Project Topic : MRI Appointment Scheduling Optimization

Name: Riya Aggarwal, 24n0452

Abstract

This study addresses the critical issue of optimizing MRI machine scheduling at TATA Memorial Hospital, a premier cancer care institution in India. The hospital faces high demand for MRI scans, resulting in machines often operating continuously beyond the ideal 20-hour limit, sometimes running 22-23 hours or even around the clock. With only two MRI machines, the hospital must balance operational efficiency with patient accessibility, as delays in scheduling often sometime cause patients to experience waiting times of over 24 hours. The project utilizes historical data on patient scan times, including mean scan durations and their variance, to develop a scheduling strategy aimed at minimizing patient waiting time. We divide daily MRI availability into ten 2-hour slots, seeking an optimal allocation that avoids overloading any slot, as overcrowding increases patient waiting. The scheduling model considers both the machine's operating constraints and the variability in patient arrival and service times, aiming to optimize MRI utilization without compromising patient experience. By formulating this as a scheduling optimization problem, we explore policies that balance MRI slot utilization with patient wait times, effectively reducing idle machine hours and enhancing service delivery. The findings and proposed scheduling strategy provide a data-driven framework for improving MRI scheduling, which has implications for better resource management and patient care in high-demand medical facilities.

1 Introduction

This study examines the complex challenge of scheduling appointments for Magnetic Resonance Imaging (MRI) machines at TATA Memorial Hospital, a leading cancer treatment facility in India. The hospital operates with an exceptionally high patient load, and its MRI machines serve diverse patient needs across departments, performing a wide range of scan types. Due to the high demand and limited MRI resources, appointment scheduling has become a significant issue, often resulting in long patient waiting times and machines operating continuously, frequently beyond the ideal 20-hour daily limit.

MRI scheduling at the hospital involves a two-part decision-making process led by the appointment desk secretary: selecting the scan date and assigning a specific time slot on the chosen day. This study focuses on the latter aspect—time slot selection. The MRI machines are typically scheduled in ten 2-hour slots each day, which ideally operate for a combined 20 hours. However, due to the overwhelming demand, machines often run for 22-23 hours, with some days reaching 24 hours of continuous operation. Despite the extended operation hours, patients can experience prolonged delays, sometimes waiting over 24 hours on the day of their appointment and up to two weeks to secure a scan date.

To address this, our project aims to balance two competing objectives: minimizing patient waiting times and reducing machine idle time to optimize MRI resource utilization. However, several sources of randomness complicate the scheduling process, such as variability in scan times due to patient conditions, doctor recommendations, and machine maintenance. This variability often leads to patients being called in early to ensure machines are not idle, further contributing to long wait times on the day of the scan.

To simplify the model, we observed that patient types can be grouped into three clusters based on the similarity in average scan times, thereby reducing the complexity of scheduling. Our study treats MRI time allocation as a block scheduling problem, where patients are assigned to pre-defined, identical-length slots. This approach must account for both the random arrival sequence of patients for appointments and the uncertainty in scan durations. We propose and evaluate slot allocation policies for managing patient arrivals through simulation and formulate an optimization model that incorporates chance constraints to minimize patient waiting times and capture idle machine hours.

2 Related Work

Marynissen and Demeulemeester[1] provide a comprehensive review of multi-appointment scheduling problems in hospitals, which is a critical aspect of healthcare operations research. Their study examines the scheduling challenges that arise when patients require sequential appointments across various hospital resources, such as diagnostic equipment, treatment rooms, and consultations with specialists. The authors emphasize the need to coordinate multiple departments to minimize patient wait times and optimize the efficiency of hospital resources, as delays can lead to adverse health outcomes. They introduce a classification framework that systematically categorizes existing literature on multi-appointment scheduling in hospitals (MASPH) and outlines various methodologies, objective functions, and critical decision points for improving scheduling processes. Their findings underscore the value of a centralized scheduling system, which can alleviate inter-departmental bottlenecks, thereby improving both patient satisfaction and clinical outcomes. This review is especially beneficial for researchers and healthcare practitioners as it identifies unexplored areas in the literature and provides a roadmap for refining hospital scheduling systems to enhance patient care.

In the study by Bhattacharjee and Ray (2016)[2], valuable insights are provided on hospital appointment scheduling systems, specifically targeting the complexities of handling multiple patient classes with varying punctuality, service times, no-show probabilities, and service time variability. By employing discrete-event simulation, the authors evaluate diverse appointment scheduling policies, focusing on sequencing rules and adjustments between appointments to optimize results. Conducting a case study on MRI scheduling in a radiology department, where inpatients and outpatients have distinct needs and punctuality patterns, they simulated patient flows to minimize waiting times and enhance machine utilization. The "individual block variable interval appointment rule," which modifies intervals based on patient class, proved the most effective in balancing wait times and resource use. Their findings emphasize the importance of a customized approach that accommodates patient heterogeneity to boost scheduling efficiency. This study's methodology of modeling diverse patient groups and analyzing scheduling options through simulation can inform your research, especially if your work aims to optimize schedules with constraints like patient variability and resource availability. The application of queueing theory and evaluation of alternative scheduling rules here aligns with efforts to improve hospital operational efficiency.

3 Proposed Approach

3.1 Motivation

The effective scheduling of MRI appointments in healthcare facilities, particularly in tertiary care cancer hospitals, poses a critical challenge due to high patient demand, limited resources, and diverse patient needs. MRI machines operate under tight schedules, with only a finite number of time slots (10 slots in our model) available each day. Idle machine time results in underutilized resources, increasing operational costs, while excessive patient waiting times can lead to dissatisfaction and a reduction in service quality. Given the mixed nature of patient appointments—including various patient classes (such as 'A', 'B', and

'P' representing different urgency and service needs)—it becomes essential to implement a scheduling model that is robust enough to manage these complexities.

In particular, patients in different classes have varying expected service times and levels of punctuality, impacting the predictability and length of appointments. For instance, 'A' patients may represent routine cases with lower variability, while 'P' patients might represent cases with higher urgency and variability in service time. The inherent uncertainty in both patient arrival patterns and service time distributions creates a need for a scheduling approach that can not only handle such stochastic elements but also ensure optimal resource allocation. By addressing these uncertainties through a model that minimizes waiting times and machine idleness, we can improve hospital efficiency, patient experience, and cost-effectiveness.

3.2 Model Formulation Overview

Our model is structured as a Mixed-Integer Nonlinear Program (MINLP), incorporating both continuous and binary decision variables. It is designed to achieve the dual goals of minimizing idle machine time and balancing patient wait times, which is particularly important in environments with a high load of patients and limited resources.

3.2.1 Problem Parameters

- **Slots and Patients:** We consider a 20-hour operating period for the MRI machine, divided into 10 fixed slots. Patients, arriving from different classes ('A', 'B', and 'P'), are assigned to these slots based on their service time needs and arrival probabilities.
- **Service Times and Variance:** The model accounts for expected service times (mean values) and service time variability (standard deviations) for each patient class. Each patient's service time is modeled using a normal distribution, with mean values (μ) and standard deviations (σ) specific to the patient's class.

3.2.2 Decision Variables

- **Binary Decision Variable $x_{i,j}$:** A binary variable $x_{i,j}$ indicates if patient i is assigned to slot j. This binary formulation ensures that each patient is assigned to exactly one slot, and no slot exceeds its capacity.
- **Non-negative Continuous Variable α_j:** For each slot j, a continuous variable α_j represents a bound on the variance constraint, ensuring that the slot utilization does not deviate excessively from the expected service level.

3.2.3 Objective Function

The objective function seeks to minimize the total "weighted time cost" across all slots, formulated as:

$$\text{minimize} \sum_{j=1}^{\text{num\_slots}} \left((\text{num\_slots} - j + 1) \sum_{i=1}^{|\text{arrival\_stream}|} \mu_i \cdot x_{i,j} \right) - \sum_{j=1}^{\text{num\_slots}} j \cdot 120$$

This formulation not only assigns patients to slots but also strategically weighs the time based on slot position to minimize the idle time towards the end of the schedule. The weight assigned to each slot (based on the total number of slots minus the slot position) incentivizes earlier slot utilization, preventing end-of-day idle times. By balancing the assignment costs in this way, we ensure that the MRI machine operates efficiently throughout its scheduled hours.

3.2.4 Slot Assignment Constraints

Each patient must be assigned to exactly one slot, captured by the constraint:

$$\sum_{j=1}^{\text{num\_slots}} x_{i,j} = 1 \quad \forall i$$

This constraint ensures no double-booking and allocates each patient to a unique time slot.

3.2.5 Time Slot Constraints

To further enhance the robustness of the schedule, an additional constraint is introduced to ensure that cumulative weighted time costs and variance factors are accounted for:

$$\sum_{i \in I} \sum_{k=1}^{j} \mu_i \cdot x_{i,k} + c \cdot \alpha_j \geq j \cdot 120 \quad \forall j$$

This constraint enforces a minimum cumulative weighted cost for each slot while incorporating a variance adjustment term ($c \cdot \alpha_j$). Here:

- μ_i represents the weight associated with each patient's assignment.
- c is a scaling constant that modulates the influence of the variance term.
- $j \cdot 120$ reflects the baseline cost threshold for the j-th slot.

By including this constraint, the model ensures that:

1. The cumulative weighted time cost across slots is balanced, promoting efficient scheduling.
2. The variance of slot utilization is effectively controlled, reducing idle times and mitigating delays due to patient heterogeneity.

3.2.6 Second-Order Cone Constraint for Variance Control[8]

One of the distinctive aspects of our model is the integration of a second-order cone constraint to address the variability in service times across patient classes. This constraint is critical in healthcare settings where stochastic elements (e.g., patient punctuality and variability in scan duration) can heavily influence schedule reliability. By controlling the variance of cumulative service times, we create a robust schedule that accommodates patient heterogeneity and mitigates the risk of idle machine times or extended waiting periods due to unanticipated service delays.

Variance Constraint Formulation The constraint is formulated as:

$$\sqrt{\sum_{i=1}^{|I|}\sum_{k=1}^{j}(\sigma_i \cdot x_{i,k})^2} \leq \alpha_j \quad \forall j$$

Here, σ_i represents the standard deviation of service times for each patient, and α_j is an upper bound on the allowed variance for each slot j. This constraint limits the standard deviation in each slot to ensure that scheduling remains feasible within the constraints of the MRI's operating hours.

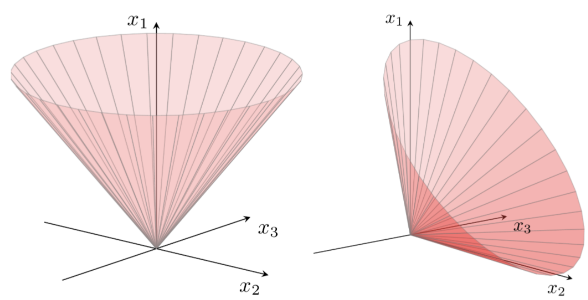

Fig. 3.1 Boundary of quadratic cone $x_1 \geq \sqrt{x_2^2 + x_3^2}$ and rotated quadratic cone $2x_1x_2 \geq x_3^2$, $x_1, x_2 \geq 0$.

Figure 1: Caption describing the image, such as "Illustration of variance constraint in the model."

Control Parameter c A critical parameter c, set using the inverse cumulative density function (derived from a specified confidence level), adjusts the model to control the probability of extreme deviations in service time. This adjustment acts as a "safety margin," reducing the likelihood that a slot's total service time exceeds its capacity due to high-variance cases. By tuning c, we balance robustness with resource utilization, allowing for slight variances without significant risk of overloading slots.

3.2.7 Operational Significance

By incorporating the second-order cone constraint, our model gains robustness against the inherent unpredictability of patient arrivals and scan durations. This feature differentiates our approach from simpler models, which may assume deterministic service times and fail to adequately handle high-variance cases. The constraint thus allows us to create a conservative yet efficient scheduling plan, ensuring that the MRI machine remains fully utilized while minimizing waiting times even under uncertainty. This approach aligns with hospital operational goals to enhance resource utilization and improve patient throughput.

4 Experimental Evaluation

In this section, we discuss the dataset used for evaluating our MRI scheduling model, including the process of data preparation and clustering for patient types.

4.1 Dataset Overview

To address the scheduling challenge effectively, we analyzed data for patients requiring MRI scans in a high-demand tertiary care hospital. The dataset originally included 15 distinct patient types, each characterized by the average time they spend on the MRI machine and the variance in their scan duration. However, we observed that certain types had similar service times and variances, indicating that they could be grouped together without significantly affecting the scheduling model's accuracy.

After analyzing these similarities, we clustered the original 15 types into three broader categories—designated as **A**, **B**, and **P**—each with representative average service times and variances. This clustering allowed us to simplify the model while preserving essential distinctions in scan durations and patient arrival patterns. Each category represents a unique combination of average service time and variability, as shown in Table 1 below:

- **Category A**: Represents patients with an average scan time of 36.596 minutes and a standard deviation of 15.60 minutes.
- **Category B**: Represents patients with an average scan time of 44.823 minutes and a standard deviation of 11.65 minutes.
- **Category P**: Represents patients with an average scan time of 56.807 minutes and a standard deviation of 18.23 minutes.

4.2 Scheduling Parameters and Patient Arrival Stream

Our study considers scheduling for a single day, utilizing 10 identical and contiguous time slots, each with a length of 120 minutes (or 2 hours). Given the machine's total operational time of 1320 minutes per day, this setup allows us to optimize the machine's utilization across all slots while balancing patient wait times and machine idle time.

The arrival stream of patients—representing the sequence in which patients book MRI appointments—is derived from a sample of 30 patients. These 30 patients are distributed across the three categories (**A**, **B**, **P**), as observed in our dataset. This arrival stream that I have randomly generated(for one day) is defined as follows:

> **Arrival Stream**: `['B', 'B', 'B', 'P', 'B', 'A', 'A', 'A', 'A', 'A', 'P', 'A', 'A', 'P', 'P', 'A', 'A', 'A', 'B', 'B', 'A', 'B', 'A', 'B', 'B', 'A', 'P', 'P', 'B', 'B']`

This arrival stream data allows us to evaluate the model's performance by simulating patient assignments based on the anticipated demand for each type of patient on a typical day. Each patient category has a probability associated with it, which corresponds to the likelihood that a newly arriving patient will belong to that category. For simplicity, we assume that patient arrivals for the three categories are independent of each other.

4.3 Appointment Booking Process

Patients typically book their MRI appointments 7-14 days in advance, allowing the hospital staff to allocate each patient to a specific time slot in a manner that optimizes resource utilization. The secretary assigns patients to one of the available slots, and a slot may be shared by several patients depending on the total service time required.

With about 20 different types of MRI scans, each varying in duration, clustering them into three representative categories allows us to streamline the scheduling process while maintaining accuracy. This classification enables the hospital to more effectively predict and manage demand, ensuring the MRI machine remains fully utilized throughout the day with minimized idle time and balanced patient wait times.

Table 1: Average Service Time and Standard Deviation by Patient Category

Patient Category	**Average Scan Time (μ_p)**	**Standard Deviation (σ_p)**
A	36.596 minutes	15.60 minutes
B	44.823 minutes	11.65 minutes
P	56.807 minutes	18.23 minutes

4.4 Solvers Used for Optimization

For this study, I used the **Minotaur**, an open-source software for solving Mixed-Integer Nonlinear Optimization (MINLO) problems, to solve the second-order cone programming (SOCP) formulation. Minotaur is an open-source framework specialized for Mixed-Integer Nonlinear Optimization (MINLO), capable of handling both convex and nonconvex constraints.

Key Features of Minotaur
Minotaur offers several key features that make it a powerful tool for complex optimization tasks. It includes native support for nonlinear functions, providing automatic differentiation capabilities that enable efficient evaluation of nonlinear constraints. Minotaur also incorporates multithreaded solvers, including algorithms like Outer Approximation (OA) and Quadratic Generalization (QG), designed to handle large and complex problems effectively. Additionally, it supports a range of subproblem solvers, such as BQPD, CBC, Clp, CPLEX, FilterSQP, and IPOPT, allowing it to tackle LP, QP, and MILP subproblems with flexibility. Minotaur is also compatible with modeling languages like AMPL and Pyomo, enhancing its versatility and adaptability across various applications.

Minotaur Solvers Employed in This Study:

- **mbnb**: A nonlinear optimization-based branch-and-bound solver for convex MINLPs.
- **mqg**: An LP/NLP-based branch-and-cut solver for convex MINLPs, balancing linear and nonlinear subproblems effectively.

Minotaur's robust support for nonlinear and mixed-integer problems and customization flexibility make it ideal for complex SOCP and MINLP tasks like those in this study.

5 Experimental Results

5.1 MQG Solver Results

The model consisted of 310 total variables, including 300 binary and 10 continuous variables, defined by 50 constraints. These constraints comprised 40 linear and 10 nonlinear components. The objective function was linear with 300 terms, and there were no quadratic or bilinear terms in the constraints, aligning the formulation with the requirements of this specific optimization problem.

Using the **'mqg'** solver, the performance showed significant progress within a 1200-second time limit. However, the solver was unable to fully converge, likely due to the complexity of the second-order cone constraints. During this period, the solver processed over 55,000 nodes and achieved bounds within approximately 70-80% of the gap threshold. The best solution value (upper bound) reached was around **843.9620**, while the lowest bound (lower bound) recorded was approximately **292.0** after 62 iterations. I also ran the model with a 3600-second time limit, but there was minimal improvement beyond 1200 seconds, making the shorter time limit a more efficient choice for this analysis.

Patient Slot Allocation
The slot allocations for the 30 patients are as follows:

Slot	Patients	Total time taken(min)	Overloading(min	Machine Idle Time(min)
Slot 1	{B, A, P, B}	227.96	107.96	0
Slot 2	{A, A, A}	114.15	102.11	0
Slot 3	{A, B, B}	104.56	86.67	0
Slot 4	{A, P, B}	117.83	84.50	0
Slot 5	{B, B, B}	128.74	93.24	0
Slot 6	{P, P}	48.15	21.39	0
Slot 7	{B, B, A}	108.34	9.73	0
Slot 8	{A, A, P}	103.58	0	6.69
Slot 9	{A, P, A}	178.70	58.7	0
Slot 10	{A, A, B}	119.34	58.02	0

Table 2: Patient slot allocations

The table summarizes the slot allocations, total time taken, overloading, and idle time for a machine scheduled across 10 slots. Notably, the machine experienced **6.69 minutes of idle time** only in Slot 8, while it was fully utilized in the remaining slots. Overloading, which indicates time exceeding the slot's allocated duration, is highest in Slot 1 at **107.96 minutes** and decreases progressively across the slots, demonstrating better time management in later slots. Slots with fewer or shorter tasks, like Slot 6, show minimal overloading (**21.39 minutes**). Overall, this highlights an effort to optimize slot usage while maintaining machine efficiency.

5.2 Encountering Jacobian Error with the `mbnb` Solver

When using the `mbnb` solver for the optimization, we encountered a recurring issue related to the Jacobian evaluation. The solver repeatedly threw the following error:

IpoptFunInterface: error in evaluating jacobian. Setting eval_within_bnds to True.

This error was traced to the non-differentiability of the Second-Order Cone Programming (SOCP) constraint at the origin. The lack of smoothness at this point caused the solver to fail in calculating the necessary Jacobian matrix, thereby halting the optimization process.

To resolve this issue, two different methods were employed:

5.2.1 Introducing a Small Epsilon Value to the SOCP Constraint

A very small epsilon value(10^{-6}) was added to the SOCP constraint to ensure that the constraint remains differentiable at the origin. This adjustment allowed the solver to bypass the non-differentiability issue, and the Jacobian was evaluated correctly, enabling the solver to proceed with optimization but introducing an epsilon value in the SOCP constraint has noticeably increased the convergence time, likely due to the added complexity it brings to the optimization process. The solver output reveals the current bounds, with a lower bound of **292.00** while the upper bound remains at infinity, indicating that an optimal upper bound has yet to be identified. This result, along with the observed gap percentage and node counts, reflects the heightened computational effort required to approach an optimal solution within the refined feasible region imposed by the epsilon adjustment.

5.2.2 Log-Sum-Exp Approximation for Variance Constraints[11,12]

A second approach was applied, involving the use of a Log-Sum-Exp (LSE) approximation to handle the variance constraints. This method smooths the L2 norm used in the constraint, making it differentiable and thus more manageable for the solver. The Log-Sum-Exp approximation replaces the direct L2 norm with a smoother function that behaves well at the origin, ensuring no issues with the Jacobian evaluation.

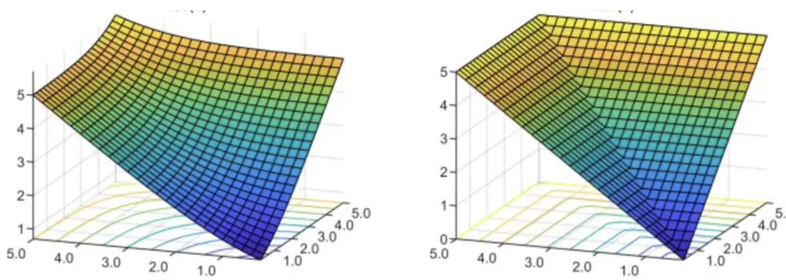

Figure 2: Caption describing the image, such as "Illustration of variance constraint in the model."

The output from the solver demonstrates that the lower bound reached 207.43, with the upper bound remaining infinite throughout the iterations.

No feasible solutions were found during this process, and the gap between the lower and upper bounds remained infinite, suggesting that the solver struggled to converge within the given time frame. The significant computational resources spent in solving node relaxations (over 1031 seconds) and numerous strong branching calls indicate the computational complexity introduced by the LSE approximation for this specific constraint setup. This may imply that, while the LSE approximation helps make the problem differentiable, it also introduces challenges for achieving a tight bound or feasible solution efficiently within the time constraints.

6 Future Work

The results highlight some key areas for improvement in the current approach. When using the `mbnb` solver, the upper bound remains at infinity, suggesting that the methods employed may not be well-suited to the type of constraints in this problem. This issue implies that exploring alternative methods and solvers could be beneficial. Future work will include testing different relaxation techniques, constraint formulations, and optimization solvers that might provide better bounds and feasible solutions, improving the effectiveness of the optimization process for handling complex variance constraints.

Additionally, with the `mqg` solver, the time taken to converge is notably high. To address this, I plan to explore strategies to improve convergence speed, such as optimizing model formulation or leveraging more efficient solution techniques. Reducing the convergence time would allow for faster computations and enhance the practical applicability of the approach in larger or time-sensitive scenarios.

7 Acknowledgment

I would like to express my heartfelt gratitude to my guide, Prof. Ashutosh Mahajan, for his invaluable guidance and support throughout this project. His encouragement and insights were instrumental in helping me understand and address the complexities of this problem.

I am also deeply thankful to Simran Lakhani for her patient explanations and practical insights. During our visit to Tata Hospital, she took the time to thoroughly explain each step of the data collection process and the categorization of patients into three types, which greatly clarified the problem's scope and challenges.

Additionally, I would like to thank Saurabh for his assistance in helping me understand and effectively use Minotaur's solver, which was crucial for the computational aspects of this project.

This work would not have been possible without the support and collaboration of each of these individuals, and I am immensely grateful for their contributions.

References

[1] J. Marynissen, E. Demeulemeester, "Literature review on multi-appointment scheduling problems in hospitals," *European Journal of Operational Research*, vol. 272, no. 2, pp. 407–419, 2019. doi: 10.1016/j.ejor.2018.03.001.

[2] A. Bhattacharjee, P. K. Ray, "Patient appointment scheduling and capacity planning for radiology services," *Operations Research for Health Care*, vol. 3, no. 4, pp. 177–190, 2016. doi: 10.1016/j.orhc.2014.10.002.

[3] MOSEK ApS, "Conic Quadratic Optimization," *MOSEK Modeling Cookbook*, [Online]. Available: https://docs.mosek.com/modeling-cookbook/cqo.html.

[4] MILP Tutorial overview, *YouTube*, available at: https://www.youtube.com/watch?v=mUjkCeLx6$_4list = PLHiHZENG6W8Dt7ub1H1RsF4xXB_YtxXv-$.

[5] S. Lakhani, "Analyzing Appointment Scheduling for Oncology Patients in a Hospital," available at: https://papers.ssrn.com/sol3/papers.cfm?abstract$_id = 4806354$.

[6] N. Higham, "What is the Log-Sum-Exp function?" available at: https://nhigham.com/2021/01/05/what-is-the-log-sum-exp-function/.

[7] P. Andersen, K. Andersen, "The log-sum-exp approximation," *European Journal of Operational Research*, available at: https://www.sciencedirect.com/science/article/pii/S0377221718302108.

[8] SOCP Example (Image), *MOSEK Modeling Cookbook*, available at: https://docs.mosek.com/modeling-cookbook/cqo.html.

[9] L. Vandenberghe, S. Boyd, "Introduction to Cone Programming and Applications," *Course Notes, University of Chicago*, available at: https://www.stat.uchicago.edu/ lekheng/courses/310w13/socp.pdf.

[10] Minotaur toolkit, at: https://minotaur-solver.github.io/.

[11] Log-Sum-Exp, at: https://en.wikipedia.org/wiki/LogSumExp.

[12] https://nhigham.com/2021/01/05/what-is-the-log-sum-exp-function/