Clinical Trial Optimization Success Story

Laia Domingo
Jannes Stubbemann

October 2024

1 Introduction

2 Use case

Each year, pharmaceutical companies invest billions of dollars into developing new medications. Despite these monumental efforts, 70% of clinical trials fail, and the average cost to bring a new drug to market reaches $1 billion.

Clinical trials are the linchpin of medical advancement, determining whether a new drug is safe and effective for public use. A failed trial doesn't just represent a financial setback; it delays potentially life-saving treatments from reaching patients who need them most. Understanding and addressing the root causes of trial failures is therefore crucial for both the medical community and society at large.

One significant factor behind these failures is poor patient stratification. In most interventional therapeutic clinical trials, patients are randomly assigned to treatment and control groups. While randomness aims to create statistically equivalent groups, it often leads to **imbalances in critical patient attributes** —such as age, gender, ethnicity, or genetic markers. These imbalances can skew results, making it difficult to determine a drug's true efficacy and safety profile.

Imagine, for example, a trial for a cardiovascular drug where one group inadvertently has a higher proportion of smokers. Since smoking is a risk factor for heart disease, this imbalance could affect the drug's apparent efficacy, leading to misleading conclusions or even the failure of the trial.

2.1 A mathematical approach to patient stratification

To address the challenges of traditional patient assignment, we can formulate patient stratification as a **constrained optimization problem**. This mathematical approach aims to create treatment and control groups that are as similar as possible across all relevant attributes.

Each patient i has r covariates $\vec{w}_i = (w_{i1}, \cdots, w_{ir})$ that are relevant for predicting outcomes. There are n patients participating in the trial. These patients are divided equally into $m \geq 2$ treatment groups, with $k = n/m$ patients per

group. We introduce decision variables $\vec{x} = \{x_{ip}\}$, where $x_{ip} = 1$ if patient i is assigned to group p, and $x_{ip} = 0$ otherwise.

The goal is to **minimize the discrepancy** d between any two groups based on the weighted sum of the first (means) and second (variances and covariances) moments of the covariates.

$$d = \sum_{s=1}^{r} |\Delta\mu_s| + \rho \sum_{s=1}^{r} |\Delta\sigma_{ss}| + 2\rho \sum_{s=1}^{r} \sum_{s'=s+1}^{r} |\Delta\sigma_{ss'}| \tag{1}$$

Here, $\Delta\mu_s$ represents the difference in the mean of covariate s between groups, calculated as:

$$\Delta\mu_s = \frac{1}{n} \sum_{i=1}^{n} w_{is}(x_{i1} - x_{i2})$$

Similarly, $\Delta\sigma_{ss'}$ represents the difference in the covariance between covariates s and s' between groups:

$$\Delta\sigma_{ss'} = \frac{1}{n} \sum_{i=1}^{n} w_{is} w_{is'}(x_{i1} - x_{i2})$$

The parameter ρ adjusts the relative importance of balancing the variances and covariances compared to the means.

Balancing the first moments, or means, ensures that the average values of each covariate are similar across groups, reducing bias in estimated treatment effects. Balancing the second moments, which include variances and covariances, ensures that the variability within each covariate is similar and that relationships between covariates are maintained across groups.

To achieve a feasible and balanced assignment, the following constraints are applied:

1. **Equal Group Sizes:** Each group must have exactly k patients, ensuring that the sample sizes are the same across groups:

$$\sum_{i=1}^{n} x_{ip} = k, \quad \forall p \in \{1, \cdots, m\}$$

2. **Unique Assignment:** Each patient must be assigned to exactly one group, preventing any overlap or omission:

$$\sum_{p=1}^{m} x_{ip} = 1, \quad \forall i \in \{1, \cdots, n\}$$

While this formulation can be very powerful, solving this problem using classical computing methods is computationally intensive. For large numbers of patients (n) and covariates (r), the problem becomes **NP-hard**, meaning that the computation time increases exponentially with the size of the input

data. This computational intensity limits the practicality of classical optimization methods for real-world clinical trials involving thousands of patients and numerous covariates.

For this competition, the data originates from the Mayo Clinic trial carried out between 1974 and 1984, involving 312 participants in a randomized trial for primary biliary cholangitis, an autoimmune condition that affects the liver. We identified three statistically significant covariates exhibiting the most substantial impact, including the age of the patient (w_1), alkaline Phosphatase in U/liter (w_2) and prothrombin time in seconds (w_3).

3 Participant solutions

We were thrilled to receive numerous high-quality submissions. To evaluate them, we developed a leaderscore system where higher scores indicated lower discrepancies within groups. Not only did the submissions score exceptionally well, but they also showcased remarkable diversity. Each brought unique insights and innovative ideas, adding immense value to the overall challenge. Here is a brief description of the winning approaches:

1. **Julien Mellaerts (1st Prize)**: This custom hybrid quantum-classical workflow optimizes the use of classical and quantum resources, reducing time complexity while maintaining optimal accuracy of patient stratification

2. **Peter Yang (2nd Prize)**: A hybrid quantum-classical algorithm that solves a constrained optimization problem, leveraging the computational power of quantum computing and the flexible formulation of classical solvers

3. **Oleksii Adamov (3rd Prize)**: This solution integrates a custom Quantum Approximate Optimization Algorithm (QAOA) with a smart clustering strategy to reduce problem size and optimize qubit usage

4. **David Esteban Bernal Neira & the SECQUOIA Team (Special Prize)**: The hybrid approach combines Graver Augmented Multistart Algorithm (GAMA) and with quantum annealing to find feasible solutions and compute elements of Graver basis, optimizing the use of classical and quantum resources.

4 Benchmarking

Based on the participant's solutions, we performed experiments comparing the performances and execution times for different patient sizes n, compared with classical benchmarks. In this study, the models evaluated included classical, quantum-inspired, and quantum approaches as follows:

- **GAMA:** Proposed by the SECQUOIA Team, it utilizes Graver bases, which provide a structure for generating integer feasible directions, to explore the search space systematically. The process begins by formulating the problem as a minimization problem $\min_{y\in\{0,1\}^n} f(y)$ with binary variables, subject to linear constraints $Ax = b$. The Graver basis G is computed by obtaining the kernel of A, where x_0 satisfies $Ax_0 = 0$, and take the Graver basis as $G = x^0 \setminus \{0\}$. Initial feasible solutions ($Ax_0 = b$) are generated through quantum annealing by solving a QUBO formulation with D-wave computers. The algorithm then iteratively improves each solution by augmenting it in the direction of Graver basis vectors until no further improvement is possible, ensuring optimal or near-optimal solutions.

- **HybridCQM:** Inspired by Peter's solution, another approach is to solve the constrained optimization problem using D-wave hybrid CQM sampler. This sampler combines classical and quantum resources to handle complex optimization problems with both linear and quadratic constraints. It divides the problem into smaller subproblems, solves them using quantum annealing and classical techniques, and then integrates the solutions to find the best result. In this case, we minimize the discrepancy as defined in Eq. 1. This formulation leads to a constrained quadratic model with $n-1$ binary variables and 9 continuous variables.

- **Kerberos:** Another alternative, inspired by Julien's solution, is to formulate this problem as a QUBO, where we need to minimize an expression of the form $\min_y y^T Q y$. To do so, we replace the original absolute values by a square, so that we redefine the discrepancy function as:

$$d_2 = \sum_{s=1}^{3} (\Delta\mu_s)^2 + \rho \sum_{s=1}^{r} (\Delta\sigma_{ss})^2 + 2\rho \sum_{s=1}^{r} \sum_{s'=s+1}^{r} (\Delta\sigma_{ss'})^2 \tag{2}$$

  In this case, the KerberosSampler is used to solve this QUBO problem. It is a hybrid solver from D-Wave that integrates both classical and quantum computing resources to solve optimization problems. It uses a combination of three techniques:

  - **Tabu Search:** A classical algorithm that helps the solver escape local minima by maintaining a list of recently visited solutions (the tabu list) and preventing the algorithm from revisiting them.
  - **Simulated Annealing:** Another classical optimization method that mimics the physical annealing process by gradually lowering the "temperature" of the system to converge towards an optimal or near-optimal solution.
  - **QPU Subproblem Sampling:** This involves quantum annealing, where a portion of the problem (typically the most impactful variables) is sent to D-Wave's Quantum Processing Unit (QPU). The

quantum annealer explores the solution space and helps to solve subproblems that are difficult for classical solvers.

- **Tabu:** Instead of using a hybrid workflow, we can also use a classical (quantum-inspired) solver, called Tabu search. Tabu Search is a metaheuristic optimization technique designed to efficiently explore the solution space of hard optimization problems. The algorithm begins by exploring the local neighborhood of the current solution. It evaluates nearby solutions, moving towards the best available option. To avoid cycling (revisiting the same solutions repeatedly), Tabu Search maintains a tabu list of recently explored solutions (or specific moves), marking them as "forbidden" or "tabu" for a certain number of iterations. The tabu list allows the algorithm to escape local minima by preventing it from immediately returning to previously explored, suboptimal solutions.

- **QAOA:** The final quantum approach, proposed by Oleksii, is to use the Quantum Approximate Optimization Algorithm (QAOA), a hybrid quantum-classical algorithm designed to solve combinatorial optimization problems which can be expressed in terms of a QUBO. QAOA alternates between two Hamiltonians: the Cost Hamiltonian H_C, which encodes the optimization problem based on the QUBO matrix Q, and the Mixer Hamiltonian H_M, which typically consists of Pauli-X operators that explore the solution space. The algorithm alternates between applying the cost and mixer Hamiltonians with parameters γ and β, while a classical optimizer is used to adjust the parameters to minimize the expectation value of H_C, which represents the problem's cost function. We also use the Conditional Value at risk (CVar) estimator.

- **Gurobi:** Gurobi is a classical leading optimization software extensively used for solving a wide range of optimization problems, including QUBOs. It is renowned for its speed and reliability in addressing classical optimization problems. It will be used to benchmark agains the quantum methods.

In this study, we executed six methods across varying patient sizes $n \in 10, 20, 50, 80, 100, 150, 200$, with GAMA, HybridCQM, and Kerberos leveraging D-Wave's quantum and hybrid solvers, while the other methods utilized classical CPUs. Notably, QAOA, though classically simulated here, could be executed on a gate-based quantum computer. The results, in terms of discrepancy and execution times, are shown in Fig. 1.

From a performance perspective, Kerberos, and Tabu search algorithms consistently delivered very similar results, demonstrating high accuracy in minimizing the discrepancy, which was very close to the classical benchmark. The hybrid and quantum-inspired approaches (such as GAMA and HybridCQM) also performed well, with only a slight increase in discrepancy compared to the classical benchmarks. However, the QAOA algorithm showed a noticeable decline in performance as the number of patients increased. This drop in accuracy may

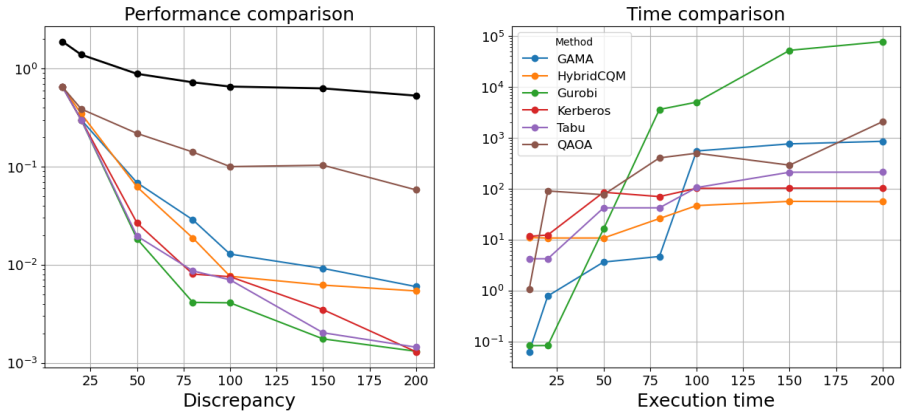

Figure 1: Time and performance comparison of multiple classical, quantum-inspired and hybrid quantum-classical solutions. The black line represents the average discrepancy obtained by random patient stratification.

be attributed to the fact that, in this study, we had to partition the patient groups into smaller subgroups to run QAOA on classical simulators. We hypothesize that running QAOA on a real quantum computer could yield better results, particularly for larger patient sets, as it can handle the problem more holistically without the need for partitioning.

Additionally, as the number of patients increased, the optimal discrepancy values showed a more significant divergence from the average discrepancy obtained through random assignments. This indicates that random assignment of patients into cohorts can lead to much higher discrepancies, which underscores the value of optimization in creating balanced patient groups for clinical trials.

When it comes to execution times, the Gurobi solver, while offering the lowest discrepancy, demonstrated a significant drawback in terms of scalability. As the number of patients increased, Gurobi's execution time grew exponentially, with the 200-patient case taking more than two orders of magnitude longer than the hybrid approaches. This makes Gurobi less practical for larger datasets, especially in time-sensitive applications like clinical trials. In contrast, the hybrid solvers, such as Kerberos and HybridCQM, achieved similar performance but within a much shorter time frame. This efficiency makes quantum-inspired and hybrid methods appealing for real-world applications where both speed and accuracy are critical.

Overall, our results suggest that while classical solvers like Gurobi provide optimal solutions, quantum-inspired and hybrid solvers offer a strong trade-off between accuracy and efficiency, making them promising candidates for fast, large-scale patient stratification in clinical settings. Moreover, as quantum hardware advances, methods like QAOA could close the performance gap and offer even more scalable solutions.