Framework for Optimal Budget Allocation of HIV Intervention Policies

Ali El Moselhy[†]

Project advisor: Prof. Anna Bershteyn

Abstract. HIV has been responsible for over 35 million deaths. Over 37.9 million people around the world live with HIV (PLHIV). Due to the development of different intervention strategies, HIV is no longer a death sentence. In this paper, we propose a framework to optimally allocate capital to those intervention strategies subject to budget constraints. The framework uses the Epidemic MODel (EMOD) model to evaluate the effectiveness of any suggested intervention policy and uses the second-order gradient-based Newton's Method algorithm for the optimization. To improve the computational efficiency of our framework, we approximate the complex EMOD model using a tensor-train-based surrogate model. Tensor trains allow us to escape the curse of dimensionality and enable efficient evaluation of the first and second order gradients. Our framework is adaptive and iterative in the sense that it efficiently improves the accuracy of the surrogate model with every optimization cycle. As such, at convergence, we are guaranteed to reach an optimum with respect to the full EMOD model and not with respect to the approximate surrogate model. We used our algorithm to optimally allocate the intervention budget for the specific cases of South Africa and Kenya. The Kenya problem is particularly high-dimensional (42-dimensional). The results show that, under a constrained budget, AntiRetroviral Therapy (ART) medication is the most effective investment in reducing HIV-related death. This is a far more effective investment than other popular intervention strategies such as Pre-Exposure Prophylaxis (PrEP), Voluntary Male Medical Circumcision (VMMC), or Home Counseling and Testing (HCT) programs.

1. Introduction.

Around the world, there exist over 37.9 million people living with HIV (PLHIV), and HIV has been responsible for over 35 million deaths to-date [9]. The most affected regions are Eastern and Southern Africa, with 20.6 million PLHIV, accounting for around 54% of all PLHIV [6].

Interventions are public health measures taken in order to limit the spread of HIV [8]. This can range from treatments, such as Anti-Retroviral Therapy (ART), to preventatives, such as Post- and Pre-Exposure Prophylaxis (PEP and PrEP), to additional programs meant to increase awareness and access to care, such as AnteNatal Care (ANC) and Home Counseling & Testing (HCT) programs. Intervention programs are expensive; per-person yearly costs for medications can be over $100 USD. Non-medical interventions are cheaper but not free [8]. Since 2021, annual resources for HIV treatment available to Eastern and Southern Africa have decreased by 5% [6]. There exists a need for efficient allocation of resources to maximise the impact of every dollar spent in the region [6].

The problem of optimal budget allocation for HIV intervention policies has been addressed by ¡!– several research groups –!¿, most notably the Optima Group. The Optima group has published numerous reports on budget optimization, typically focusing on a single country [5]. They use their Optima HIV model, a deterministic compartmental model, in their analyses [5]. They usually focus on around 7 intervention strategies [7]. In this work, we use the Epidemiological MODeling software (EMOD) model, a stochastic

[†]Location (myemail@gmail.com, website).

agent-based model, and scale up the dimensionality of the problem to more than 40 dimensions. This allows us to better tailor the intervention policies to the specific population at hand.

EMOD is a stochastic agent-based model developed by the Institute for Disease Modeling (IDM) [1]. EMOD is designed to be highly modular, allowing for many different diseases and regions to be simulated, based upon the input files provided to the model [1]. For this work, we will be using two sets of input files, the first modeling HIV in South Africa and the second modeling HIV in Kenya. These input files were calibrated by Prof. Kim in the Bershteyn Lab group at NYU Langone[1].

The main contribution of this paper is a complete framework for efficient budget allocation to HIV intervention policies. We use a second-order Newton's method optimization algorithm. We did several contributions to improve the computational efficiency of the algorithm. Those contributions can be divided into two main categories. The first is the use of tensor train-based surrogate models to reduce the evaluation time of the model. The second is a novel approach to ensure that the solution obtained by our algorithm is optimal with respect to the true EMOD model and not with respect to the surrogate model.

With regards to the surrogate model, please note that EMOD typically takes 15 minutes to complete a simulation, prohibitively long for most optimization algorithms. Case in point, evaluating 100 simulations would take 25 hours to run. In order to reduce evaluation time, we will use tensor-train based surrogate models to approximate the EMOD model.

Tensor trains were originally conceived in a seminal paper by Prof. Ivan Oseledets, as high-dimensional extensions of the Singular Value Decomposition (SVD) [10]. They were successfully used in the approximation of high-dimensional functions and uncertainty quantification [3] [4] [13]. In addition to their ability to break the curse of dimensionality and efficiently approximate high-dimensional functions, tensor trains offer us multiple benefits. First, they fit well within gradient based optimization algorithms because computing their gradients is extremely efficient. Second, they allow us to separate the algorithm into an offline mode and an online mode. In the offline mode, we fit a surrogate model without the need for online data gathering. In the online mode, we use additional data to refine the surrogate model.

With regards to the novel termination criteria, we added a step at the end of every optimization cycle where we used the EMOD model to certify the obtained solution. We only accept the solution if it is within the budget constrains while yielding an optimal strategy. Otherwise, we use that new EMOD simulation to update the surrogate model and thereby improving our predictions for the next optimization cycle. It goes without saying that all function and gradient evaluations within any optimization cycle is performed using the surrogate model. This reduces the computational time for each cycle by several orders of magnitude. We used our algorithm to optimally allocate the intervention budget for the specific cases of South Africa and Kenya. The Kenya problem is particularly high-dimensional (42-dimensional). The results show that, under a constrained budget, AntiRetroviral Therapy (ART) medication is the most effective investment in reducing HIV-related death. This is a far more effective investment than other popular intervention strategies such as Pre-Exposure Prophylaxis (PrEP), Voluntary Male Medical Circumcision

(VMMC), or Home Counseling and Testing (HCT) programs.

This paper is organized as follows: In section 2, we describe EMOD in greater detail. In section 3, we define the objective functions and metrics we will be using for the remainder of the paper. In section 4, we describe the proposed algorithm, and in section 5, we report the results of our application of the proposed algorithm to two budget optimization problems. We conclude this paper in section 6, with a discussion on the implication of our results and directions for future work.

2. Model Description. Epidemic MODel (EMOD) is a stochastic, agent-based disease model [1]. EMOD simulates the propagation of an endemic disease within a population [1]. EMOD has enough degrees of freedom in order to allow the user to simulate a variety of diseases and populations [1]. In this paper, we will be using EMOD to model HIV epidemics in both South Africa and Kenya [1]. The population characteristics are specified using config files that have been calibrated by Prof. Kim in the Bershteyn Lab group.

Upon specifying a parameter set, EMOD generates a sample population consistent with the input parameters, and starts simulating the disease propagation among the population using discrete time steps among the population over the period of 1960 to 2050. These 90 years are discretized using timesteps of length 1 month [2]. Each member of the population can be at one of several states. Each state describes the status of the individual with respect to HIV. The actual HIV states within EMOD are rather complex and fine-grained, but for simplicity, coarser-grain examples are HIV negative, HIV positive, on ART or on PrEP [1]. At each time step, there is a probability that any one agent will transition from one state to another. These probabilities are determined by the input parameters describing the population characteristics. As a consequence of these transition probabilities, the output of an EMOD simulation for a given set of input parameters is stochastic. This means that if we run the simulation twice, we will get two slightly different simulation results. To minimize the effect of randomness, EMOD runs 50 simulations for each set of input parameters. The results are then averaged in order to get a single final result for a given simulation [1].

Within each EMOD simulation, there are parameters which control the interventions used to limit the spread and effect of HIV. The user can change those parameters in order to simulate the effect of different intervention strategies. In this paper, we will deal with 5 interventions: Anti-Retroviral Therapy (ART), Pre-Exposure Prophylaxis (PrEP), Voluntary Male Medical Circumcision (VMMC), Ante-Natal Care (ANC) and Home Counseling and Testing (HCT).

- ART is a medication taken by PLHIV. ART lowers the amount of HIV in the body to undetectable levels, meaning that PLHIV on ART can live as if they do not have HIV [8].
- PrEP is a medication that is taken by those who do not have HIV and greatly reduces their chances of infection upon exposure to HIV [8]. People on PrEP are typically high-risk individuals [8].
- VMMC is a form of circumcision that lowers the chance for a man to both receive and transmit HIV [8].
- ANC is a form of testing and medication given to pregnant women to reduce their

probability of passing HIV to their child [8].
- HCT a form of outreach which teaches people about the symptoms of HIV and encourages them to get tested [8].

The output of each simulation is a comprehensive description of the events within the simulation. For example, the amount and time any intervention was administered, the amount of HIV infections and deaths, and progression of each person through different states. From those outputs, one can derive the total amount of money spent on interventions, the number of people that died, the cause of their deaths, the time at which they died, and their age at the time of death.

3. Optimization Problem. Money is a limiting factor for the interventions policies that can be implemented within any population. Consequently, figuring out the combination of interventions that best reduces HIV-attributed death for any given amount of money is a matter of life or death [11].

We define Cost as the total amount of money spent on interventions past a threshold year, t_0

$$\text{Cost} = \sum_{t \geq t_0} \sum_{i \in I} (N_{ti} * C_i * df^{t-t_0})$$

where:
- N_{ti} is the number of times intervention i is given in year t. This is obtained from the simulation results.
- C_i is the per-unit cost for intervention i. This is obtained from [8].
- df is the decay factor, typically 0.97. This factor is applied to the product to mitigate uncertainty caused by randomness.
- t_0 is a threshold year we chose to be 2020.
- I is the set of 5 intervention policies described in Section 2.

To compute HIV-attributed death we use a quantity called disability adjusted life years (DALYs). Given a PLHIV who has died from HIV, DALYs are calculated by taking the difference of their life expectancy and their simulated life span. In short, DALYs are the measure of how many years were taken from them due to HIV.

We begin counting DALYs after t_0. For each person who died due to HIV:

$$\text{DALYs} = \sum_{d \in D} (E_d - R_d) * df^{t_d - t_0}$$

where:
- D is the set of all people who have died due to HIV.
- E_d is the expected lifespan. This is obtained from UN average lifespan measures [12].
- R_d is the actual lifespan. This is obtained from the simulation results.
- t_d is the time of death, in years
- t_0 is the threshold year we chose to be 2020.
- df is defined as before.

We define the optimal investment strategy as a set of input parameters to EMOD. This set of input parameters controls the usage of interventions. Note that these parameters

do not represent population characteristics or disease characteristics. Our optimization problem to find the set of input parameters $(x_0,\ x_1,\ \cdots,\ x_{n-1})$ which minimize DALYs subject to certain budget constrains:

$$\begin{aligned} \min_{x_0, x_1 \cdots x_{n-1}} \quad & \text{DALYs}(x_1, x_2, \cdots x_{n-1}) \\ \text{s.t.} \quad & \text{Cost}(x_1, x_2, \cdots x_{n-1}) < \text{Budget} \end{aligned} \tag{3.1}$$

4. Proposed Algorithm. Each one of our parameters x_i is normalized to be in the range between [0-1]. However, it does not make practical sense for some parameters to take on the full range of [0-1], so we will limit their input range.

In this section, we will give a top-level description of the main ingredients of our optimization algorithm. The details will follow in subsequent subsections. To solve the optimization problem, we need to run many simulations corresponding to different parameter sets. As previously stated, each such simulation takes approximately 15 minutes to run. This results in an extremely slow optimization problem. In order to accelerate the optimization problem, we propose to use a surrogate model-based approach where we approximate the EMOD simulation using tensor trains. The tensor trains typically take less than one second to evaluate. After each optimization cycle (which involves many evaluations of the surrogate model), we reach a hypothesized optima. This optima is based on the surrogate model and may or may not hold true under the EMOD simulation. To validate that optima, we simulate it using EMOD. If the EMOD simulation agrees that the hypothesized optima is indeed optimal, then we stop our algorithm. Otherwise, we refit our surrogate model on the new data, and rerun the optimization algorithm. Therefore, the main algorithm consists of the following steps:

Sample parameter space to obtain 'training' data
while True **do**
  Construct surrogate models using the training data
  Find optimal allocation policy using the surrogate models
  Simulate hypothesized optimum using EMOD
  if simulation satisfies stopping criteria **then**
    Break
  else
    Augment training data using new data
  end if
end while

Two important ingredients are required to make the above work. First is the optimization algorithm. For that, we use Newton's algorithm (a gradient-based algorithm). This algorithm is explained in subsection 4.2. The second ingredient is the surrogate model. For that we use Tensor Trains as described in the following section.

4.1. Tensor Train. We use tensor trains [10] in order to accurately approximate the output of the EMOD simulation for any given set of input parameters. Tensor trains allow us to find a high-dimensional surrogate model while breaking the curse of dimensionality [10]. That is because the degrees of freedom to fit the tensor train grow linearly with the number of input parameters (and not exponentially) [10].

A tensor train is an approximation of a complex model in terms of the product of univariate matrix functions. This can be interpreted in the context of separation of variables and/or high-dimensional SVD

$$\hat{f}(x_0, x_1, \cdots x_{n-1}) = G_0^T(x_0) G_1(x_1) \cdots G_{n-1}(x_{n-1})$$

where:

- $G(x_i)$ is a matrix function of the variable x_i, and is of dimension $r_i \times r_{i+1}$.
- $r_0 = r_n = 1$. This means that the end product of all the $G_i(x_i)$'s is a scalar.
- r_i is the approximation rank. It can be determined by the user or revealed by the algorithm. In this paper, we choose to determine the rank by the user. In a later extension of the work we will show how the rank can be revealed.

We choose to approximate the matrix function $G_i(x_i)$ in terms of monomial basis functions ($b_j(x_i) = \frac{x_i^j}{j!}$), leading to:

$$G_i(x_i) = \sum_{j=0}^{k_i} b_j(x_i) C_{ij}$$

where C_{ij} is the coefficient matrix of dimension $r_i \times r_{i+1}$ and k_i is the order of the polynomial approximation for x_i. For the rest of the paper, we will drop the i suffix and assume all parameters use the same order approximation.

Fitting the surrogate model involves computing all C_{ij} in order to best approximate $f(x_0, x_1, \ldots x_{n-1})$. The total number of unknowns is

$$\sum_{i=0}^{n-1} r_i r_{i+1} k_i = k \sum_{i=0}^{n-1} r_i r_{i+1}$$

4.1.1. Fitting Procedure. To fit our model, we need data. We use an offline procedure to collect outputs from EMOD corresponding to random samples of the parameter space. In the rest of this paper, we will refer to the samples of the input parameters with $x_0^m, \ldots, x_{n-1}^m$, where m is the index of the sample, and we will refer to the output of EMOD with $f(x_0^m, \ldots, x_{n-1}^m)$. Then, we solve the following nonlinear least squares optimization problem:

$$\min_{C_{ij}} \quad \sum_m (f(x_0^m, \ldots, x_{n-1}^m) - \hat{f}(x_0^m, \ldots, x_{n-1}^m))^2 \tag{4.1}$$

where x^m refers to the m^{th} input parameter set, $f(x_0^m, \ldots, x_{n-1}^m)$ is output of the EMOD simulation and $\hat{f}(x_0^m, \ldots, x_{n-1}^m)$ is output of the surrogate model. We will now intentionally drop the m superscript in order to simplify the following explanation, but we will restore it at the end.

This fitting problem is a nonlinear optimization problem, and we use an iterative algorithm to solve it. Instead of attempting to solve for all C_{ij} simultaneously, we will iterate over i, and for each i, we will fix all unknowns $C_{\ell j}$ where $\ell \neq i$. As such, we will solve for $G_i(x_i)$ individually. Each $G_i(x_i)$ is a linear combination of unknown coefficient matrices, which can be reformulated into a linear least squares problem as follows:

$$\begin{aligned} \hat{f}(x_0, \ldots, x_{n-1}) &= f(x_0, \ldots, x_{n-1}) \\ G_0^T(x_0) \ldots G_i(x_i) \ldots G_{n-1}(x_{n-1}) &= f(x_0, \ldots, x_{n-1}) \end{aligned}$$

Let $\mathbf{u}^T = G_0^T(x_0) \ldots G_{i-1}(x_{i-1})$ and $\mathbf{v} = G_{i+1}(x_{i+1}) \ldots G_{n-1}(x_{n-1})$. Substitute $G_i(x_i) = \sum_{j=0}^{k} b_j(x_i) C_{ij}$ to obtain

$$\mathbf{u}^T \left(b_0(x_i) C_{i,0} + \cdots + b_{k-1}(x_i) C_{i,k-1} \right) \mathbf{v} = f(x_0, \ldots, x_{n-1})$$

Using Kronecker product notation, rewrite the above equation as

$$(\mathbf{v}^T \otimes \mathbf{u}^T) \operatorname{vec}(b_0(x_i) C_{i,0} + \cdots + b_{k-1}(x_i) C_{i,k-1}) = \operatorname{vec}(f(x_0, \ldots, x_{n-1}))$$

Let $\mathbf{b}^T = [b_0(x_i), \ldots, b_{k-1}(x_i)]^T$.

$$(\mathbf{v}^T \otimes \mathbf{u}^T \otimes \mathbf{b}^T)[\operatorname{vec}(C_{i,0}), \ldots, \operatorname{vec}(C_{i,k-1})] = \operatorname{vec}(f(x_0, \ldots, x_{n-1}))$$

We now have the form of a linear least squares solve. We can repeat the same analysis for every one of the sample points. We then assemble those equations into a linear system of equations of the form:

$$A[\operatorname{vec}(C_{i,0}), \ldots, \operatorname{vec}(C_{i,k-1})] = \mathbf{y}$$

where the m^{th} row of A is $\mathbf{v}^T \otimes \mathbf{u}^T \otimes \mathbf{b}^T$ and the m^{th} row of $\mathbf{y}$ is $f(x_0^m, \cdots, x_{n-1}^m)$.

Solving this least squares problem will yield the desired coefficient matrices for the unknown $G_i(x_i)$. We will then iterate over i, in order to find all $G_i(x_i)$. We terminate this procedure after reaching desired accuracy, where we define the accuracy based on two measures:

- The L_2 norm of the EMOD output prediction error: $\frac{||\mathbf{f}(\mathbf{x})-\hat{\mathbf{f}}(\mathbf{x})||}{||\mathbf{f}(\mathbf{x})||}$.
- The L_2 norm of the change in C_{ij} from one iteration to the next.

It should be mentioned that, before the least squares solve is performed, a preconditioner is applied to A. This preconditioner is a diagonal matrix which reduces the magnitude of the entries of A, mitigating the chance of numerical error when performing the least squares solve. This preconditioner, P, has diagonal entries $p_{ii} = \frac{1}{\max A}$, where $\max A$ denotes the largest entry of A.

Note that $b_j(x_i)$ did not necessarily need to be monomials. We could have replaced the monomials with orthogonal polynomials (e.g. Hermite polynomials, Legendre polynomials, etc.), and utilized the same process. By the end of the fitting algorithm we will end up with all C_ij and consequently with a complete description of the function $\hat{f}(x_0, \ldots, x_{n-1})$ which we can then evaluate at any point in the parameter space. The fitting algorithm is repeated for both DALYs and Cost so that we have a surrogate model for each output quantity of interest.

4.1.2. Application. We apply the above fitting procedure to fit two separate functions. Recall that $\mathbf{x}$ is an n-dimensional vector representing the parameters controlling the intervention policies. The first function is for DALYs. We will refer to that function with $\hat{f}_{DALY}(\mathbf{x})$. The second function is for Cost. We will refer to that function as $\hat{f}_{Cost}(\mathbf{x})$. As will be seen in Section 5, we managed to construct high-dimensional surrogate models for $n = 42$ corresponding to 5 different intervention policies. We believe that the proposed fitting algorithm can scale up to $n > 100$.

4.2. Optimization Algorithm. As part of our proposed algorithm, we require an optimization algorithm to find hypothesized optima. We will use Newton's algorithm (a gradient-based approach) for optimization.

Recall the constrained optimization problem:

$$\begin{aligned} \min_{x_0, x_1 \cdots x_{n-1}} \quad & \text{DALYs}(x_1, x_2, \cdots x_{n-1}) \\ \text{s.t.} \quad & \text{Cost}(x_1, x_2, \cdots x_{n-1}) < \text{Budge} = \text{B} \\ \text{and} \quad & lb < \mathbf{x} < ub \end{aligned} \tag{4.2}$$

Where lb and ub are lower and upper bounds on the value of $\mathbf{x}$, and B is an upper bound on the budget. We use a system of Lagrange multipliers to transform this constrained optimization into an unconstrained optimization with penalty terms. We will use three penalty terms:

$$\text{Lower Bound Penalty} = LBP(\mathbf{x}, \lambda_0) = \sum_{i=0}^{n-1} \exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)$$

$$\text{Upper Bound Penalty} = UBP(\mathbf{x}, \lambda_1) = \sum_{i=0}^{n-1} \exp\left(\frac{1}{\lambda_1}(x_i - ub)\right)$$

$$\text{Budget Penalty} = BP(\mathbf{x}, \lambda_2) \quad = \exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right)$$

Where λ_0, λ_1, and λ_2 are Lagrange multipliers. These values are chosen based upon the tolerance allowed for each relevant constraint. The lower the value of λ, the larger the penalty upon a violation of the constraint. The exact values of lb, ub, B, λ_0, λ_1, and λ_2 will be detailed in the results section. Our optimization algorithm will therefore aim to solve the following problem:

$$\min_{\mathbf{x}} \quad U(\mathbf{x}, \boldsymbol{\lambda_0}, \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}) = \hat{f}_{\text{DALY}}(\mathbf{x}) + LBP(\mathbf{x}, \lambda_0) + UBP(\mathbf{x}, \lambda_1) + BP(\mathbf{x}, \lambda_2) \tag{4.3}$$

Recall that Newton's algorithm is a gradient-based algorithm where the solution at every step is incrementally updated using the following update function:

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell - H^{-1}\mathbf{j}$$

Where ℓ is the index of the iteration, $\mathbf{x}_\ell$ is the current estimate of the solution, $\mathbf{x}_{\ell+1}$ is the updated guess to the solution, H^{-1} is the inverse of the Hessian matrix, and $\mathbf{j}$ is the Jacobian. The Jacobian is a vector of length n and the Hessian is a matrix of size $n \times n$. The elements of the Jacobian are:

$$\mathbf{j}_i = \frac{\partial U}{\partial \mathbf{x}_i}(\mathbf{x}_\ell)$$

The elements of the Hessian are:

$$H_{ij} = \frac{\partial^2 U}{\partial \mathbf{x}_i \partial \mathbf{x}_j}(\mathbf{x}_\ell)$$

Detailed formulas are included in Appendix A. We stop once our update of $\mathbf{x}$ becomes smaller than a certain threshold. Please note that one of the main advantages of using a tensor train-based surrogate model is that it is very easy to compute the derivatives.

5. Results. In this section, we will discuss the application of our algorithm to two separate problems. First, on identifying optimal intervention policies to for HIV in South Africa. Second, on identifying optimal intervention policies to for HIV in Kenya. The first problem will be referred to as South Africa, and the second problem will be referred to as Kenya. The South Africa problem is parameterized using 11 parameters, and the Kenya problem is parameterized using 42 parameters. For each of the two problems, we will explain the processes of data gathering, tensor train fitting, and the identified optimal allocation policies.

To run each simulation, we use a config file calibrated by Prof. Kim in the Bershteyn Lab group [1]. As mentioned above, the config files allow us to accurately simulate HIV spread within their respective countries. The parameters that we change only control the intervention policies, but do not affect population demographics or HIV behavior. Please note that some of the changes we do to the parameters may be hard to implement in real life. That does not undermine the validity of the final results, but slightly increases the difficulty of a real-world implementation. With that being said, our algorithm is flexible enough to allow tighter constrains on the parameters in order to better mimic real-life constraints. We leave this refinement up to the public health experts.

5.1. HIV in South Africa.

5.1.1. Parameter Descriptions. We identified 11 parameters which controlled the coverage of intervention policies: 6 for ART, 1 for VMMC, 1 for PrEP, and 3 for HCT.

These parameters are listed in Table 1:

Table 1
List of Parameters for South Africa Model Healthcare System

Parameter Name	Parameter Description	Default Value	Range
PresProb	Prob. a symptomatic agent gets HIV test	0.94	[0,1]
Child_6w	Prob. 6-week-old child gets HIV test	0.337	[0,1]
TestUptake	Increased rate of testing after 2009	0.9	[0,1]
Staging	Prob. agent begins ART Staging	0.837	[0,1]
PreART	Prob. agent continues PreART	0.75	[0,1]
FastART	Prob. agent begins ART after 2016	0.95	[0,1]
On_ART	Prob. agent who re-enrolls for ART	0.9	[0,1]
KeepART	Prob. agent keeps using ART after 2016	0.9	[0,1]
ARTInterrupted	Prob. ART is interrupted	0.2	[0,0.2]
PrEP	Prob. HIV negative agent begins to use PrEP	0.15	[0,1]
VMMC	VMMC coverage over the male population	0.8	[0,1]

5.1.2. Surrogate Model Fitting. We will start by fitting a surrogate model. To do that, we start our process by running simulations on 1000 randomly sampled points in the parameter space, and generating the corresponding Cost and DALY outputs. Then we fit two surrogate models; one for Cost and one for DALYs.

We choose the same specifications for both surrogate models, namely, rank $r_i = 5$ and

order $k = 4$, yielding a third-order polynomial approximation.

Our choices of 1000 samples and the rank and order of approximation has been determined based on both available computational resources and sensitivity analyses to ensure robustness of our results. A rank of 5 and third-order polynomials produced the most accurate model while avoiding overfitting to the available data.

We defined the surrogate model error as the L_2 norm of the difference of the sampled points from the training data divided by the L_2 norm of the training data:

$$\text{Error} = \frac{||\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})||}{||\mathbf{f}(\mathbf{x})||}$$

The model achieved an error of under 1% for both Cost and DALY. The error was 0.29% for the Cost model, and 0.91% for the DALY model. The surrogate typically took 2 minutes to train on 50 iterations of 1000 training points, and evaluated a single data point in under 100 ms. This represents a run time close to 4 orders of magnitude faster than the base EMOD model.

5.1.3. Optimal Allocation. We implemented the proposed algorithm as described in the previous section, with a cost constraint of $1M. The Cost and DALY combinations generated, compared to those generated randomly, are found in Figure 1.

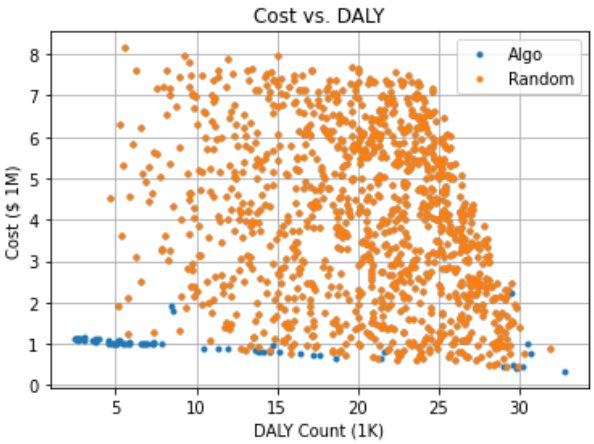


Figure 1. *Graph of Cost vs DALYs for both Random and Chosen Points*

The above graph shows 1000 randomized points in orange, comparing the cost for each simulation with the DALY count for that simulation. In blue are the points computed with our algorithm. The most notable feature is the 'tail' of blue points on the bottom left of the graph, which represents the convergent behavior of the algorithm: all these points carry a low DALY measure and are very close to the cost constraint of $1M. The scattered points across the bottom and right represent points where our surrogate models made certain inaccurate predictions. In those cases, our algorithm was able to take advantage of these inaccuracies and use them to reinforce the accuracy of our surrogates.

We wanted to evaluate the effect of increased budget on DALY Count and optimal parameters of the intervention policies. To achieve that, we changed the allotted budget from \$1M to \$3M in steps of \$500K. The results of these runs are shown in Figure 2.

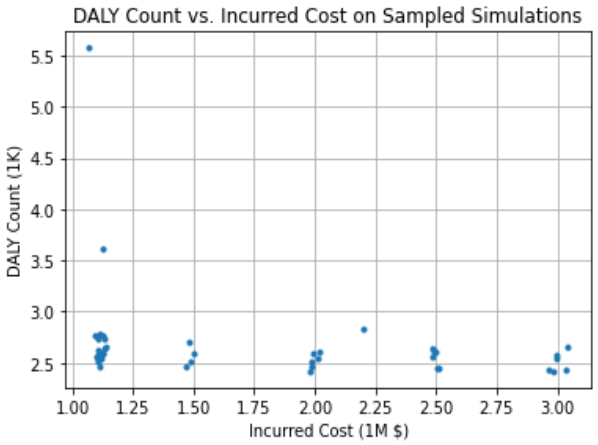


Figure 2. *Graph of Cost vs DALYs for Varied Budgets*

We have multiple observations. The DALY count was relatively stable, ranging from 2416 to 2470, a small variance relative to the possible range of DALYs. For small budgets, ART is by far the most cost-effective policy, while PrEP remains at zero. For larger budgets, PrEP becomes more cost-effective. One surprising observation is that our algorithm sometimes terminates at optimal points that do not use all the allowed budget. For example, even though we set a budget of \$3M, the algorithm has terminated at both \$1.1M and \$2M. We studied this observation and found that there are 5 different local minima at which the algorithm terminates. In Table 2, we give the parameter set yielding the best point within each cluster and the corresponding DALY count and Cost.

It seems that ART is the most heavily emphasized intervention, with 3 different parameters set to 1 between all 5 clusters. HCT is a close second, with 2 parameters set to 1 between all 5 clusters. This may be due to the way they work within the simulation. ART is a medication taken by those with HIV which extends their lifespan while preventing them from passing HIV on to other people. This has a dual-effect in reducing DALYs, limiting the DALYs created from their own deaths while limiting the DALYs created from the deaths of others. Testing works by taking those who may not know they have HIV, and putting them in a position where they have the option to get treated. It is also relatively inexpensive compared to the other interventions, meaning that, even with a lower reduction in DALYs than other medications, it can still be more cost-effective as it is cheaper.

PrEP, on the other hand, begins with 0 investment, and slowly increases as the total amount of money spent increases. PrEP is a preventative, taken by those without HIV, which protects them from catching HIV. This has an impact compared to other medications only if the number of people *catching* HIV is creating more DALYs than those already with HIV. PrEP is also expensive, meaning that it needs to have a very large impact in order to be cost-effective. It seems like this is not the environment of the South Africa model.

Table 2
Parameter and Output Values for South Africa Model Clusters

Parameter Name	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
KeepART	1	1	1	1	1
TestUptake	1	1	1	1	1
FastART	0.423	0.567	0.623	0.003	0.304
PresProb	1	1	1	1	1
Child_6w	1	0.743	0.31	0.802	0.998
Staging	1	0.979	1	1	1
PreART	1	1	1	1	1
On_ART	1	1	1	1	1
VMMC	0.016	1	0.64	1	0.003
PrEP	0	0.05	0.12	0.186	0.247
ARTInterrupted	0.2	0.129	0.071	0.035	0
Cost	$1,113,055.47	$1,465,936.00	$1,978,418.95	$2,513,331.67	$2,985,608.23
DALY	2,470.75	2,461.45	2,417.12	2,442.63	2,416.71

5.2. HIV in Kenya.

5.2.1. Parameter Descriptions. In the Kenya disease model, there exist 42 parameters which control the available interventions: 6 parameters for ART, 18 parameters for VMMC, 12 parameters for PrEP, 4 parameters for HCT, and 2 parameters for ANC.

5.2.2. Surrogate Model Fitting. As with the South Africa problen, we fit two surrogate models, one for Cost and one for DALYs. The surrogate models were fit with 1000 randomly generated points, which took EMOD 2 days to evaluate. Initially, we chose rank $r_i = 5$ and order $k = 4$, yielding a third-order polynomial approximation with matrix coefficients of shape 5×5. Upon beginning optimization using Newton's method, this tensor train structure had too many unknown coefficients to solve for, as compared to the surrogate models used in the South Africa example. This led to overfitting the surrogate models to the available data, meaning that the surrogate's projections became unreliable. Outputs ranged from -10^{12} to 10^{12} and were not accurate to EMOD. As a result, r_i was reduced to 3, and k was reduced to 2, yielding a linear function for each variable, with coefficients matrices of shape 2×2. This greatly reduced the number of free variables, mitigating the overfitting.

5.2.3. Optimal Allocation. We implemented the proposed algorithm as described in the previous section, with a cost constraint of $3M. The Cost and DALY combinations computed by the framework, compared to those generated randomly, are found in Figure 3.

As before, there are some points dispersed in suboptimal areas with high DALY counts and varied Costs. These are points where the framework has predicted inaccurately, and is able to update its predictions based on the real EMOD value. Notably, there is a thick cluster of points ranging between $2M budget and $3M budget with low DALY counts, ranging from the low thousands up to $10k$ DALYs. This region is highlighted in Figure 4.

The most optimal points range in DALY counts from 1800 to 2500, a value significantly

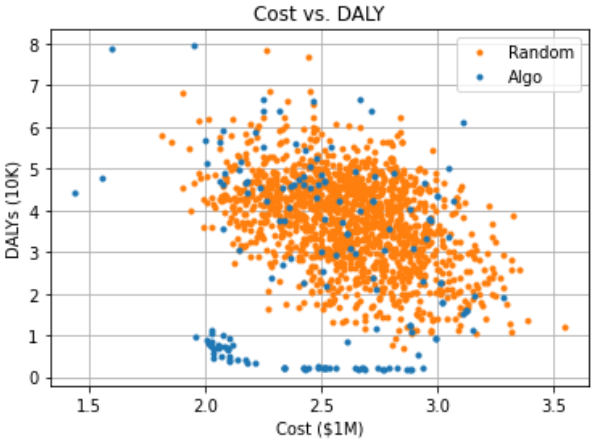


Figure 3. *Graph of Cost vs DALYs for Varied Budgets*

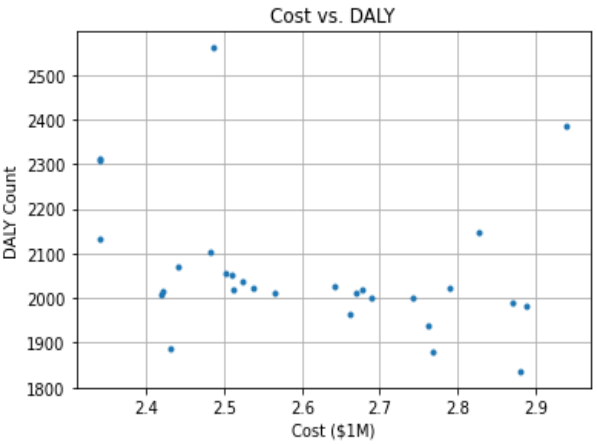


Figure 4. *Graph of Cost vs DALYs for Optimal Simulations in Kenya*

lower than the randomly generated simulations. Optimal Cost also ranges between $2.4M to $3M, well within our allowed budget of $3M. Notably, there are no clusters within the located points. There is only a scattering of lower values spread out fairly evenly within the area. Importantly, while all of these points have low DALY counts, the predictions that the surrogate models made were not accurate within this region, as shown in Figure 5.

The surrogate models predicted DALY values in the 10k range, but EMOD actually returned DALY values near 2k DALYs. This is most likely because we used a linear model for each term in the tensor train. Compared to cubic polynomials, which we were using for the South Africa problem, linear polynomials have less ability to model complex behavior of a single parameter. They would tend to model the bulk of the data while ignoring the fringe behavior. In our case, the DALY values of the points used for fitting range from 10k DALYs to 70K DALYs. It could make sense for a model with relatively few parameters to stick

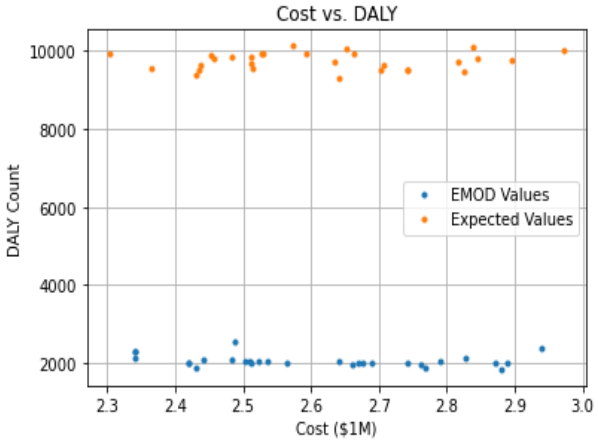


Figure 5. *Graph of Cost vs DALYs for Surrogate Projections Compared to EMOD Values in Kenya*

to the bulk of the data, precisely this range. Points with low DALY counts are abnormal, which could explain why the surrogate models drastically overestimate the DALY count of low-DALY points.

It should be remembered that these points, while having high predicted DALYs (as predicted by the surrogate), still had low DALYs when evaluated by EMOD. This is probably due to the fact that a gradient-based optimization algorithm was used. The algorithm was able to accurately traverse the surrogate models to a place of relative minimum, and although the actual value of this relative minimum was overestimated, it was a relative minimum nonetheless, in the surrogate model's estimation and within EMOD as well.

The Cost within this range is predicted accurately, which would align with our explanation. The Cost of the randomly generated points ranges from \$2M to \$3.25M, and our predicted Costs landed within the middle of this range. A model with few degrees of freedom may be able to predict Costs accurately if they fall within the range of the Costs used to fit the model.

6. Conclusions & Future Work. In this project, we were able to construct a framework for HIV intervention optimization which greatly reduced the number of DALYs computed within the EMOD simulation tool. Our framework is able to adapt to various price points and recommend optimal investment strategies for each one. Our framework is also self-improving, in that it was able to either locate an optima with respect to the main EMOD model, and if not, it improved upon its accuracy and predictive ability. This framework is highly customizable in terms of metaparameters and since it is not specific to HIV models, it can be applied to a variety of optimization problems. It is powerful enough to solve high-dimensional problems ($d \geq 40$), and reports a variety of results to the end user. It uses a gradient-based optimization algorithm which takes advantage of the surrogate model's differentiability and low evaluation time. It can be fit to already-existent data.

In terms of optimal investment, it seems that ART and HCT are the most cost effective interventions which should be prioritized on a limited budget. Preventative medications

like PrEP, however, should be skipped out when strapped for cash, but as available funding increases, investment should increase.

This work can be extended in a number of ways. First and foremost, it should be taken by a public health researcher and applied to a realistic example of HIV. In this project, the parameters used for each application represented the maximum possible reach over each country's healthcare system. In reality, South Africa may be less able to give ART medication and more able to give HCT kits. This would be represented by bounds on the values of the corresponding parameters easily inputted into the existing software. Additionally, this framework should be tested on other models and optimization problems. It is not specific to HIV models, or disease models as a whole, and is instead general. It can be applied to any form of optimization problem, and should be tested as such.

7. Acknowledgements. I would like to thank my mentor, my parents, and my science research teacher.

REFERENCES

[1] A. Bershteyn, J. Gerardin, D. Bridenbecker, C. W. Lorton, J. Bloedow, R. S. Baker, G. Chabot-Couture, Y. Chen, T. Fischle, K. Frey, J. S. Gauld, H. Hu, A. S. Izzo, D. J. Klein, D. Lukacevic, K. A. McCarthy, J. C. Miller, A. L. Ouedraogo, T. A. Perkins, J. Steinkraus, Q. A. ten Bosch, H.-F. Ting, S. Titova, B. G. Wagner, P. A. Welkhoff, E. A. Wenger, C. N. Wiswell, and for the Institute for Disease Modeling, *Implementation and applications of EMOD, an individual-based multi-disease modeling platform*, Pathogens and Disease, 76 (2018), https://doi.org/10.1093/femspd/fty059, https://doi.org/10.1093/femspd/fty059, https://arxiv.org/abs/https://academic.oup.com/femspd/article-pdf/76/5/fty059/25336697/fty059.pdf. fty059.

[2] A. Bershteyn, M. Sharma, A. Akullian, K. Peebles, S. Sarkar, R. Braithwaite, and E. Mudimu, *Impact along the hiv pre-exposure prophylaxis "cascade of prevention" in western kenya: a mathematical modelling study*, Journal of the International AIDS Society, 23 (2020), https://doi.org/10.1002/jia2.25527.

[3] D. Bigoni, A. P. Engsig-Karup, and Y. M. Marzouk, *Spectral tensor-train decomposition*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2405–A2439, https://doi.org/10.1137/15m1036919, https://doi.org/10.1137%2F15m1036919.

[4] A. Gorodetsky, S. Karaman, and Y. Marzouk, *A continuous analogue of the tensor-train decomposition*, Computer Methods in Applied Mechanics and Engineering, 347 (2019), pp. 59–84, https://doi.org/https://doi.org/10.1016/j.cma.2018.12.015, https://www.sciencedirect.com/science/article/pii/S0045782518306133.

[5] C. C. Kerr, R. M. Stuart, R. T. Gray, A. J. Shattock, N. Fraser-Hurt, C. Benedikt, M. Haacker, M. Berdnikov, A. M. Mahmood, S. A. Jaber, M. Gorgens, and D. P. Wilson, *Optima*, JAIDS Journal of Acquired Immune Deficiency Syndromes, 69 (2015), pp. 365–376, https://doi.org/10.1097/qai.0000000000000605, https://doi.org/10.1097%2Fqai.0000000000000605.

[6] A. B. Kharsany and Q. A. Karim, *HIV infection and AIDS in sub-saharan africa: Current status, challenges and opportunities*, The Open AIDS Journal, 10 (2016), pp. 34–48, https://doi.org/10.2174/1874613601610010034, https://doi.org/10.2174%2F1874613601610010034.

[7] B. Lukoba, J. Simiyu, W. Chege, S. Kelly, M. Minnery, L. Sithole, and Z. Shubber, *Improving allocative efficiency of the hiv response in kenya: A country-level analysis using the optima hiv model*, report, The World Bank, March 2021.

[8] G. Meyer-Rath, C. van Rensburg, C. Chiu, R. Leuner, L. Jamieson, and S. Cohen, *The per-patient costs of HIV services in south africa: Systematic review and application in the south african*

HIV investment case, PLOS ONE, 14 (2019), p. e0210497, https://doi.org/10.1371/journal.pone.0210497, https://doi.org/10.1371%2Fjournal.pone.0210497.

[9] J. U. N. P. on HIV/AIDS, *UNAIDS DATA 2021*, UNAIDS, Geneva, 2021.

[10] I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286, https://doi.org/10.1137/090752286, https://arxiv.org/abs/https://doi.org/10.1137/090752286.

[11] M. Sharma, E. Mudimu, K. Simeon, A. Bershteyn, J. Dorward, L. R. Violette, A. Akullian, S. S. A. Karim, C. Celum, N. Garrett, and P. K. Drain, *Cost-effectiveness of point-of-care testing with task-shifting for HIV care in south africa: a modelling study*, The Lancet HIV, 8 (2021), pp. e216–e224, https://doi.org/10.1016/s2352-3018(20)30279-4, https://doi.org/10.1016%2Fs2352-3018%2820%2930279-4.

[12] United Nations: Department of Economic and Social Affairs, *World mortality 2019*, United Nations, New York, NY, Jan. 2020.

[13] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis, and L. Daniel, *Enabling high-dimensional hierarchical uncertainty quantification by anova and tensor-train decomposition*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34 (2015), pp. 63–76, https://doi.org/10.1109/TCAD.2014.2369505.

Appendix A. Tensor Train Differentiation.

Our function to minimize was:

$$\text{Lower Bound Penalty} = LBP(\mathbf{x}) = \sum_{i=0}^{n-1} \exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)$$

$$\text{Upper Bound Penalty} = UBP(\mathbf{x}) = \sum_{i=0}^{n-1} \exp\left(\frac{1}{\lambda_1}(x_i - ub)\right)$$

$$\text{Budget Penalty} = BP(\mathbf{x}) \quad = \exp\left(\frac{1}{\lambda_2}(\hat{f}_{Cost}(\mathbf{x}) - B)\right)$$

$$U(\mathbf{x}) = \hat{f}_{DALY}(\mathbf{x}) + LBP(\mathbf{x}) + UBP(\mathbf{x}) + BP(\mathbf{x})$$

To apply Newton's method to this function, we must calculate the vector of first derivatives, the Jacobian vector, and the matrix of second derivatives, the Hessian matrix. The elements of the Jacobian are:

$$\mathbf{j}_i = \frac{\partial U}{\partial \mathbf{x}_i}(\mathbf{x})$$

The elements of the Hessian are:

$$H_{ij} = \frac{\partial^2 U}{\partial \mathbf{x}_i \partial \mathbf{x}_j}(\mathbf{x})$$

Therefore, we must calculate take the first and second derivatives of $\hat{f}(\mathbf{x})$, $LBP(\mathbf{x})$, $UBP(\mathbf{x})$, and $BP(\mathbf{x})$. As a product of univariate functions, particularly polynomials, the tensor train is highly differentiable. Recall:

$$\hat{f}(\mathbf{x}) = G_0^T(x_0) \cdots G_i(x_i) \cdots G_{n-1}(x_{n-1})$$

$$\begin{aligned}\frac{\partial \hat{f}}{\partial x_i}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\left(G_0^T(x_0) \cdots G_i(x_i) \cdots G_{n-1}(x_{n-1})\right) \\ &= G_0^T(x_0) \cdots \frac{\partial G_i}{\partial x_i}(x_i) \cdots G_{n-1}(x_{n-1})\end{aligned}$$

Given that $G_i(x_i) = \sum_{j=0}^{k} b_j(x_i) C_{ij}$,

$$\frac{\partial G_i}{\partial x_i}(x_i) = \sum_{j=0}^{k} \frac{\partial}{\partial x_i}(b_j(x_i)) C_{ij}$$

We are using Taylor polynomials of the form $b_j(x_i) = \sum_{n=0}^{j} \frac{x_i^n}{n!}$, so $\frac{\partial b_j}{\partial x_i}(x_i) = b_{j-1}(x_i)$, where $b_j(x_i) = 0$ for $j < 0$. Therefore:

$$\frac{\partial G_i}{\partial x_i}(x_i) = \sum_{j=0}^{k-1} b_j(x_i) C_{ij}$$

$$\frac{\partial \hat{f}}{\partial x_i}(\mathbf{x}) = G_0^T(x_0) \cdots \left(\sum_{j=0}^{k-1} b_j(x_i) C_{ij} \right) \cdots G_{n-1}(x_{n-1})$$

A similar procedure can be applied to the second partial derivatives of tensor trains:

$$\begin{aligned}
\frac{\partial^2 \hat{f}}{\partial x_i^2}(\mathbf{x}) &= G_0^T(x_0) \cdots \frac{\partial^2 G_i}{\partial x_i^2}(x_i) \cdots G_{n-1}(x_{n-1}) \\
&= G_0^T(x_0) \cdots \sum_{j=0}^{k} \frac{\partial^2}{\partial x_i^2}(b_j(x_i)) C_{ij} \cdots G_{n-1}(x_{n-1}) \\
&= G_0^T(x_0) \cdots \left(\sum_{j=0}^{k-2} (b_j(x_i)) C_{ij} \right) \cdots G_{n-1}(x_{n-1})
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \hat{f}}{\partial x_i \partial x_j}(\mathbf{x}) &= G_0^T(x_0) \cdots \frac{\partial G_i}{\partial x_i}(x_i) \cdots \frac{\partial G_j}{\partial x_j}(x_j) \cdots G_{n-1}(x_{n-1}) \\
&= G_0^T(x_0) \cdots \left(\sum_{j=0}^{k-1} (b_j(x_i)) C_{ij} \right) \cdots \left(\sum_{m=0}^{k-1} (b_m(x_j)) C_{jm} \right) \cdots G_{n-1}(x_{n-1})
\end{aligned}$$

The derivatives for the Lower Bound Penalty are as follows:

$$\begin{aligned}\frac{\partial LBP}{\partial x_i}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\sum_{m=0}^{n-1}\exp\left(\frac{1}{\lambda_0}(lb - x_m)\right)\\ &= -\frac{1}{\lambda_0}\exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 LBP}{\partial x_i^2}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\left(\frac{\partial LBP}{\partial x_i}(\mathbf{x})\right)\\ &= \frac{\partial}{\partial x_i}\left(-\frac{1}{\lambda_0}\exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)\right)\\ &= \frac{1}{\lambda_0^2}\exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 LBP}{\partial x_i \partial x_j}(\mathbf{x}) &= \frac{\partial}{\partial x_j}\left(\frac{\partial LBP}{\partial x_i}(\mathbf{x})\right)\\ &= \frac{\partial}{\partial x_j}\left(-\frac{1}{\lambda_0}\exp\left(\frac{1}{\lambda_0}(lb - x_i)\right)\right)\\ &= 0\end{aligned}$$

The derivatives for the Upper Bound Penalty are as follows:

$$\begin{aligned}\frac{\partial UBP}{\partial x_i}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\sum_{m=0}^{n-1}\exp\left(\frac{1}{\lambda_1}(x_m - ub)\right)\\ &= \frac{1}{\lambda_1}\exp\left(\frac{1}{\lambda_1}(x_m - ub)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 UBP}{\partial x_i^2}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\left(\frac{\partial UBP}{\partial x_i}(\mathbf{x})\right)\\ &= \frac{\partial}{\partial x_i}\left(\frac{1}{\lambda_1}\exp\left(\frac{1}{\lambda_1}(x_m - ub)\right)\right)\\ &= \frac{1}{\lambda_1^2}\exp\left(\frac{1}{\lambda_1}(x_m - ub)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial^2 UBP}{\partial x_i \partial x_j}(\mathbf{x}) &= \frac{\partial}{\partial x_j}\left(\frac{\partial UBP}{\partial x_i}(\mathbf{x})\right)\\ &= \frac{\partial}{\partial x_j}\left(\frac{1}{\lambda_1}\exp\left(\frac{1}{\lambda_1}(x_m - ub)\right)\right)\\ &= 0\end{aligned}$$

Finally, the derivatives for the Budget Penalty are as follows:

$$\begin{aligned}
\frac{\partial BP}{\partial x_i}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right) \\
&= \frac{1}{\lambda_2}\frac{\partial \hat{f}_{\text{Cost}}}{\partial x_i}(\mathbf{x})\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right) \\
\frac{\partial^2 BP}{\partial x_i^2}(\mathbf{x}) &= \frac{\partial}{\partial x_i}\left(\frac{\partial BP}{\partial x_i}(\mathbf{x})\right) \\
&= \frac{1}{\lambda_2}\frac{\partial^2 \hat{f}_{\text{Cost}}}{\partial x_i^2}(\mathbf{x})\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right) + \frac{1}{\lambda_2^2}\left(\frac{\partial \hat{f}_{\text{Cost}}}{\partial x_i}\right)^2\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right) \\
\frac{\partial^2 BP}{\partial x_i \partial x_j}(\mathbf{x}) &= \frac{\partial}{\partial x_j}\left(\frac{\partial BP}{\partial x_i}\right) \\
&= \frac{1}{\lambda_2}\frac{\partial^2 \hat{f}_{\text{Cost}}}{\partial x_i x_j}(\mathbf{x})\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right) + \frac{1}{\lambda_2^2}\left(\frac{\partial \hat{f}_{\text{Cost}}}{\partial x_i}\right)\left(\frac{\partial \hat{f}_{\text{Cost}}}{\partial x_j}\right)\exp\left(\frac{1}{\lambda_2}(\hat{f}_{\text{Cost}}(\mathbf{x}) - B)\right)
\end{aligned}$$

The total Jacobian and Hessian can be formed by taking the sum of the relevant terms:

$$\begin{aligned}
\mathbf{j}_i &= \frac{\partial U}{\partial x_i}(\mathbf{x}) \\
&= \frac{\partial \hat{f}_{\text{DALY}}}{\partial x_i}(\mathbf{x}) + \frac{\partial LBP}{\partial x_i}(\mathbf{x}) + \frac{\partial UBP}{\partial x_i}(\mathbf{x}) + \frac{\partial BP}{\partial x_i}(\mathbf{x})
\end{aligned}$$

$$\begin{aligned}
H_{ii} &= \frac{\partial^2 U}{\partial x_i^2}(\mathbf{x}) \\
&= \frac{\partial^2 \hat{f}_{\text{DALY}}}{\partial x_i^2}(\mathbf{x}) + \frac{\partial^2 LBP}{\partial x_i^2}(\mathbf{x}) + \frac{\partial^2 UBP}{\partial x_i^2}(\mathbf{x}) + \frac{\partial^2 BP}{\partial x_i^2}(\mathbf{x})
\end{aligned}$$

$$\begin{aligned}
H_{ij} &= \frac{\partial^2 U}{\partial x_i \partial x_j}(\mathbf{x}) \\
&= \frac{\partial^2 \hat{f}_{\text{DALY}}}{\partial x_i \partial x_j}(\mathbf{x}) + \frac{\partial^2 LBP}{\partial x_i \partial x_j}(\mathbf{x}) + \frac{\partial^2 UBP}{\partial x_i \partial x_j}(\mathbf{x}) + \frac{\partial^2 BP}{\partial x_i \partial x_j}(\mathbf{x})
\end{aligned}$$