STATISTICS IN BIOPHARMACEUTICAL RESEARCH

Statistics in Biopharmaceutical Research

ISSN: (Print) 1946-6315 (Online) Journal homepage: https://www.tandfonline.com/loi/usbr20

Optimizing Graphical Procedures for Multiplicity Control in a Confirmatory Clinical Trial via Deep Learning

Tianyu Zhan, Alan Hartford, Jian Kang & Walter Offen

STATISTICS IN BIOPHARMACEUTICAL RESEARCH
2020, VOL. 00, NO. 0, 1–11
https://doi.org/10.1080/19466315.2020.1799855

Taylor & Francis
Taylor & Francis Group

Optimizing Graphical Procedures for Multiplicity Control in a Confirmatory Clinical Trial via Deep Learning

Tianyu Zhan[a], Alan Hartford[b], Jian Kang[c], and Walter Offen[d]

[a]Data and Statistical Sciences, AbbVie Inc., North Chicago, IL; [b]Statistical and Quantitative Sciences, Data Sciences Institute, Research and Development, Takeda Pharmaceuticals USA, Inc., Cambridge, MA; [c]Department of Biostatistics, University of Michigan, Ann Arbor, MI; [d]Retired from AbbVie Inc.

ABSTRACT

In confirmatory clinical trials, it has been proposed to use a simple iterative graphical approach to construct and perform intersection hypotheses tests with a weighted Bonferroni-type procedure to control Type I errors in the strong sense. Given Phase II study results or other prior knowledge, it is usually of main interest to find the optimal graph that maximizes a certain objective function in a future Phase III study. In this article, we evaluate the performance of two existing derivative-free constrained methods, and further propose a deep learning enhanced optimization framework. Our method numerically approximates the objective function via feedforward neural networks (FNNs) and then performs optimization with available gradient information. It can be constrained so that some features of the testing procedure are held fixed while optimizing over other features. Simulation studies show that our FNN-based approach has a better balance between robustness and time efficiency than some existing derivative-free constrained optimization algorithms. Compared to the traditional stochastic search method, our optimizer has moderate multiplicity adjusted power gain when the number of hypotheses is relatively large. We further apply it to a case study to illustrate how to optimize a multiple testing procedure with respect to a specific study objective.

ARTICLE HISTORY
Received January 2020
Accepted July 2020

KEYWORDS
Clinical trial optimization;
Constrained optimization;
Deep neural network;
Graphical approach;
Family-wise error rate control

1. Introduction

Most clinical trials performed in drug development contain multiple endpoints to assess the effects of the drug and to document the ability of the drug to favorably affect one or more disease characteristics (Kelly et al. 2015; Kazda et al. 2016; Food and Drug Administration 2017). Adequate multiple testing procedures (MTPs) are required to protect the family-wise error rate (FWER), which is the probability of rejecting at least one true null hypothesis. Proper MTPs should be employed to reflect relative importance of multiple endpoints and different study objectives. A variety of weighted Bonferroni-based test procedures have been proposed, for example, the weighted or unweighted Bonferroni–Holm procedure (Holm 1979), fixed sequence tests (Westfall and Krishen 2001), the fallback procedure (Wiens 2003), and gatekeeping procedures based on Bonferroni adjustments (Dmitrienko, Offen, and Westfall 2003).

Those aforementioned approaches usually need to specify a large number of intersection hypotheses tests according to the closure principle (Marcus, Eric, and Gabriel 1976). It is often difficult to apply those methods in practice, especially when the number of endpoints is relatively large. Taking a study with 10 hypotheses as an example, there are $2^{10}-1 = 1,023$ intersection hypotheses in the full closure. Bretz et al. (2009) proposed a graphical approach to represent a wide range of MTPs with weighted Bonferroni tests for intersection hypotheses. Based on the monotonicity for local significance levels, the graphical approach essentially establishes a shortcut to the closure test procedure and leads to a sequentially rejective procedure with up to m steps, where m is the number of null hypotheses to be tested. The graphical representation of this approach is easier to communicate with clinical teams and facilitates the discussion of different strategies to fulfill distinct study objectives. However, choosing a graph in complex testing situations can still be overwhelming. While practical considerations for achieving the most desired drug label may take precedence over the most efficient graphical testing procedure, the decision of which graphical testing to choose will be served well by being informed of the optimal graph with respect to an objective function.

Since graphical approaches analytically control FWER at a desired level in the strong sense (Bretz et al. 2009), can we further identify an optimal graph in a confirmatory trial with respect to certain objective functions based on prior knowledge such as from Phase II studies? Rubin, Dudoit, and Van der Laan (2006) and Wasserman and Roeder (2006) studied the power function for the weighted Bonferroni procedure, but the graphical approach we considered is more general. As can be seen later in this article, it is difficult to evaluate the objective function and its derivatives in closed forms due to the complex correlations between the decision functions from different endpoints. The stochastic search method (SSM; Zabinsky 2013) is popular in practice due to its ease of implementation. This approximating approach is to find the graph with the maximum working objective function among a certain number of randomly simulated candidates under constraints. However,

CONTACT Tianyu Zhan tianyu.zhan@abbvie.com Data and Statistical Sciences, AbbVie Inc., 1 North Waukegan Road, North Chicago, IL 60064.

this method is very likely to miss the optimal target when the number of endpoints is relatively large, as demonstrated later in this article.

Another stream is to adopt existing derivative-free constrained optimization methods. The ability to handle both bounded and inequality constraints is desired to accommodate different study objectives and the constraints in the graphical approach. There are vast numbers of those approaches available in the field of machine learning, but their performances in finding either global or local optima vary depending on the problem at hand (Kramer, Ciaurri, and Koziel 2011). In this article, we evaluate the performance of the improved stochastic ranking evolution strategy (ISRES; Runarsson and Yao 2005) and the constrained optimization by linear approximations (COBYLA; Powell 1994) on optimizing graphical approaches by simulation studies. They are readily implemented in the R package *nloptr* (Ypma 2018). As an alternative, we propose an optimization framework based on deep learning with moderate power gain and tolerable extra computing time.

Deep learning has made substantial success in various domains such as image recognition and natural language processing (Goodfellow, Bengio, and Courville 2016), and is also receiving attention from the pharmaceutical industry. For example, Liang, Ye, and Fu (2018) proposed a novel outcome weighted deep learning algorithm to estimate individualized optimal combination therapy; and Zhan and Kang (2019) constructed test statistics based on deep neural networks to increase power in sample size reassessment adaptive clinical trials. In this article, we use feedforward neural networks (FNNs) to approximate the underlying complex objective function and further identify the optima with available gradient information. Our method has several distinguishing features. First of all, flexible utility functions can be defined to accommodate different study objectives. Moreover, our method is able to perform optimization when certain structures in the graph are fixed. Additionally, gradients are readily available from the fitted FNN, and do not need to be computed from the complex objective function, which is often not feasible even with numerical methods. Compared with the two derivative-free optimization approaches, our FNN-based optimizer offers a better balance between time efficiency and robustness. More details are provided in Section 4.

The remainder of this article is organized as follows. In Section 2, we review the graphical approach for multiple hypotheses testing and further define the objective function to optimize. In Section 3, we introduce our optimizing methods via deep learning techniques. Simulations under multiple scenarios are conducted to evaluate the performance of our procedures in Section 4. In Section 5, we implement our method in a case study. Finally, concluding remarks are provided in Section 6.

2. The Graphical Approach to Sequentially Rejective Multiple Testing Procedures

In this section, we first review the graphical approach as an MTP which strongly controls the FWER at a nominal level α in Section 2.1. It is essentially a shortcut to the closed testing procedure with the weighted Bonferroni test for intersection hypotheses. In Section 2.2, we introduce an objective function to evaluate the performance of a specific graph.

2.1. Review of the Graphical Approach

Suppose in a clinical trial, we are interested in testing m elementary null hypotheses, $H_1, H_2, \dots, H_m$, with observed unadjusted p-values $\boldsymbol{p} = (p_1, p_2, \dots, p_m)$. Let α denote the one-sided FWER (usually $\alpha = 0.025$ in practice). An MTP is said to control the FWER at α in the strong sense that the probability of rejecting at least one true null hypothesis does not exceed α under any configuration of true and false null hypotheses. The MTPs can be derived from the closure principle (Marcus, Eric, and Gabriel 1976), which requires $2^m - 1$ local α-level tests of each non-empty intersection hypothesis $H(I) = \cap_{i \in I} H_i$, where $I \subseteq M = \{1, 2, \dots, m\}$ (Tamhane and Gou 2018). An intersection hypothesis $H(I)$ is rejected if and only if all $H(J)$ for $J \supseteq I$ are rejected by their α-level tests. As a shortcut, if the local tests are consonant (Gabriel 1969), then the corresponding MTP requires only up to m local tests. For example, the Holm MTP uses Bonferroni tests as the local tests for all intersection hypotheses (Holm 1979).

The graphical approach defines a shortcut MTP for a closed testing procedure with weighted Bonferroni tests for the intersection hypotheses to strongly control the FWER at α. Specifically, the weighted Bonferroni rejects $H(I)$ if $\left\{\min_{i \in I} \left(p_i / w_i\right)\right\} \leq \alpha$, where $\sum_{i=1}^{|I|} w_i = 1$ and $|I|$ denotes the number of elements in I. To specify a graph, one needs to define two components: the initial alpha allocation vector $\boldsymbol{\alpha}$ and the transition matrix $\boldsymbol{T}$. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ denote the initial assignment of overall significance level under the constraint,

$$\sum_{i=1}^{m} \alpha_i = \alpha. \qquad (1)$$

Note that the equality sign in (1) is to make full use of all available significance levels to gain the highest power. It can be replaced by the sign "$\leq$" while still controlling FWER at α. The transition matrix $\boldsymbol{T}$ is an $m \times m$ matrix, where each element T_{ij} specifies the proportion of local significance level α_i that is passed to H_j if H_i is rejected at α_i. For all $i, j = 1, 2, \dots, m$, T_{ij} has to satisfy the following conditions:

$$0 \leq T_{ij} \leq 1, \qquad T_{ii} = 0, \qquad \sum_{k=1}^{m} T_{ik} = 1. \qquad (2)$$

We further use $g(\boldsymbol{\alpha}, \boldsymbol{T})$ to denote a graph with vector $\boldsymbol{\alpha}$ and matrix $\boldsymbol{T}$. The graphical approach $g(\boldsymbol{\alpha}, \boldsymbol{T})$ can represent a variety of weighted Bonferroni-based test procedures.

Consider a motivating example of a Phase III clinical trial with two doses (high and low) and two endpoints (primary and secondary) in each dose. The team may want to consider a design represented by the graphical procedure in Figure 1. One first tests the primary endpoint in each dose with $0.5 \times \alpha$; 80% of it will be passed to the secondary endpoint and 20% to the primary endpoint in the other dose if rejected. Once rejected, the significance level of the secondary endpoint can also be fully recycled to the primary endpoint in the alternative dose. In this

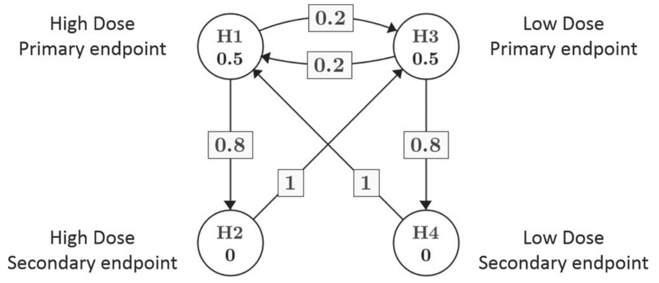


Figure 1. A motivating example of a graphical approach for multiplicity control of two doses and two endpoints.

case, the initial alpha allocation vector $\boldsymbol{\alpha}$ is $(0.0125, 0, 0.0125, 0)$ and the transition matrix $\boldsymbol{T}$ is given by

$$\boldsymbol{T} = \begin{pmatrix} 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 1 & 0 \\ 0.2 & 0 & 0 & 0.8 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Given the observed unadjusted p-value vector $\boldsymbol{p}$, the graphical approach establishes a sequentially rejective test procedure that is illustrated in Algorithm 1. Basically, one tests the most significant hypothesis with its nonzero local significance level. If it is rejected, then update the graph according to the prespecified rules. We further define a decision function $D_i\left(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}\right)$ for endpoint i, which takes value 1 if its null hypothesis is rejected under a graphical approach $g(\boldsymbol{\alpha}, \boldsymbol{T})$, and 0 otherwise.

Algorithm 1 Graphical approaches (Bretz et al. 2009)

```
0. Set $I = M$.
1. Let $j = \text{argmin}_{i \in I}\ p_i/\alpha_i$.
2. If $p_j \le \alpha_j$, then reject $H_j$; otherwise stop.
3. Update the graph:
    $I \to I \setminus \{j\}$
    $\alpha_l = \begin{cases} \alpha_l + \alpha_j T_{jl}, & l \in I \\ 0, & \text{otherwise} \end{cases}$
    $T_{lk} = \begin{cases} \dfrac{T_{lk} + T_{lj}T_{jk}}{1 - T_{lj}T_{jl}}, & l, k \in I,\ l \neq k \\ 0, & \text{otherwise} \end{cases}$
4. If $|I| \ge 1$, go to step 1; otherwise stop.
```

Since all graphs under constraints (1) and (2) and defined by Algorithm 1 control FWER at α in the strong sense, then a natural question for drug development is how to obtain the optimal one based on the results from a previous study. Before diving into this optimization problem, we first define an objective function to evaluate different graphs in the following section.

2.2. An Objective Function to Evaluate Performance

Remember that in the previous section, we use $\boldsymbol{p}$ to denote the unadjusted p-value vector for m endpoints. Given this underlying multivariate data-generating mechanism, we further define an objective function $O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$ to measure the performance of a graphical procedure with initial alpha vector $\boldsymbol{\alpha}$ and transition matrix $\boldsymbol{T}$,

$$O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right) = \sum_{i=1}^{m} v_i \, \mathrm{E}_{\boldsymbol{p}}\left\{D_i\left(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}\right)\right\}, \qquad (3)$$

where the expectation is with respect to the multivariate distribution of $\boldsymbol{p}$, and v_i is prespecified to represent the relative importance of endpoint i with the constraint $\sum_{i=1}^{m} v_i = 1$. As a starting point, we focus on the objective function defined in (3) for illustration. In the case study in Section 5, we generalize this objective function to be more clinically meaningful based on the study's objective. We denote the stack of v_i's with the vector $\boldsymbol{v}$. If $v_i = 1/m$ for all i's, then (3) is interpreted as the average of multiplicity adjusted power from all endpoints. In the motivating example, the team can set $\boldsymbol{v} = (0.4, 0.2, 0.3, 0.1)'$ if they treat H_1 as the most important target.

Let $\mathcal{A}$ denote the parameter space of $\boldsymbol{\alpha}$, and correspondingly $\mathcal{T}$ for $\boldsymbol{T}$'s. The space $\mathcal{A}$ and $\mathcal{T}$ should satisfy the conditions of a valid graphical approach as in (1) and (2), and be constrained under a specific study design. For example, only α_1 and α_3 in the motivating example are allowed to be nonzero with sum equal to one-sided FWER α. Therefore, $\mathcal{A} = \{(\alpha_1, 0, \alpha_3, 0); \alpha_1 \in [0, \alpha], \alpha_3 \in [0, \alpha], \alpha_1 + \alpha_3 = \alpha\}$. The α_1 is a free parameter to be optimized, and further, set $\alpha_3 = \alpha - \alpha_1$. Denote all free parameters in $\boldsymbol{\alpha}$ and $\boldsymbol{T}$ as $\bar{\boldsymbol{\alpha}} \in \bar{\mathcal{A}}$ and $\bar{\boldsymbol{T}} \in \bar{\mathcal{T}}$, respectively. In this motivating example, we have $\bar{\mathcal{A}} = \{\alpha_1; \alpha_1 \in [0, \alpha]\}$. There is no inequality constraint on $\alpha_1 \in \bar{\mathcal{A}}$ in this simple problem because α_3 is excluded from $\bar{\mathcal{A}}$. In the simulation studies considered in Section 4 and the case study in Section 5 with more endpoints, both bounded and inequality constraints exist.

The goal of our optimization task is to find the optimal graphical approach with $\boldsymbol{\alpha}^{\text{opt}}$ and $\boldsymbol{T}^{\text{opt}}$ within their corresponding parameter spaces that maximize the object function $O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$ in (3),

$$\left\{\boldsymbol{\alpha}^{\text{opt}}, \boldsymbol{T}^{\text{opt}}\right\} = \underset{\boldsymbol{\alpha} \in \mathcal{A}, \boldsymbol{T} \in \mathcal{T}}{\text{argmax}}\ O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right). \qquad (4)$$

However, $O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$ in (3) does not necessarily have a closed form solution due to: (1) the underlying correlation structure in the multivariate distribution of $\boldsymbol{p}$ and (2) the additional dependence in the decision function $D_i\left(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}\right)$ among endpoints introduced by Algorithm 1. In practice, a Monte Carlo approach can be implemented to estimate $O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$. By simulating n sets of unadjusted p-values $\boldsymbol{p}_j = (p_{j1}, p_{j2}, \ldots, p_{jm}), j = 1, 2, \ldots, n$, for m endpoints based on prior knowledge, one can use the following working objective function to estimate (3) empirically,

$$\widehat{O}\left(\boldsymbol{\alpha}, \boldsymbol{T}\right) = \frac{1}{n} \sum_{i=1}^{m} v_i \sum_{j=1}^{n} D_i\left(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}_j\right). \qquad (5)$$

Some standard softwares, for example, R package *gMCP* (Rohmeyer and Klinglmueller 2018), can calculate $D_i\left(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}_j\right)$ given each set of simulated unadjusted p-values $\boldsymbol{p}_j$. By the law of large numbers, we have

$$\widehat{O}\left(\boldsymbol{\alpha}, \boldsymbol{T}\right) = O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right) + o_p(1). \qquad (6)$$

The approximation error of estimating $O\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$ by $\widehat{O}\left(\boldsymbol{\alpha}, \boldsymbol{T}\right)$ can be arbitrarily small to satisfy practical numerical precision requirements by choosing a sufficiently large n in the

Monte Carlo method. In the next section, we introduce our proposed optimization algorithm by using FNN to approximate $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$, and then conduct optimization with available gradient information.

3. FNN-Based Optimizer

In Section 3.1, FNNs in deep learning are briefly reviewed as powerful representations of complex objective functions. In Section 3.2, we illustrate our proposed FNN-based optimization method in detail. It takes advantage of FNN to characterize the nonconvex working objective function $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$ and then performs constrained optimization with gradient information.

3.1. Feedforward Neural Networks (FNNs)

We first review some basic knowledge of FNNs, which form a very popular and useful set of deep learning models.

An FNN defines a mapping $y = f(\boldsymbol{x}; \boldsymbol{\theta})$ and learns the value of parameters $\boldsymbol{\theta}$ that result in the best function approximation with input vector $\boldsymbol{x}$ and output y (Goodfellow, Bengio, and Courville 2016). It typically has four essential components: input data with corresponding targets, layers, loss function, and optimizer (Chollet and Allaire 2018; Liang, Ye, and Fu 2018). Figure 2 represents an FNN with two hidden layers, which have three and two nodes, respectively. From left to right, input data $\boldsymbol{x}$, which is the vector stack of x_1 and x_2, are transformed by two hidden layers and further mapped to output target Y. The loss function represents the quantity that is minimized during training, for example, the cross-entropy for binary classification and mean squared error (MSE) for regression. We choose MSE because our output $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$ ranges from 0 to 1. The optimizer determines how the network will be updated based on the loss function. The RMSProp algorithm (Hinton, Srivastava, and Swersky 2012) modifies AdaGrad (Duchi, Hazan, and Singer 2011) to perform better in the nonconvex setting by changing the gradient accumulation into an exponentially weighted moving average. It has been shown to be an effective and practical optimization algorithm for deep neural networks (Goodfellow, Bengio, and Courville 2016), and is used in this article.

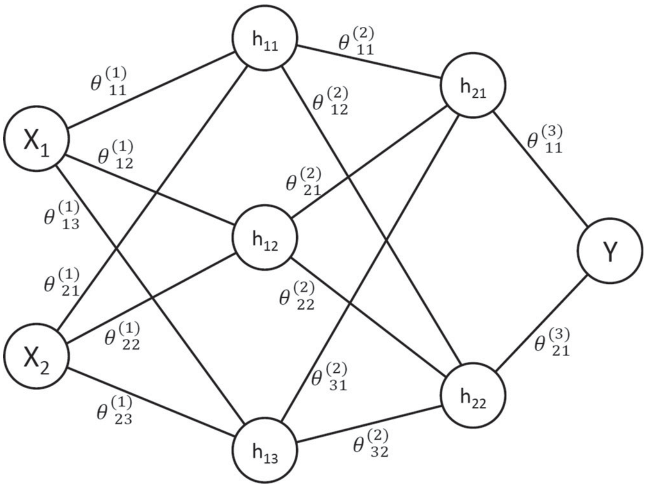


Figure 2. Feedforward neural networks with two hidden layers.

For an FNN with $L - 1$ hidden layers and one output layer, it can be recursively formulated as

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = f^{(L)}\left[\theta^{(L)\prime} \ldots f^{(2)}\left\{\theta^{(2)\prime} f^{(1)}\left(\theta^{(1)\prime}\boldsymbol{x} + \boldsymbol{b}_1\right) + \boldsymbol{b}_2\right\} \cdots + b_L\right]. \quad (7)$$

In the most inner layer, $\theta^{(1)}$ is a weight matrix that transforms input $\boldsymbol{x}$ to the first hidden layer. For example, the dimension of $\theta^{(1)}$ is 2 × 3 in Figure 2. The number of elements in bias vector $\boldsymbol{b}_1$ is equal to the number of nodes in the first layer (i.e., 3). The vector $\boldsymbol{\theta}$ denotes a stack of all those weight and bias parameters. There are many choices for the activation function $f^{(1)}()$, for example, the rectified linear unit or ReLU (Nair and Hinton 2010), the softplus function (Dugas et al. 2001), and the sigmoid function.

The approximation error of an FNN $f(\boldsymbol{x}; \boldsymbol{\theta})$ in approximating the objective function $O(\boldsymbol{x})$ in (3) is defined as

$$\sup_{\boldsymbol{x}} \left| f(\boldsymbol{x}; \boldsymbol{\theta}) - O(\boldsymbol{x}) \right|,$$

where $\boldsymbol{x} = (\boldsymbol{\alpha}, \boldsymbol{T})$. Previously, it has been shown that a depth-2 neural network with sigmoid activation function can approximate any continuous function to a desired accuracy, with sufficiently large number of nodes (Cybenko 1989). Since then, interest has shifted toward a deeper network, since the multilayer feedforward architecture itself gives neural networks the potential of being universal approximators (Hornik 1991). Recently, Bach (2017) provided the uniform approximation error of Lipschitz-continuous functions in the context of high-dimensional nonlinear variable selection. The error bound for approximation functions in Sobolev spaces by deep ReLU networks is studied by Yarotsky (2017).

As discussed in Section 2.2, theoretical properties of the objective function $O(\boldsymbol{\alpha}, \boldsymbol{T})$ in (3) are hard to study, mainly due to the dependence between decision function $D_i(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})$'s introduced by Algorithm 1. In this article, we use simulation studies to empirically check the MSE of modeling the working objection function $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$ by FNN, with more details in the following Section 3.2.4. Moreover, our algorithm has a last step in Section 3.2.6 to fine tune the optimal solution obtained from the FNN model to correct approximation errors.

3.2. Optimizing Procedures

In this section, we illustrate our optimizing procedures in six steps.

3.2.1. Define an Objective Function

The first step is to specify an objective function $O(\boldsymbol{\alpha}, \boldsymbol{T})$ to measure the performance of the graphical procedure for MTP. The vector $\boldsymbol{v}$ in (3) needs to be prespecified and reflects the relative importance of different endpoints.

3.2.2. Obtain Training Data

The second step is to generate training data with B graphs and their corresponding objective functions (3). In each graph b, one randomly generates $\boldsymbol{\alpha}_b \in \mathcal{A}$ and $\boldsymbol{T}_b \in \mathcal{T}$ under conditions (1) and (2), along with other constraints based on different study objectives. In the motivating example, we have $\mathcal{A} = \{(\alpha_1, 0, \alpha_3, 0), \alpha_1 \in [0, \alpha], \alpha_3 \in [0, \alpha], \alpha_1 + \alpha_3 = \alpha\}$. In this case, α_1 can be sampled from a uniform distribution Unif$(0, \alpha)$ and further set $\alpha_3 = \alpha - \alpha_1$. The free parameter vector $\bar{\boldsymbol{\alpha}}_b$ only contains α_1 in this case. It is important to enforce these constraints at this stage to achieve constraint optimization of the graphical approach.

We further simulate n sets of unadjusted p-values $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{im}), i = 1, 2, \ldots, n$ based on prior knowledge. Suppose that the marginal powers of four endpoints are 95%, 88%, 92%, and 85%, which correspond to a test statistic's mean at $\boldsymbol{e} = (3.60, 3.13, 3.37, 3.00)$ with one-sided Type I error $\alpha = 0.025$. We adopt a popular assumption that the test statistics from m hypotheses follow a multivariate normal distribution (Dmitrienko and D'Agostino Sr 2013; Bretz, Hothorn, and Westfall 2016). Unit variance is assumed for demonstration. Without loss of generality, by assuming that a larger statistic corresponds to a better clinical outcome, the one-sided p-value is calculated as the upper cumulative distribution function from a standard normal distribution. Having $\boldsymbol{p}_i$'s simulated, one calculates $\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$ in (5) for each graph b. The input covariate vector $\bar{\boldsymbol{x}}_b$ of FNN is $(\bar{\boldsymbol{\alpha}}_b, \bar{\boldsymbol{T}}_b)$, while the output variable is $\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$. The dimension of $\bar{\boldsymbol{x}}_b$ is equal to the number of input parameters of FNN on the left hand side of Figure 2.

3.2.3. Select FNN Structure

The next step is to select the structure of FNN, specifically the width (number of nodes), the depth (number of layers), and the rate of the dropout technique, which randomly deactivates a certain proportion of nodes in each iteration, to accommodate the potential overfitting issue in FNN.

The most common practice is to perform a k-fold cross-validation procedure on several reasonable candidate structures (Goodfellow, Bengio, and Courville 2016). In cross-validation, a partition of the dataset is formed by splitting it into k non-overlapping subsets. On each trial i, for $i = 1, \ldots, k$, the ith subset of data is used as the validation set while the rest of the data is used as the training set. The validation error is calculated by averaging test error across k trials. We let $k = 5$ to implement a 5-fold cross-validation. The final FNN structure is selected as the one with the smallest training error among candidates. Validation error or other measures can also be used, and the performance of our method is consistent.

We recommend starting with an architecture with a relatively large capability to reduce the training error (MSE) under a desired level of tolerance, for example, 10^{-4}. This ensures that the functional space defined by the structure is large enough to include the underlying objection function, or a very good approximation of it. However, this structure may be overwhelmed with a high validation error. Then we apply the dropout technique as a regulation approach to prevent overfitting and to increase the generalizability of the model. In the context of this article, exploratory simulations show that the performance of our FNN-based optimizer is robust to different choices of FNN structures, when both the training MSE and the validation MSE are less than 10^{-4}.

3.2.4. Train FNN

The following step is to train the FNN with structure obtained in 5-fold cross-validation with input covariates $\bar{\boldsymbol{x}}_b$ and output $\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$, $b = 1, 2, \ldots, B$. Covariates $\bar{\boldsymbol{x}}_b$ are standardized to achieve better performance of a gradient-based optimizer, and are further transformed back to the original scale after fitting. MSE is used as the loss function,

$$\frac{1}{B} \sum_{b=1}^{B} \left[\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b) - f(\bar{\boldsymbol{x}}_b; \boldsymbol{\theta})\right]^2. \tag{8}$$

The least squares estimator $\widehat{\boldsymbol{\theta}}$ is obtained from the RMSProp algorithm discussed in Section 3.1 as the $\boldsymbol{\theta}$ that minimizes this loss function. The fitted FNN is denoted as $f(\bar{\boldsymbol{x}}; \widehat{\boldsymbol{\theta}})$. For a specific problem, it is critical to check this MSE to evaluate the approximation error of FNN empirically. The estimation error between $\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$ and $O(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$ is further controlled by using a relatively larger n in (6). Before implementing our proposed optimization method, it is critical to quantify the approximation ability of FNN by checking the MSE in (8).

Since sigmoid functions saturate (have small gradients) when input data are at two tails, we further normalize $\widehat{O}(\boldsymbol{\alpha}_b, \boldsymbol{T}_b)$ to a subset of $[0, 1]$, for example, $[0.3, 0.7]$. The optimal solution would be invariant under this transformation. The whole training process is implemented by the R interface *keras* (Allaire and Chollet 2018; Allaire and Tang 2018) to a high-level neural networks API *Keras* (Chollet 2015) with back-end engine *Tensorflow* (Abadi et al. 2015) developed by Google Inc. We set the training epoch as 10^3.

3.2.5. Perform Constrained Optimization

Up to this point, we have transformed the original optimization problem in (4) to the following constrained minimization problem of identifying optimal solution $\bar{\boldsymbol{x}}_b^{\text{opt}}$ on $-f(\bar{\boldsymbol{x}}_b; \widehat{\boldsymbol{\theta}})$,

$$\bar{\boldsymbol{x}}_b^{\text{opt}} = \underset{\bar{\boldsymbol{\alpha}} \in \bar{\mathcal{A}}, \bar{\boldsymbol{T}} \in \bar{\mathcal{T}}}{\operatorname{argmax}} \; f(\bar{\boldsymbol{x}}_b; \widehat{\boldsymbol{\theta}}) = \underset{\bar{\boldsymbol{\alpha}} \in \bar{\mathcal{A}}, \bar{\boldsymbol{T}} \in \bar{\mathcal{T}}}{\operatorname{argmin}} \left\{-f(\bar{\boldsymbol{x}}_b; \widehat{\boldsymbol{\theta}})\right\}. \tag{9}$$

The optimal graph parameters $\boldsymbol{\alpha}^{\text{opt}}$ and $\boldsymbol{T}^{\text{opt}}$ are further calculated from $\bar{\boldsymbol{x}}_b^{\text{opt}}$.

Since $-f(\bar{\boldsymbol{x}}_b; \widehat{\boldsymbol{\theta}})$ in (7) is not necessarily a convex function, then the Karush–Kuhn–Tucker (KKT) conditions (Karush 1939; Kuhn and Tucker 1951) are not sufficient for a point to be globally optimal. Even with gradient information available, finding the global optimal solution is still challenging depending on the objective function at hand (Kramer, Ciaurri, and Koziel 2011). We turn to the augmented Lagrangian method (Hestenes 1969; Powell 1969), which seeks the solution by replacing the original constrained problem by a sequence of unconstrained subproblems (Nocedal and Wright 2006). This algorithm is related to the quadratic penalty method (Courant 1943), but reduces the possibility of ill conditioning of the subproblems by introducing a Lagrange multiplier into the function to be minimized.

This algorithm, as well as COBYLA and ISRES discussed later on, is implemented by the R package *nloptr* (Ypma 2018), which is the R interface to a nonlinear optimization library *NLopt* (Johnson 2007; Conn, Gould, and Toint 1991; Birgin and Martínez 2008). The fractional tolerance on the input data is 10^{-5}, which means that the algorithm terminates when the changes of each parameter in one iteration are less than 10^{-5} multiplied by the absolute value of the parameter. The maximum number of iterations is 10^5.

3.2.6. Fine Tune the Final Optimal Solution

As a final step, we fine-tune the solution with COBYLA, an existing derivative-free optimization method that can handle inequality constraints. Essentially, our optimal solution from the previous step is used as the starting values in COBYLA. The fractional tolerance on the input data is 10^{-4}, and the maximum number of iterations is 10^4.

4. Simulation Studies

Now we move on to a simulation study to evaluate the performance of our proposed FNN-based optimizer against the stochastic search method (SSM) and two derivative-free optimization methods that can handle bound and inequality constraints: COBYLA and ISRES.

Suppose that the study objective is to identify the optimal graphical procedure that maximizes a weighted average of multiplicity adjusted power for $m = 6$ endpoints. One can work out that the input covariate vector $\bar{\boldsymbol{x}}$ of FNN is $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, T_{12}, T_{13}, T_{14}, T_{15}, T_{21}, T_{23}, T_{24}, T_{25}, T_{31}, T_{32}, T_{34}, T_{35}, T_{41}, T_{42}, T_{43}, T_{45}, T_{51}, T_{52}, T_{53}, T_{54}, T_{61}, T_{62}, T_{63}, T_{64})$ with 29 elements. The constraints are

$$0 \le \alpha_j \le \alpha \text{ for } j \in \{1,2,3,4,5\}, \tag{10}$$

$$\sum_{j \in \{1,2,3,4,5\}} \alpha_j \le \alpha, \tag{11}$$

$$0 \le T_{ij} \le 1 \text{ for } i \in \{1,2,3,4,5\}, j \in \{1,2,3,4,5, j \ne i\},$$
$$0 \le T_{6j} \le 1, j \in \{1,2,3,4\}, \tag{12}$$

$$\sum_{j \in \{1,2,3,4,5, j \ne i\}} T_{ij} \le 1 \text{ for } i \in \{1,2,3,4,5\} \quad \sum_{j \in \{1,2,3,4\}} T_{6j} \le 1. \tag{13}$$

Condition (10) says that the initial significance level from each of the first 5 endpoints is bounded between 0 and FWER at α, while constraint (11) ensures this for the last endpoint because $\alpha_6 = \alpha - \sum_{j=1}^{5} \alpha_j$. Constraints (12) and (13) are the corresponding constraints for each of the 6 rows in the transitional matrix $\boldsymbol{T}$.

We consider $\boldsymbol{v} = (0.3, 0.3, 0.1, 0.1, 0.1, 0.1)$ as the relative importance in (3) in this section, and turn to a different $\boldsymbol{v}$ in the Section 5 case study. As discussed in Section 3.2.2, we assume that the test statistics from m endpoints follow a multivariate normal distribution with unit variance and mean computed from their corresponding marginal power under a one-sided FWER at 0.025. The setup parameters are specified in Table 1 with varying marginal power, different correlation structures and varying magnitudes of correlation.

For the FNN-based optimizer as described in Section 3.2, we simulate $B = 10^3$ random graphs and $n = 10^6$ sets of p-values to establish the training dataset. The size $B = 10^3$ is sufficient to give us the training MSE and the validation MSE less than 10^{-4} in all scenarios considered, but it can be increased in more complicated cases. In cross-validation while selecting the FNN structure, the following 6 sets of candidate structures are considered: 2 layers with drop-out rate 0, 3 layers with rate at 0, 4 layers with rate 0, 2 layers with rate 0.3, 3 layers with rate 0.3, and 4 layers with rate 0.3. The number of nodes per layer is considered at 30. This cohort of FNN candidate skeletons are used throughout this article.

Table 1. Parameter specifications for simulations.

Scenario	Marginal power	Correlation structure	Correlation magnitude
L_1	(0.8, 0.8, 0.6, 0.6, 0.4, 0.4)	Compound symmetry	0
L_2			0.3
L_3			0.5
L_4	(0.9, 0.9, 0.8, 0.8, 0.6, 0.6)	Compound symmetry	0.3
L_5		AR(1)	
L_6		Banded Toeplitz	
L_7	(0.9, 0.8, 0.7, 0.6, 0.5, 0.4)	Compound symmetry	0.3
L_8	(0.9, 0.9, 0.7, 0.7, 0.6, 0.6)		
L_9	(0.95, 0.95, 0.8, 0.8, 0.6, 0.6)		

Table 2. Optimal $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$ identified by FNN, ISRES, COBYLA, and SSM with the maximum solution highlighted in bold.

Scenario	Optimal $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$				Convergence time (min)			
	FNN	COBYLA	ISRES	SSM	FNN	COBYLA	ISRES	SSM
L_1	**55.9%**	55.5%	47.8%	53.4%	25.0	13.8	–	3.6
L_2	**57.9%**	57.4%	48.7%	55.6%	29.2	18.2	–	4.5
L_3	**59.1%**	58.8%	48.2%	54.0%	30.8	20.2	–	3.8
L_4	**74.6%**	74.4%	68.9%	72.6%	30.2	18.3	–	4.4
L_5	**74.0%**	73.8%	69.7%	72.7%	28.0	17.4	–	4.5
L_6	**74.0%**	73.7%	69.7%	72.9%	28.0	15.2	–	4.6
L_7	**64.2%**	63.7%	55.6%	61.5%	30.7	18.7	–	4.1
L_8	**71.8%**	71.6%	65.7%	69.3%	30.6	20.3	–	4.4
L_9	**79.4%**	79.2%	74.6%	78.0%	34.3	23.8	–	4.7

In ISRES and COBYLA, fractional tolerance on the input data is 10^{-4}, which is consistent with the termination condition at our fine-tuning step at Section 3.2.6. The maximum evaluation time is set as 1.5 times the fitting time of the FNN-based optimizer as described in Section 3.2. The initial values are randomly generated under the constraints in (10)–(13). We consider a size of 10^3 for the random search in the SSM. This is equal to the training size B in our method.

In Table 2, we summarize the optimal working objective function $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$ identified by the FNN-based optimizer, ISRES, COBYLA and SSM along with their corresponding convergence times in minutes based on a MacBook Pro with 2.3 GHz Intel Core i7. The average from five separate optimizations are reported for FNN-based method, ISRES and COBYLA to evaluate the robustness of their performance. Our method has a smaller standard deviation at 0.14%, compared with COBYLA at 0.72% and ISRES at 1.09%, averaging the standard deviations across nine scenarios. In all scenarios evaluated, our FNN method consistently identifies a graph with the highest $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$. However, the performance of the other two methods is not stable; for example, COBYLA finds 63.7% compared to 64.2% from the FNN method in L_7, and ISRES yields 48.2% compared to 59.1% in L_3. The SSM method can also lead to substantial optimal power loss in some scenarios. For example, the deviance can be as high as 5.1% compared to our FNN method in L_3 (59.1% vs. 54.0%). As for convergence, SSM is the fastest, and COBYLA is the second fastest, followed by our FNN-based method, followed by ISRES. The convergence time of ISRES is missing because it does not converge before the maximum wall time, which is the computational time of the FNN-based optimizer multiplied by 1.5. Our FNN-based optimizer offers a better balance between time efficiency and robustness in identifying the optimal graphical approach.

We observe that ISRES does not converge within the given wall time, and further delivers a low optimal value. A possible reason is that it takes longer for ISRES to comprehensively walk through the whole parameter space in this setup with a moderate dimension of $\bar{\boldsymbol{x}}$ at 29. The performance of COBYLA is not stable in those settings, as demonstrated by the relatively larger standard deviation and smaller mean optimal values compared with FNN. The reason can be that COBYLA is more likely to get stuck in the local optimal. Our proposed method, on the other hand, first seeks a parametric surrogate function approximating the working objective function in (5) by FNN, and then performs optimization with available gradient information. Therefore, our FNN-based optimizer consistently achieves the highest power across all scenarios.

From a practical point of view, both FNN and COBYLA have satisfactory performance based on this simulation study. COBYLA takes a moderate computational time of approximately 20 min to deliver the solution. Our proposed FNN-based method offers an alternative option to further enhance the power. An extra 15 min is tolerable as compared with study duration over years in confirmatory trials. Moreover, even a fraction of a percent of power gain is nontrivial considering the high cost of clinical trials. More discussions on this are provided in Section 6.

In Figure 3, we visualize the performance of our FNN-based method and the other three comparators. In each scenario, we plot 800 training datasets in green in the order of their working objective functions from small to large, and then 200 validation datasets with blue in order. The maximum of both training and validation datasets is the solution of SSM. The optimal graph identified by ISRES in triangle, COBYLA in rhombus and FNN in orange circle are plotted on the right. Next, we evaluate the residuals of using FNN to approximate $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T})$, which is $\widehat{O}(\boldsymbol{\alpha}, \boldsymbol{T}) - f(\bar{\boldsymbol{x}}; \widehat{\boldsymbol{\theta}})$ as in (8). The residuals from the left 800 training datasets are generally smaller than those from the right 200 validation datasets (Figure 4).

5. A Case Study

In this section, we apply our FNN-based optimizing approach to a generic study with one primary endpoint, denoted as H_1,

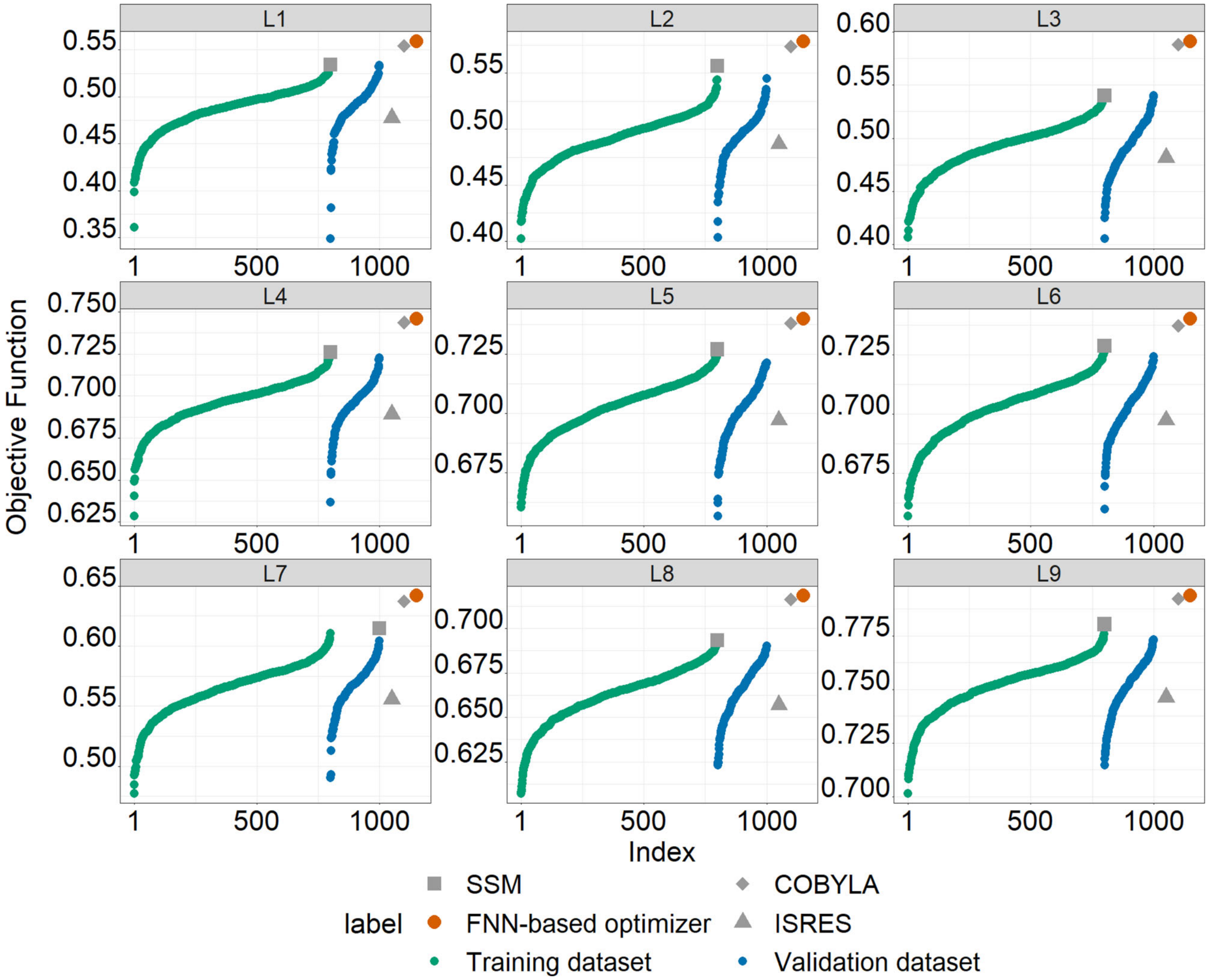


Figure 3. The optimal objective function identified by FNN, COBYLA, ISRES, and SSM.

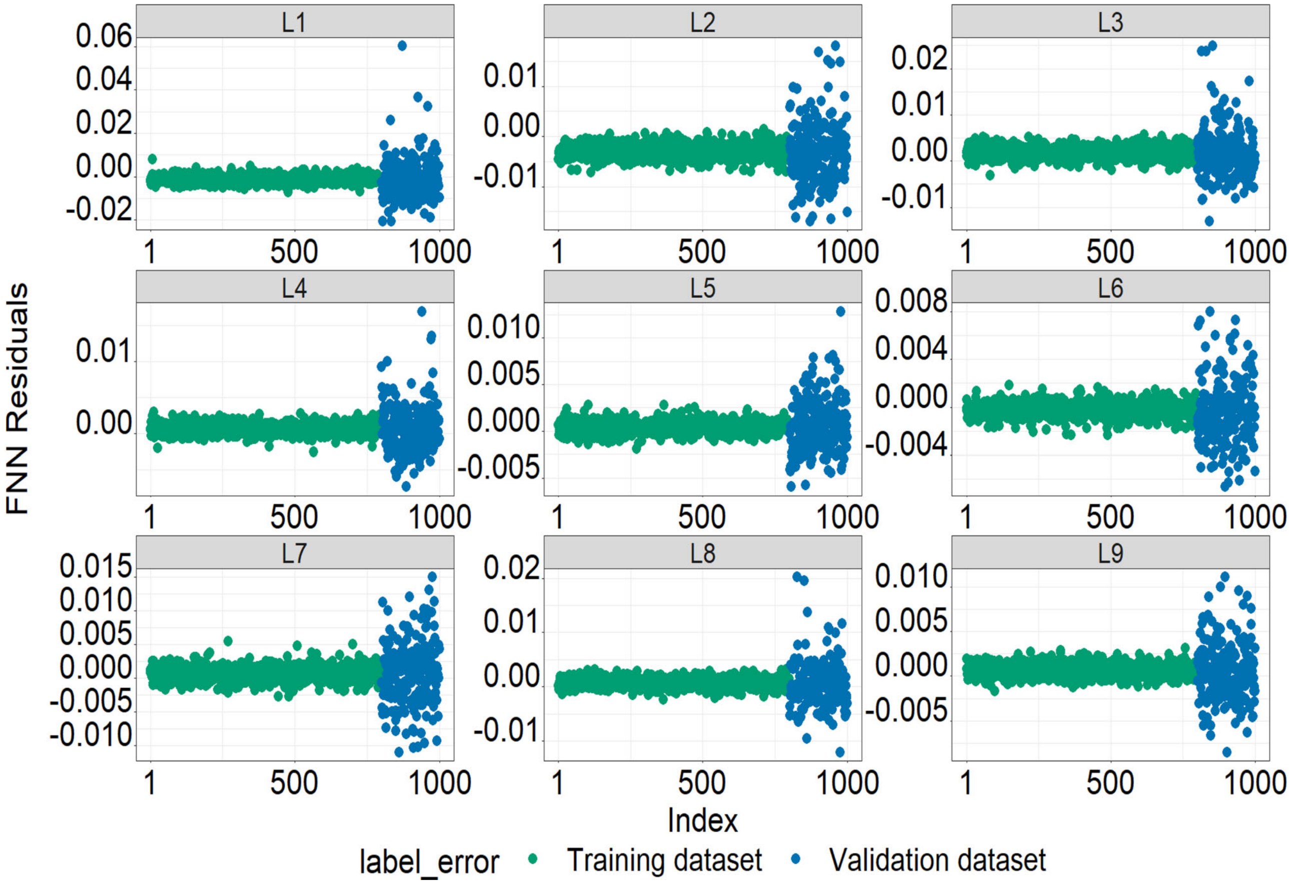


Figure 4. FNN residuals of estimating the working objective function.

and four secondary endpoints, denoted as H_2, H_3, H_4, and H_5. This example is particularly relevant because while testing the primary endpoint first is clear, the tactic for testing the secondary endpoints can be flexible. Since the primary endpoint is tested first, we fix the first element in $\boldsymbol{\alpha}$ at the one-sided FWER 0.025 and the remaining components at 0. There is no element in $\bar{\boldsymbol{\alpha}}_b$ to be optimized. In the transition matrix $\boldsymbol{T}$, H_1 can freely pass its error rate to all secondary endpoints, and each secondary endpoint can recycle theirs to the other 3 secondary endpoints but not back to the primary endpoint since it must have been rejected first. Therefore, there are $3 + 4 \times 2 = 11$ elements in $\bar{\boldsymbol{T}}_b = (T_{12}, T_{13}, T_{14}, T_{23}, T_{24}, T_{32}, T_{34}, T_{42}, T_{43}, T_{52}, T_{53})$, and each element is bounded between 0 and 1. The additional constraints are

$$\begin{aligned} T_{12} + T_{13} + T_{14} &\le 1, \\ T_{23} + T_{24} &\le 1, \\ T_{32} + T_{34} &\le 1, \\ T_{42} + T_{43} &\le 1, \\ T_{52} + T_{53} &\le 1. \end{aligned}$$

Suppose that the study team assigns $v_2 = 0.6$, $v_3 = 0.2$, $v_4 = v_5 = 0.1$ to the following objective function,

$$O'(\boldsymbol{\alpha}, \boldsymbol{T}) = \sum_{i=2}^{5} v_i \, \mathrm{E}_{\boldsymbol{p}} \left\{ D_i'(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}) \right\}, \qquad (14)$$

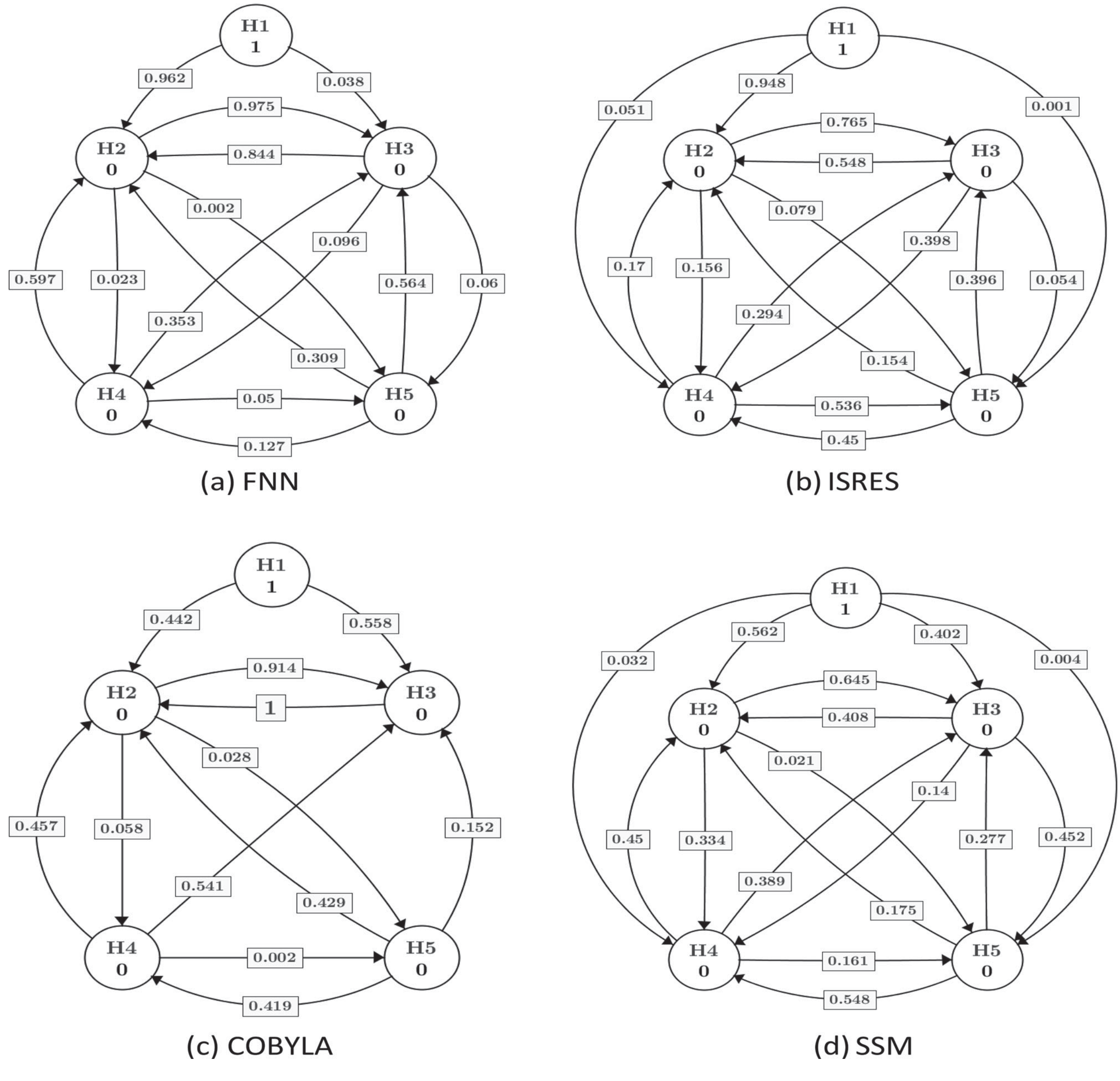
where $D_i'(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p}) = 1$ if both H_i and H_1 are rejected by the graphical approach $g(\boldsymbol{\alpha}, \boldsymbol{T})$, and 0 otherwise, for $i = 2, 3, 4, 5$. This reflects the clinical interpretation that the rejection of a secondary endpoint is only meaningful if the primary endpoint has been rejected. Note that we exclude the adjusted power of the primary endpoint in Equation (14), because the optimizer is equivalent given the constraints on $\boldsymbol{\alpha}$ in the study setup.

We further assume that the test statistics follow a multivariate normal distribution with a compound symmetric structure and a common correlation at 0.5. The marginal power of the primary endpoint is assumed to be 95%, and 90%, 85%, 65% and 60% for secondary endpoints. The parameters in the FNN-based optimizer, ISRES and COBYLA are the same as those specified in Sections 3.2 and 4.

In Table 3, we list the optimal $\widehat{O}'(\boldsymbol{\alpha}, \boldsymbol{T})$ and the multiplicity adjusted power of each endpoint from the FNN-based method, ISRES, COBYLA, and SSM. Our method achieves the highest objective function at 78.0%, which is approximately 0.6%–1.4% higher than the other three methods. When it comes to convergence time, COBYLA takes 4.0 min, which is shorter than the 17.1 min from our method. ISRES does not converge in the given wall time at 25.7 min. The optimal graphs identified by the four methods are also visualized in Figure 5. To demonstrate the reproducibility of our findings, we further perform 100 replications of this case study. Our method has the highest mean of optimal $\widehat{O}'(\boldsymbol{\alpha}, \boldsymbol{T})$ at 78.0%, compared with 77.0% from ISRES, 77.7% from COBYLA, and 76.4% from SSM. The results are consistent with our report at Table 3. Our proposed method also has a relatively small standard deviation at 0.04%, while ISRES has 0.23%, COBYLA has 0.54%, and SSM has 0.03%.

Table 3. Optimal $\widehat{O}'(\boldsymbol{\alpha}, \boldsymbol{T})$ and $\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_i(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$ identified by FNN, ISRES, COBYLA, and SSM.

Method	$\widehat{O}'(\boldsymbol{\alpha}, \boldsymbol{T})$	$\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_1(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$	$\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_2(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$	$\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_3(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$	$\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_4(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$	$\widehat{\mathrm{E}}_{\boldsymbol{p}}\{D'_5(\boldsymbol{\alpha}, \boldsymbol{T}, \boldsymbol{p})\}$
FNN	78.0%	95.0%	86.7%	78.4%	54.6%	48.5%
ISRES	77.4%	95.0%	86.6%	75.2%	56.8%	47.6%
COBYLA	77.2%	95.0%	84.7%	80.8%	54.5%	48.4%
SSM	76.6%	95.0%	84.0%	79.0%	54.4%	49.7%



(a) FNN

(b) ISRES

(c) COBYLA

(d) SSM

Figure 5. Optimal graph identified by FNN, ISRES, COBYLA, and SSM.

6. Concluding Remarks

In this article, we propose an FNN-based optimization framework for the graphical procedure of multiplicity control in confirmatory clinical trials. This framework takes advantage of the strong functional representation of deep neural networks and further uses constraint optimization techniques to locate the solution. Simulation studies show that our FNN-based optimizer consistently identifies the optimal graph, and has a better balance between robustness and time efficiency as compared to two popular derivative-free optimization methods that can handle bound and inequality constraints.

Our proposed method numerically approximates the optimal graph from the graphical approach with respect to the objective function under specified constraints. An optimal solution may not be unique. Numerical approximation is deemed appropriate due to the intractable nature of the underlying objective function. Numerical precision needs to be considered case by case because of the different number of simulated graphs (B) and finite simulated p-values (n) in the training dataset. For increased precision of the approximate solution, one can further increase B and n.

In practice, a relatively simplified graph may be more palatable for the clinical team, as compared to the numerically optimized upper bound with respect to the objective function identified by our method. If the distance in the objective function is relatively small, then this evidence adds more justification to the usage of the proposed simple graph. On the other hand, if power is of main interest, then our method has moderate power gain as compared with the two existing derivative-free methods and the SSM demonstrated by the simulation studies and the case study. As shown in Table 2, the average multiplicity adjusted power increase can be as high as 5.1% compared with the SSM method, over 10% as compared with ISRES, and 0.5% compared with COBYLA. This makes our proposed method appealing in confirmatory studies where several secondary endpoints are targeted for labeling purposes. Even though the gain in some cases is merely a fraction of a percent of power, it is still worth the additional computing time, which is never more than a couple of minutes, especially if either the cost of the study is high or the stakes are high based on participation of subjects with serious afflictions.

Acknowledgments

The authors would like to thank an anonymous associate editor and anonymous referees for their many insightful comments and suggestions.

Funding

The support of this article was provided by AbbVie Inc. and Takeda Pharmaceuticals USA, Inc. AbbVie Inc. and Takeda Pharmaceuticals USA, Inc. participated in the review and approval of the content. Tianyu Zhan is an employee of AbbVie Inc. Alan Hartford is employed by Takeda Pharmaceuticals USA, Inc. Jian Kang is Professor in the Department of Biostatistics at the University of Michigan, Ann Arbor. Kang's research was partially supported by NIH R01 GM124061 and R01 MH105561. Walter Offen is a former employee of AbbVie and is retired. All authors may own AbbVie stock. The R code is available at *https://github.com/tian-yu-zhan/DNN_optimization*.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015), "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," available at *https://www.tensorflow.org/*. [5]

Allaire, J., and Chollet, F. (2018), "keras: R Interface to 'Keras,'" R Package Version 2.2.0, available at *https://CRAN.R-project.org/package=keras*. [5]

Allaire, J., and Tang, Y. (2018), "tensorflow: R Interface to 'TensorFlow'," R Package Version 1.9, available at *https://CRAN.R-project.org/package=tensorflow*. [5]

Bach, F. (2017), "Breaking the Curse of Dimensionality With Convex Neural Networks," *The Journal of Machine Learning Research*, 18, 629–681. [4]

Birgin, E. G., and Martínez, J. M. (2008), "Improving Ultimate Convergence of an Augmented Lagrangian Method," *Optimization Methods and Software*, 23, 177–195. [5]

Bretz, F., Hothorn, T., and Westfall, P. (2016), *Multiple Comparisons Using R*, Boca Raton, FL: CRC Press. [5]

Bretz, F., Maurer, W., Brannath, W., and Posch, M. (2009), "A Graphical Approach to Sequentially Rejective Multiple Test Procedures," *Statistics in Medicine*, 28, 586–604. [1,3]

Chollet, F. (2015), "Keras," available at *https://keras.io*. [5]

Chollet, F., and Allaire, J. (2018), *Deep Learning With R*, Shelter Island, NY: Manning Publications. [4]

Conn, A. R., Gould, N. I., and Toint, P. (1991), "A Globally Convergent Augmented Lagrangian Algorithm for Optimization With General Constraints and Simple Bounds," *SIAM Journal on Numerical Analysis*, 28, 545–572. [5]

Courant, R. (1943), "Variational Methods for the Solution of Problems With Equilibrium and Vibration," *Bulletin of the American Mathematical Society*, 49, 1–23. [5]

Cybenko, G. (1989), "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, 2, 303–314. [4]

Dmitrienko, A., and D'Agostino, R., Sr. (2013), "Traditional Multiplicity Adjustment Methods in Clinical Trials," *Statistics in Medicine*, 32, 5172–5218. [5]

Dmitrienko, A., Offen, W. W., and Westfall, P. H. (2003), "Gatekeeping Strategies for Clinical Trials That Do Not Require All Primary Effects to Be Significant," *Statistics in Medicine*, 22, 2387–2400. [1]

Duchi, J., Hazan, E., and Singer, Y. (2011), "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, 12, 2121–2159. [4]

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R. (2001), "Incorporating Second-Order Functional Knowledge for Better Option Pricing," in *Advances in Neural Information Processing Systems*, pp. 472–478. [4]

Food and Drug Administration (2017), "Multiple Endpoints in Clinical Trials Guidance for Industry," available at *https://www.fda.gov/downloads/drugs/guidancecomplianceregulatoryinformation/guidances/ucm536750.pdf*. [1]

Gabriel, K. R. (1969), "Simultaneous Test Procedures—Some Theory of Multiple Comparisons," *The Annals of Mathematical Statistics*, 40, 224–250. [2]

Goodfellow, I., Bengio, Y., and Courville, A. (2016), *Deep Learning*, Cambridge, MA: MIT Press. [2,4,5]

Hestenes, M. R. (1969), "Multiplier and Gradient Methods," *Journal of Optimization Theory and Applications*, 4, 303–320. [5]

Hinton, G., Srivastava, N., and Swersky, K. (2012), "Neural Networks for Machine Learning," *Coursera*, 264. [4]

Holm, S. (1979), "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, 6, 65–70. [1,2]

Hornik, K. (1991), "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, 4, 251–257. [4]

Johnson, S. G. (2007), "The NLopt Nonlinear-Optimization Package," available at *http://ab-initio.mit.edu/nlopt*. [5]

Karush, W. (1939), "Minima of Functions of Several Variables With Inequalities as Side Constraints," Master's Thesis, Department of Mathematics, University of Chicago, Chicago. [5]

Kazda, C. M., Ding, Y., Kelly, R. P., Garhyan, P., Shi, C., Lim, C. N., Fu, H., Watson, D. E., Lewin, A. J., Landschulz, W. H., Deeg, M. A., Moller, D. E., and Hardy, T. A. (2016), "Evaluation of Efficacy and Safety of the Glucagon Receptor Antagonist LY2409021 in Patients With Type 2 Diabetes: 12- and 24-Week Phase 2 Studies," *Diabetes Care*, 39, 1241–1249. [1]

Kelly, R., Garhyan, P., Raddad, E., Fu, H., Lim, C., Prince, M., Pinaire, J., Loh, M., and Deeg, M. (2015), "Short-Term Administration of the Glucagon Receptor Antagonist LY2409021 Lowers Blood Glucose In Healthy People and in Those With Type 2 Diabetes," *Diabetes, Obesity and Metabolism*, 17, 414–422. [1]

Kramer, O., Ciaurri, D. E., and Koziel, S. (2011), "Derivative-Free Optimization," in *Computational Optimization, Methods and Algorithms*, eds. S. Koziel and X-S. Yang, Berlin, Heidelberg: Springer-Verlag, pp. 61–83. [2,5]

Kuhn, H. W., and Tucker, A. W. (1951), "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492. [5]

Liang, M., Ye, T., and Fu, H. (2018), "Estimating Individualized Optimal Combination Therapies Through Outcome Weighted Deep Learning Algorithms," *Statistics in Medicine*, 37, 3869–3886. [2,4]

Marcus, R., Eric, P., and Gabriel, K. R. (1976), "On closed Testing Procedures With Special Reference to Ordered Analysis of Variance," *Biometrika*, 63, 655–660. [1,2]

Nair, V., and Hinton, G. E. (2010), "Rectified linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814. [4]

Nocedal, J., and Wright, S. J. (2006), *Nonlinear Equations*, New York: Springer. [5]

Powell, M. J. D. (1969), "A Method for Nonlinear Constraints in Minimization Problems," in *Optimization*, ed. R. Fletcher, London: Academic Press, pp. 283–298. [5]

——— (1994), "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation," in *Advances in Optimization and Numerical Analysis*, eds. S. Gomez and J. P. Hennart, Dordrecht: Springer, pp. 51–67. [2]

Rohmeyer, K., and Klinglmueller, F. (2018), "gMCP: Graph Based Multiple Test Procedures," R Package Version 0.8-14, available at *https://cran.r-project.org/package=gMCP*. [3]

Rubin, D., Dudoit, S., and Van der Laan, M. (2006), "A Method to Increase the Power of Multiple Testing Procedures Through Sample Splitting," *Statistical Applications in Genetics and Molecular Biology*, 5, 19. [1]

Runarsson, T. P., and Yao, X. (2005), "Search Biases in Constrained Evolutionary Optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, 35, 233–243. [2]

Tamhane, A. C., and Gou, J. (2018), "Advances in *p*-Value Based Multiple Test Procedures," *Journal of Biopharmaceutical Statistics*, 28, 10–27. [2]

Wasserman, L., and Roeder, K. (2006), "Weighted Hypothesis Testing," arXiv no. math/0604172. [1]

Westfall, P. H., and Krishen, A. (2001), "Optimally Weighted, Fixed Sequence and Gatekeeper Multiple Testing Procedures," *Journal of Statistical Planning and Inference*, 99, 25–40. [1]

Wiens, B. L. (2003), "A Fixed Sequence Bonferroni Procedure for Testing Multiple Endpoints," *Pharmaceutical Statistics*, 2, 211–215. [1]

Yarotsky, D. (2017), "Error Bounds for Approximations With Deep ReLU Networks," *Neural Networks*, 94, 103–114. [4]

Ypma, J. (2018), "nloptr: R Interface to NLopt," R Package Version 1.2.1, available at *https://cran.r-project.org/web/packages/nloptr/index.html*. [2,5]

Zabinsky, Z. B. (2013), *Stochastic Adaptive Search for Global Optimization* (Vol. 72), New York: Springer. [1]

Zhan, T., and Kang, J. (2019), "Targeting the Uniformly Most Powerful Unbiased Test in Sample Size Reassessment Adaptive Clinical Trials With Deep Learning," arXiv no. 1912.07433. [2]