xi1) *dxi;

subject to g12_def {(x1,x2,xi3) in C1B2}:
  g12[x1,x2,xi3] = 2*sum {xi2 in Xis} g21[x1,xi2,xi3]*cos(2*pi*x2*xi2)*dxi
  - sum {xi2 in Xis: xi2=0 || xi2=C_s} g21[x1,xi2,xi3]*cos(2*pi*x2*xi2)*dxi;

subject to f_def {(x1,x2,x3) in B}:
  f[x1,x2,x3] = 2*sum {xi3 in Xis} g12[x1,x2,xi3]*cos(2*pi*x3*xi3) *dxi
  - sum {xi3 in Xis: xi3=0 || xi3=C_s} g12[x1,x2,xi3]*cos(2*pi*x3*xi3) *dxi;

# extra constraints
subject to K_boundary_zero {(xi1,xi2,xi3) in C:
  (xi1,xi2,xi3) in K_boundary}: fh[xi1,xi2,xi3] = 0;

option solver loqo;
option loqo_options "verbose=2 timing=1 epsdiag=1.0e-6 sigfig=5";

############################## Instances  ##################################
let $C_s := 3.5$; /* size of the truncated frequency space*/
let $B_s := 5$; /* size of the truncated time space*/
let $m := 2C_s B_s$; /* number of pixels in frequency space*/
let $n := 2m$; /* $C_s >= 1$ otherwise undersampling */

let $K_s := 1$; /* size of the convex body */

let $dx := B_s/n$;
let $dxi := C_s/m$;

let $x_{pix} := dx dx dx$;
let $xi_{pix} := dxi dxi dxi$;

let $Xs := setof {k in 0..n} k*dx$;
let $Xis := setof {j in 0..m} j*dxi$;

let $B := setof {x1 in Xs, x2 in Xs, x3 in Xs} (x1, x2, x3)$;
let $C := setof {xi1 in Xis, xi2 in Xis, xi3 in Xis} (xi1, xi2, xi3)$;

/* intermediate space, for fast fourier transform variable */
let $C2B1 := setof {x1 in Xs, xi2 in Xis, xi3 in Xis} (x1, xi2, xi3)$;
let $C1B2 := setof {x1 in Xs, x2 in Xs, xi3 in Xis} (x1, x2, xi3)$;

/* $K$ is $l$ infinity ball */
let $K := setof {(xi1, xi2, xi3) in C: max(xi1, xi2, xi3) <= K_s}$ (xi1, xi2, xi3);
let K_boundary := setof {(xi1,xi2,xi3) in C:
max(xi1,xi2,xi3) = K_s} (xi1,xi2,xi3);
let K_vol := 2^N * K_s^N;
solve;

/* K is lp ball */
let p := 5; /* for instance; K is l1 ball then p = 1 */
let K := setof {(xi1,xi2,xi3) in C:
xi1^p + xi2^p + xi3^p <= K_s^p} (xi1,xi2,xi3);
let K_boundary := setof {(xi1,xi2,xi3) in C:
xi1^p + xi2^p + xi3^p = K_s^p} (xi1,xi2,xi3);

# uncomment the associated volume
#let K_vol := 4/3 *K_s^N; /* l1 ball volume*/
#let K_vol := 2.94277 *K_s^N; /* l1.5 ball volume*/
#let K_vol := 4*pi/3 *K_s^N; /* l2 ball volume*/
#let K_vol := 5.696583541509 * K_s^N; /* l3 ball volume*/
#let K_vol := 6.481987351786* K_s^N; /* l4 ball volume*/
#let K_vol := 6.930354992554* K_s^N; /* l5 ball volume*/

/* K is arbitrary; elongated molecule-shaped */
let obj_par := 0.5; # radius of the end half spheres
let K := setof {(xi1,xi2,xi3) in C:
xi1<1-obj_par && xi2 <= obj_par && xi3 <= obj_par ||
xi1>= 1-obj_par && xi2^2 + (xi1-(1-obj_par))^2 + xi3^2 <= obj_par^2} (xi1,xi2,xi3);
let K_boundary := setof {(xi1,xi2,xi3) in C:
\[ xi1 < 1 - \text{obj}_\text{par} \land xi2 = \text{obj}_\text{par} \land xi3 = \text{obj}_\text{par} \]
\[ \lor xi1 > 1 - \text{obj}_\text{par} \land xi2^2 + (xi1 - (1 - \text{obj}_\text{par}))^2 + xi3^2 = \text{obj}_\text{par}^2 \]
\[(xi1, xi2, xi3);\]
let \( K_{vol} := \frac{4}{3}\pi \text{obj}_\text{par}^3 + 2(1 - \text{obj}_\text{par})\pi \text{obj}_\text{par}^2; \)
solve;

C.3 Matlab Implementation of the Solver with Embedded Sparse Fourier Transform

%%%%
%script 1 : Mail Module
%%%%
% Interior point algorithm for problems involving Fourier transform operations
function \([x, xhat, xbar, n\_iter] \ldots = K2K1problem(c, chat, u, uhat, l, lhat, K1, K2, percision)\)

%%% INPUTS %%%
n = length(c); % x dimension
[m, p] = size(K2); % m: xhat dimension, p: xbar dimension

%%% INITIALIZATIONS %%%
[x, xhat, xbar, w, what, y1, y2, s, shat, z, zhat, Dxbar, Dy1] = initi_vars(n, m, p);

mu = .1; % complimentarity perturbation
%%% Solver Iterations %%%

n_iter = 0;
terminate = 0;

while ~terminate;
    [x, xbar, xhat, w, what, y1, y2, s, shat, z, zhat, det_A] = ...
    one_iter_sparse(x, xbar, xhat, w, what, y1, y2, s, shat, z, ...
    zhat, c, chat, u, uhat, K1, K2, mu);

    mu = eval_mu(mu);
    terminate = eval_termination(x, xhat, w, what, s, shat, z, ...
    zhat, percision, n_iter, det_A);
    n_iter = n_iter + 1;
end

% check if constraints are satisfied
sanity_check(x, xhat, K2*K1, u, uhat, l, lhat, percision)

%%% %script 2 : One Iteration of the Solver

function [x_new, xbar_new, xhat_new, w_new, what_new, y1_new, ...
    y2_new, s_new, shat_new, z_new, zhat_new, det_A] = ...
    one_iter_sparse(x_old, xbar_old, xhat_old, w_old, what_old, y1_old, ...
    y2_old, s_old, shat_old, z_old, zhat_old, c, chat, u, uhat, K1, K2, mu)
% K = K2K1, where K2 and K1 are sparse and K is dense

%%% Retrive dimensions %%%
n = length(x_old);
m = length(xhat_old);
p = length(xbar_old);

%%% CONVENTIONAL NOTATION %%%
X = diag(x_old); Xhat = diag(xhat_old);
W = diag(w_old); What = diag(what_old);

X_inv = diag(1./x_old); Xhat_inv = diag(1./xhat_old);
W_inv = diag(1./w_old); What_inv = diag(1./what_old);

S = diag(s_old); Shat = diag(shat_old);
Z = diag(z_old); Zhat = diag(zhat_old);

S_inv = diag(1./s_old); Shat_inv = diag(1./shat_old);
Z_inv = diag(1./z_old); Zhat_inv = diag(1./zhat_old);

e = ones(n,1); ehat = ones(m,1);

%%% SECONDARY VALUES %%%
D = X_inv * Z + S * W_inv;
E = Xhat_inv * Zhat + Shat * What_inv;

D_inv = diag(1./((1./x_old) .* z_old + s_old .* (1./w_old)));

\[ E_{\text{inv}} = \text{diag}(1./((1./x_{\text{hat old}}) \cdot z_{\text{hat old}} + shat_{\text{old}} \cdot (1./what_{\text{old}}))); \]

\[ \text{sigma} = c - K1' \cdot y1_{\text{old}} - s_{\text{old}} + \ldots \]
\[ \text{mu} \cdot X_{\text{inv}} \cdot e + S \cdot W_{\text{inv}} \cdot (u - x_{\text{old}}) - \text{mu} \cdot W_{\text{inv}} \cdot e; \]
\[ \text{sigmahat} = chat + y2_{\text{old}} - shat_{\text{old}} + \ldots \]
\[ \text{mu} \cdot Xhat_{\text{inv}} \cdot ehat + Shat \cdot What_{\text{inv}} \cdot (uhat - xhat_{\text{old}}) - \text{mu} \cdot What_{\text{inv}} \cdot ehat; \]

%%% Linear System of Equations %%%
\[
\begin{bmatrix}
D & \text{zeros}(n,p), K1', \text{zeros}(n,m); \\
\text{zeros}(p,n), \text{zeros}(p,p), -\text{eye}(p), K2'; \\
K1, -\text{eye}(p), \text{zeros}(p,p), \text{zeros}(p,m); \\
\text{zeros}(m,n), K2, \text{zeros}(m,p), -E_{\text{inv}}
\end{bmatrix}
\]

\[ \text{det}_A = \text{det}(A); \]
\[ A = \text{sparse}(A); \% \text{creating sparse format of A} \]
\[ b = [\text{sigma}; y1_{\text{old}} - K2' \cdot y2_{\text{old}}; -K1 \cdot x_{\text{old}} + xbar_{\text{old}}; \ldots \]
\[ -K2 \cdot xbar_{\text{old}} + xhat_{\text{old}} + E_{\text{inv}} \cdot \text{sigmahat}]; \]
\[ x = A \backslash b; \]
\[ \%[x,f] = \text{pcg}(A, b) \]

%%% Extract desirables %%% \% order matters
\[ D_{x} = x(1:n); \]
\[ D_{xbar} = x(n+1:n+p); \]
\[ D_{y1} = x(n+p+1:n+p+p); \]
\[ D_{y2} = x(n+p+p+1:end); \]
Dxhat = E_inv *(Dy2 + sigmahat);

Ds = S*W_inv*Dx - S*W_inv*u + ... 
S*W_inv*x_old + mu*W_inv*e;
Dshat = Shat*What_inv*Dxhat - Shat*What_inv*uhat + ... 
Shat*What_inv*xhat_old + mu*What_inv*ehat;

Dz = mu*X_inv*e - z_old - X_inv*Z*Dx;
Dzhat = mu*Xhat_inv*ehat - zhat_old - Xhat_inv*Zhat*Dxhat;
Dw = mu*S_inv*e - w_old - S_inv*W*Ds;
Dwhat = mu*Shat_inv*ehat - what_old - Shat_inv*What*Dshat;

%%% Calculate new variables %%%
%scale factor of increments to assure  
%non-negativity of slack variables  
theta = eval_theta(w_old,what_old,z_old,zhat_old,Dw,Dwhat,Dz,Dzhat);

x_new = x_old + theta*Dx;
xbar_new = xbar_old + theta*Dxbar;
xhat_new = xhat_old + theta*Dxhat;
w_new = w_old + theta*Dw;
what_new = what_old + theta*Dwhat;
y1_new = y1_old + theta*Dy1;
y2_new = y2_old + theta*Dy2;
s_new = s_old + theta*Ds;
shat_new = shat_old + theta*Dshat;
z_new = z_old + theta*Dz;
\[ \text{zhat}_{\text{new}} = \text{zhat}_{\text{old}} + \theta \cdot D\text{zhat}; \]

end

%%%%
%script 3 : Evaluate THETA; scale factor of increments
%%%%

function \( \theta \) = eval_theta(\( w_{\text{old}} \),\( \text{what}_{\text{old}} \),\( z_{\text{old}} \),\( \text{zhat}_{\text{old}} \),\( Dw \),\( D\text{what} \),\( Dz \),\( D\text{zhat} \))

all_slacks = [\( w_{\text{old}} \),\( \text{what}_{\text{old}} \),\( z_{\text{old}} \),\( \text{zhat}_{\text{old}} \)];
all_Dslacks= [\( Dw \),\( D\text{what} \),\( Dz \),\( D\text{zhat} \)];

all_new_slacks = all_slacks + all_Dslacks;
if \( \sum(\text{all_new_slacks} < 0) \) \% at least one negative slack variable
    \([m,I]\) = \text{min}(\text{all_new_slacks});
    \% so the smallest new slack variable is zero:
    \( \theta = -m/\text{all_Dslacks}(I) + 1; \)
    \( \theta = \theta / 2; \% \text{to avoid singularity in the algorithm} \)
else \% all new slack variables are non-neg without any scaling
    \( \theta = 1; \)
end
Bibliography


