ADAPTIVE EXPERIMENTAL DESIGN FOR OPTIMIZING COMBINATORIAL STRUCTURES

By

ARYAN DESHWAL

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

JULY 2024

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of ARYAN DESHWAL find it satisfactory and recommend that it be accepted.

Janardhan Rao Doppa, Ph.D., Chair

Ananth Kalyanaraman, Ph.D.

Yan Yan, Ph.D.

Alan Fern, Ph.D.

ACKNOWLEDGEMENTS

First and foremost, I am deeply grateful to my advisor Prof Jana Doppa for his immense support and guidance. This small space is not enough to talk about all the impact he's had on my life. From the beginning, he was always patient with me and taught me how to be a good scientist. He went out of his way to create opportunities for me and treated me no short of family. I have been extraordinarily fortunate to have him as my mentor. Since he and I both enjoy the sport of cricket, I must draw an analogy of how he played a role similar to John Wright/Sourav Ganguly in my life in making me believe and instill confidence that I can do good work similar to how they did for the Indian cricket team back in early 2000s.

I am thankful to all my thesis committee members, Prof Alan, Prof Ananth, Prof Yan who are inspiring role models for me. They have been extremely generous in supporting me in my academic career. I have also been fortunate to collaborate with several other amazing faculty members. Prof Partha taught me all about electronic design automation and I am thankful for the opportunity to work with him and his students. I really enjoyed my collaboration with Prof Cory whose infectious enthusiasm about research and so many things in life is very contagious. I am grateful to Prof Haipeng who's been extremely kind towards me and cheered me on from the beginning of my PhD.

I would like to thank Richard Song, Chansoo Lee and the Vizier team for host-

ing me for an internship at Google. I learned a lot about software engineering and organization skills through my interactions with them including our pair programming sessions. My meta internship with the AE team has been one of the biggest contributors to my career growth. I've always mentioned that AE is a special team and everyone there is very inspiring who taught me a lot. I've been fortunate to have a friend and mentor in David whose thoughtfulness and composure I admire. Sebastian's immense curiosity and creativity serves as an inspiration. I am also thankful to Eytan, Max, Susan and Sait, with whom I worked closely.

I have so many happy memories of WSU, largely due to the friendships I've formed here. It was a great delight to work together and learn from Syrine. I learnt a lot from her wonderfully focused work ethic and dedication towards research. She also made sure to always remind me about PhD well-being. I will cherish all our collaborations together. All my labmates including Alaleh, Taha, Rakib, Chibuike, Yassine, Azza, Amine, Subhankar, Hooman have become close friends.

I will fondly remember my time with my roommates and friends, masterchef fan Harsha and cinema expert Nitthilan. I would like to thank my undergrad friends Rishi, Ankit, Dev and Sachin for their years of friendship. Thanks to Dheeraj and Shlok for being family away from home.

I would like to thank all the staff at Washington State University for helping me over the years. I gratefully acknowledge the support from all funding agencies that

supported this work.

I am indebted to my mother Neelam and my father Yogesh for all their sacrifices and support throughout my life. They've always had unwavering belief in me and I am blessed to have been their son. Thanks to my brother Lakshay who inspires me everyday with his willpower and courageous attitude towards so many things in life. I have seen my sisters Anika and Saanvi grow in front of me and they have always been a great source of energy. I've always idolized my uncle Pravindra who has had a big part to play in my life (including choosing my name) and really shaped the course of my life with his guidance in everything from personal to professional life. Thanks to my grandparents who always pushed everyone in my family towards a good education. I could not do this work without the support of the love of my life and my best friend Tenzin. She has brought true happiness to my life and has supported me in all the highs and lows of doing a PhD. She is the true foundation and pillar of strength behind this work.

ADAPTIVE EXPERIMENTAL DESIGN FOR OPTIMIZING COMBINATORIAL STRUCTURES

Abstract

by Aryan Deshwal, Ph.D.
Washington State University
July 2024

Chair: Janardhan Rao Doppa

Many real-world scientific and engineering problems can be formulated as instances of goal-driven adaptive experimental design, wherein candidate experiments are chosen sequentially, with each choice informed by the outcomes of past experiments. The objective of this sequential decision-making process is to achieve a specific goal or learn about an unknown quantity of interest in a resource-efficient manner. Different goals lead to distinct instantiations of adaptive experimental design. For instance, in active learning, the goal is to iteratively select input examples to be labeled, with the goal of learning a predictor from a hypothesis class that achieves the highest accuracy with minimum number of labeled inputs. Similarly, in bandit problems (e.g., A/B testing and advertising), the learner repeatedly chooses an arm from a set of available options, aiming to maximize the cumulative reward over time.

In this dissertation, we focus on another important instantiation of adaptive experimental design: expensive black-box optimization over combinatorial spaces. In this problem setting, the goal is to identify the optimal input design within a large input design space consisting of discrete or hybrid structures, where the evaluation of each input design requires performing an experiment which is expensive in terms of the consumed resources (computational or physical). For example, in materials discovery, we are commonly interested in searching the space of candidate materials for a desired property while minimizing the total resource-cost of physical lab experiments for their evaluation. Remarkably, a large number of scientific discovery and engineering design problems including biological sequence design, nanoporous materials discovery, molecule optimization, and manycore systems design can be formulated as an instance of this general problem setting.

Bayesian optimization (BO) is an effective framework for tackling the challenge of expensive black-box optimization. The key idea behind BO is to learn a surrogate model from past experiments and use it intelligently select the sequence of experiments to find high-quality inputs by minimizing the number of experiments. In spite of the huge successes of BO, current approaches focus primarily on fixed-size continuous spaces and there is little principled work on combinatorial search spaces. Unlike continuous spaces, combinatorial spaces come with many unique challenges such as difficulty of defining a general representation, non-smoothness, etc. which require

specialized treatment for different types of structures (e.g., sequences, graphs, permutations etc). In this thesis, we explore and address the challenges of this new problem space by developing a series of novel approaches for Bayesian optimization over combinatorial spaces. First, we develop novel Gaussian process based surrogate models for a wide variety of combinatorial structures motivated by real-world applications (e.g., high-dimensional binary/categorical spaces, hybrid inputs containing a mixture of continuous and discrete variables, varying-sized graphs, and permutations.). Second, we developed a general tool for sampling functions from GP posteriors using explicit feature maps for discrete diffusion kernels referred to as Mercer features (analogous to random fourier features). Mercer features allow us to leverage advanced decision policies to select experiments in continuous spaces for discrete spaces. Third, we develop effective search strategies for large combinatorial spaces to select the candidate input with highest utility for experimental evaluation. For binary spaces, we showed a connection between optimization of Thompson sampling acquisition function with the binary quadratic optimization and derived an efficient submodular optimization approach. For permutation spaces, we derive a tractable approach with Thompson sampling by formulating it as a quadratic assignment problem. We also developed a general learning-to-search framework that allows using machine learning to improve the accuracy of search procedures to select inputs for evaluation. This framework is applicable for any complex surrogate model and acquisition function pair.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

Dedication

I dedicate this thesis to:

My mother Neelam, grandmothers (Ramkali/Sheela) and all Indian mothers who set aside their own dreams for their children through sacrifices, indomitable will and determination.

CHAPTER ONE

INTRODUCTION

The society is facing many sustainability challenges ranging from climate change to health, water, food, and manufacturing. Artificial intelligence (AI) and Machine learning (ML) holds tremendous promise towards addressing these challenges by accelerating discovery of new advances in science and engineering. For example, AI-driven rapid discovery of nanoporous materials have the potential to solve some of society's biggest challenges, from absorbing carbon dioxide from air to storing hydrogen gas for fuel. Similarly, AI-driven drug/vaccine design could help find significantly better therapeutics for major human ailments. The design of high performance and low-power hardware by overcoming Moore's law will enable sustainable computing from edge to cloud.

One important class of real-world problems commonly faced by scientists and engineers that lies at the heart of achieving these transformative goals is that of *(goal-directed) adaptive experimental design* where we need to select a sequence of experiments in order to achieve a goal[1]. In this thesis, we focus on this problem setting where the goal is to find the best design over a given input design space in order to optimize an objective of interest guided by expensive experiments. Commonly, evaluating each candidate input design requires performing an expensive physical

[1]Broadly, adaptive experimental design also includes areas such as bandits and active learning but this dissertation is only focused on expensive black-box function optimization problems.

lab-experiment or computational simulation. This process is iterative and adaptive, i.e., it is done over multiple rounds and the design of new experiments is conditioned on the measurements from previous experiments.

Consider the following design optimization problems motivated by science and engineering applications:

- Hardware design optimization, where evaluating each design involves performing a computationally-expensive simulation to emulate the real hardware.
- Material design optimization, where making and evaluating a candidate material involves performing an expensive physical lab experiment.
- Microbiome design optimization, where we need to perform an expensive physical experiment to evaluate each design in the form of a subset of microbes, their relative concentrations, and environmental conditions.

In all of these problems, the design space is very large and each candidate design is a discrete combinatorial structure (e.g., set, sequence, graph, permutation) or a hybrid structure (mixture of discrete and continuous design variables). For example, each hardware design can be seen as a discrete structure, where design variables correspond to the locations of processing elements (cores) and locations of the communication links for data transfer between cores. Similarly, each microbiome design can be seen as a hybrid structure, where discrete design variables correspond to the

subset of microbes and continuous design variables correspond to the relative concentrations of those microbes and the environmental conditions. Many design optimization problems in engineering and scientific domains including the above examples are instantiations of the following adaptive experimental design problem: *optimizing the design of discrete and hybrid structures guided by expensive experiments*, where expense is measured by the resources consumed by the experiments. These experiments are often performed in a heuristic manner by the humans without formally reasoning about the available (computational or physical) resource budget and the usefulness of potential information that they may provide.

This dissertation explores a new problem space of optimizing discrete and hybrid spaces via expensive evaluations. This problem space has very limited prior work. The SMAC algorithm [106], an instantiation of the Bayesian Optimization (BO) framework [192], is a canonical baseline. The key idea behind the BO framework is to build a cheap surrogate model (e.g., Gaussian Process) using the real experimental evaluations and employ it to intelligently select the sequence of experiments using an acquisition strategy guided by the surrogate model. In spite of the huge successes of BO [66, 83], current approaches focus primarily on continuous spaces and there is little principled work on discrete and hybrid spaces. Unlike continuous spaces, discrete spaces come with many unique challenges which form the key research questions addressed in this thesis [70]. We describe the challenges in detail below:

1.1 Key Research Challenges

- **Defining an effective surrogate model over combinatorial structures:** The first challenge lies in constructing a surrogate model that can accurately represent and predict the behavior of objective functions over combinatorial spaces in *small-data setting*. This task is considerably more complex than in continuous spaces because unlike continuous spaces, combinatorial spaces often lack a natural ordering, making it difficult to define concepts such as smoothness or continuity. This makes it important to consider specialized treatment of different types of combinatorial structures (sequences, graphs, permutations etc.). Moreover, in real-world domains, we typically need to deal with high-dimensional combinatorial spaces which further complicates the modeling challenge.

- **Optimizing acquisition functions over large combinatorial spaces:** In each step of Bayesian optimization, we need to optimize an acquisition/utlity function to select the most promising next structure to evaluate. First, in order to define many modern acquisition functions from the continuous BO literature for combinatorial spaces, we need to address the challenge of sampling from the surrogate model's posterior. For example, this is required in parallelizable approaches such as Thompson sampling [] and information-theoretic acquisition

functions such as input/output space entropy search []. Second, it is challenging to search over exponentially large combinatorial spaces since traditional continuous optimization techniques such as gradient descent are not applicable.

Additionally, these two challenges are related in the sense that there is an inherent trade-off between complexity of the surrogate model and tractability of acquisition function optimization.

1.2 Technical Contributions

The main contribution of this dissertation is the development and evaluation of a series of algorithmic ideas to address the above-mentioned challenges of Bayesian optimization (BO) over combinatorial spaces to significantly push the frontiers of adaptive experimental design research area. Specific contributions include:

- Developing new Gaussian Process (GP) based probabilistic models over different kinds of combinatorial structures:
  - *High-dimensional fixed size structures [68]:* We developed a novel dictionary embeddings based GP model for handling the challenge of high-dimensional binary/categorical spaces. We theoretically analyze this model to show that dictionary kernel reduces regret bounds for BO by reducing the cardinality of the search space. To the best of our knowledge, this is the first result showing a direct connection between GP modeling of

combinatorial structures and compressed sensing techniques.

  - *Hybrid/Mixed structures [63]:* We developed a GP surrogate model for hybrid spaces by designing a novel diffusion kernel via additive GP formulation. The representation power of kernel methods is typically studied in terms of the notion of universality: whether given sufficient data, the function class of the kernel can approximate any black-box function defined over hybrid spaces or not. We proved that this hybrid diffusion kernel satisfies this property of universality by composing a known result for continuous diffusion kernels with a novel result for discrete diffusion kernels.

  - *High-dimensional varying size structures [58]:* We developed the LADDER approach that combines kernels over latent space of a deep generative model (DGM) with expert-designed structured kernels to improve surrogate models. For example, kernels over morgan fingerprints (capture different substructures in a molecule while being invariant to atom relabeling) are a good example of structured kernels. This approach has the advantage that it can leverage any advances in both DGMs as well as structured kernels in a *plug-and-play* manner.

- Developing tractable and effective approaches for acquisition function optimization in specialized and general scenarios:

– We construct explicit feature maps for diffusion kernels over fixed-size binary design variables referred to as Mercer features [64]. They are analogous to random fourier features for continuous diffusion kernels and allow us to sample functions from Gaussian process posterior using the dual weight-space representation of GPs. Mercer features allow us to leverage advanced acquisition functions (e.g., Thompson sampling and input/output space entropy search) to select experiments in continuous spaces for discrete spaces. GPs parameterized in terms of mercer features allow us to study the trade-off between surrogate modeling and acquisition function optimization for binary (and categorical spaces via binary encoding). For example, we developed tractable submodular optimization approach for Thompson sampling acquisition function.

– We developed two algorithms for BO over permutation spaces with varying trade-offs between complexity of surrogate model and tractability of acquisition function optimization [67]. We also proved first regret bounds for the tractable algorithm based on Thompson sampling.

– We designed a general learning-to-search framework referred to as L2S-DISCO [61] for effectively solving acquisition function optimization problems involving arbitrarily complex surrogate models and acquisition functions. The key idea is to use machine learning to improve the accuracy of

search procedures to select structures for evaluation by leveraging search experience from both past and current BO iteration.

1.3 Outline of the Thesis

The remaining part of the dissertation is organized as follows:

In Chapter 2, we formally describe the overall problem setup and its different instantiations considered in this thesis. Next, we provide background on the generic Bayesian optimization (BO) framework and its key components. Finally, we describe the two key challenges in extending BO for continuous spaces to combinatorial spaces.

In Chapter 3, we address the challenge of defining surrogate models over high-dimensional fixed-size combinatorial inputs by constructing a novel dictionary embeddings based Gaussian Process model. We analyze its theoretical properties showing that the regret bounds for GP bandits trained on the embeddings is a function of measure of dictionary's variability similar to the mutual coherence property which is widely studied in compressed sensing. Our experiments on diverse real-world benchmarks from AutoML, satisfiability, pest-control domains demonstrate the effectiveness of the proposed surrogate model and corresponding BO performance.

In Chapter 4, we tackle the challenge of hybrid input spaces that represent a generalization of continuous and discrete spaces. We developed a GP surrogate model for hybrid spaces by designing a diffusion kernel via additive GP formulation. We theoretically analyze the modeling strength of additive hybrid kernels. We evaluate

the proposed method on diverse synthetic and real-world benchmarks showing that it significantly outperforms the state-of-the-art methods.

In Chapter 5, we consider optimization problems over the input space consisting of all permutations over d objects. We propose and evaluate two algorithms for Bayesian optimization over permutation spaces that make varying trade-offs between the complexity of statistical model and tractability of search for selecting permutations for evaluation. Our experiments on synthetic benchmarks and three important real-world applications from the domain of computer-aided design of integrated circuits (ICs) demonstrate the effectiveness of both algorithms.

In Chapter 6, we consider the problem space of richer varying sized combinatorial structures where we have additional access to a large database of unsupervised structures from the input space. We develop a Gaussian process model that synergistically combines the strengths of deep generative models and structured kernels. We demonstrate its effectiveness in BO on two real-world benchmarks of chemical design and arithmetic expressions optimization.

In Chapter 7, we revisit the problem space of fixed-size combinatorial inputs (concretely binary or categorical spaces). We develop Mercer features which are explicit feature map for discrete diffusion kernels. We describe how Mercer features allows efficient sampling from GP posteriors which is a key step in many modern acquisition function strategies. Furthermore, we develop a efficient and scalable submodular

optimization formulation of the Thompson Sampling based acquisition function optimization. Experimental results on multiple synthetic and real world benchmarks (e.g., biological sequence design tasks) demonstrate the effectiveness of this approach.

In Chapter 8, we focus on the problem of acquisition function optimization over general combinatorial spaces. We develop a general learning to search framework that works with any choice of the surrogate model and acquisition function. Experimental results on multiple benchmarks demonstrate the effective of this learned search strategy.

In the final Chapter 9, we provide a summary of the thesis' contributions and discuss some future directions to expand on this work.

CHAPTER TWO

PROBLEM SETUP AND BACKGROUND

In this chapter, we formally define the overall problem setting that we address in this dissertation and provide an overview of the generic Bayesian optimization (BO) framework along with the key challenges in extending BO to combinatorial spaces.

2.1 Problem Setup

Without loss of generality, let $\mathcal{X}$ be an input space. We are given an unknown real-valued objective function $f : \mathcal{X} \mapsto \mathbb{R}$, which can evaluate each input $\mathbf{x} \in \mathcal{X}$ to produce an evaluation $y = f(\mathbf{x})$. Each evaluation $f(\mathbf{x})$ is **expensive** in terms of the consumed resources. For example, in hardware design optimization application, $\mathbf{x}$ corresponds to a candidate hardware design with some choices to input variables (e.g., placement of cores and communication links), and $f(\mathbf{x})$ corresponds to running a computationally-expensive simulation to mimic the real hardware. Similarly, in nanoporous design optimization application, $\mathbf{x}$ corresponds to a candidate nanoporous material (e.g., selection of topology, inorganic nodes, and organic linkers), and $f(\mathbf{x})$ corresponds to making the material in a physical lab and measuring its adsorption property.

The main goal is to find an input $\mathbf{x}^* \in \mathcal{X}$ that approximately optimizes f by performing a limited number of function evaluations. In this work, *unlike* the commonly studied setting of continuous input spaces ($\mathcal{X} \subseteq \mathbb{R}^d$), we focus on the problem settings

where the input space $\mathcal{X}$ consists of combinatorial structures (e.g., fixed-size structures such as sequences, permutations and varying-size structures such as graphs). Some example combinatorial spaces that we study in this thesis include:

1. $\mathcal{X}$ represents the space of all binary structures $\{0,1\}^d$ where d is the number of the variables. For example, feature selection in automated machine learning (AutoML) can be formulated by a binary search space where the inclusion/exclusion of a given feature can be represented by a binary parameter.

2. $\mathcal{X}$ represents the categorical space $\{0,k\}^d$ where d is the number of the variables and k is the number of candidate values each variable can take. For example, each candidate input in manycore systems design can be seen as a categorical sequence where each design variable corresponds to a choice from a fixed set of processing elements (CPUs, GPUs, TPUs, domain-specific accelerator).

3. $\mathcal{X} = \mathcal{S}_d$ represents the space of all permutations of a given set of d objects. Given $[1,d] := \{1,2,\cdots,d\}$, indexing the d objects, a permutation is defined as a bijective mapping $\pi : [1,d] \mapsto [1,d]$. We refer to the set of all permutations of d objects by $\mathcal{S}_d$. For example, in the design of integrated circuits (ICs), we need to find the best placement of a set of functional blocks which requires searching over the space of permutations of the blocks.

4. $\mathcal{X} = \mathcal{X}_d \times \mathcal{X}_c$ represents the space of hybrid structures where $x = (x_d \in \mathcal{X}_d, x_c \in$

$\mathcal{X}_c) \in \mathcal{X}$ be represented using m discrete variables and n continuous variables, where x_d and x_c stands for the discrete and continuous sub-space of $\mathcal{X}$. For example, in microbiome analysis, the inclusion/exclusion of a microbial species is a binary parameter and environmental variables correspond to continuous parameters making it a hybrid search space.

5. $\mathcal{X}$ represents the space of varying sized combinatorial structures (trees, graphs). For example, in drug design application, each candidate molecule can be represented as a graph.

2.2 Bayesian Optimization Background

Bayesian Optimization (BO) is an effective framework for solving global optimization problems using *black-box evaluations of expensive objective functions*. BO algorithms learn a cheap surrogate model from training data obtained from past function evaluations. They intelligently select the next input for evaluation by trading-off exploration (inputs for which the statistical model has high uncertainty) and exploitation (inputs for which the model has high prediction value) to quickly direct the search towards optimal inputs. The three key elements of BO framework are:

1) Statistical Model of the true function $f(x)$. *Gaussian Process (GP)* [220] is the most commonly used model. It is characterized by a mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$ and a covariance or kernel function $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. The kernel is an important object

that characterizes the space of functions represented by GP models. For example, two popular kernels for continuous input spaces are given below:

- Squared Exponential Kernel (also known as Radial Basis Function (RBF) Kernel):

$$k(x, x') = \exp(-\frac{d(x, x')^2}{2l^2})$$

- Matern 5/2 Kernel

$$k(x, x') = (1 + \sqrt{5 \cdot d(x, x')^2} + \frac{5}{3} \cdot d(x, x')^2) \cdot \exp\left(-\sqrt{5 \cdot d(x, x')^2}\right)$$

where $d(x, x')$ is the euclidean distance between two inputs x and x' and l is a length-scale hyper-parameter of the kernel.

2) Acquisition Function (AF) to score the utility of evaluating a candidate input $\mathbf{x} \in \mathcal{X}$ based on the statistical model. Some popular acquisition functions in the BO literature include expected improvement (EI), upper confidence bound (UCB), lower confidence bound (LCB), Thompson sampling (TS), and max-value entropy search (MES) [216]. For the sake of completeness, we formally define some

of the acquisition functions:

$$UCB(x) = \mu(x) + \beta^{1/2}\sigma(x) \tag{2.1}$$

$$TS(x) = f(x)\ with\ f(.) \sim GP \tag{2.2}$$

$$EI(x) = \sigma(x)(\gamma\Phi(\gamma) + \phi(\gamma)),\ \gamma = \frac{\tau - \mu(x)}{\sigma(x)} \tag{2.3}$$

There are also reduction-style acquisition functions for multi-objective BO [24, 3] building on the above single-objective acquisition functions.

3) Optimization Procedure to select the best scoring candidate input according to acquisition function AF depending on the statistical model. For continuous input spaces, first and second order gradient based algorithms like L-BFGS and hierarchical partitioning algorithms like DIRECT [120] are commonly employed for acquisition function optimization.

2.3 Key Challenges for Combinatorial Spaces

There are two key challenges in employing Bayesian optimization framework for combinatorial input spaces. The first challenge is to define an effective surrogate model over combinatorial structures. The second challenge is, given such a surrogate model, to search through the combinatorial space to identify the most promising next structure to evaluate, i.e., acquisition strategy. Each of our proposed approach tackles

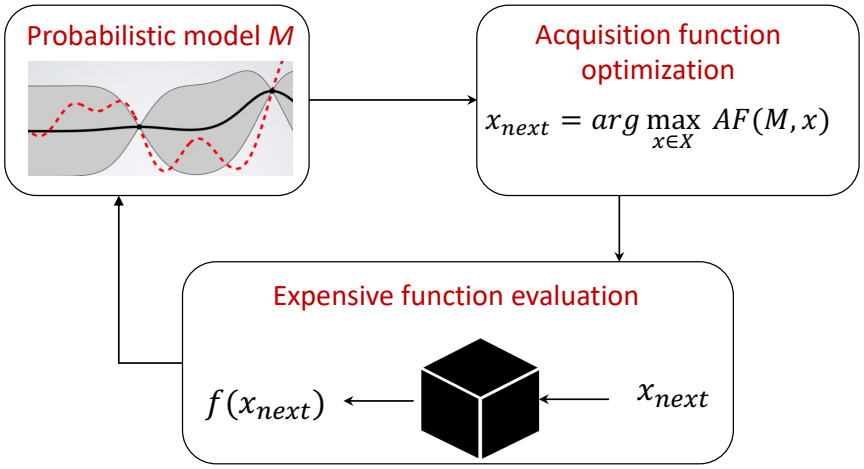


Figure 2.1: Overview of key steps of the Bayesian optimization procedure.

one or both of these challenges for different combinatorial spaces.

2.4 Related Work

There was limited prior work on this problem space before this dissertation. However, since the starting of this thesis' work, multiple methods have been proposed in the literature for solving similar problems. With this context, we describe the related work in this section. We will divide the related work along the two key challenges mentioned above: surrogate modeling and acquisition function optimization.

2.4.1 Surrogate Modeling on Combinatorial Structures

Fixed-size structures: First, we focus on the surrogate models that have been proposed for the low-dimensional setting. These models, however, are typically not

effective for high-dimensional spaces. BOCS [17] targets binary spaces and employs a second-order Bayesian linear regression surrogate model, which exhibits poor scaling in the input dimension and may not support applications where the underlying black-box function requires a more complex model. SMAC [106] employs random forest as the surrogate model which naturally handles both discrete and continuous inputs. However, random forest are prone to poor uncertainty estimation especially in small-data setting [130]. Prior work also considers different instantiations of Gaussian Process (GP) models which provide better uncertainty estimation. COMBO [169] employs GPs with discrete diffusion kernels over a combinatorial graph representation of the input space. Garrido-Merchán and Hernández-Lobato [89] round the input variables before passing it to a GP with a canonical kernel. Kim et al. [131] proposed an approach for combinatorial spaces based on continuous embeddings. Papalexopoulos et al. [173] employ a feed-forward neural network as surrogate model.

As opposed to continuous spaces which has seen a considerably large amount of work [79, 218, 142, 174, 163, 78, 88, 124, 84], there is still limited work on BO over fixed-size high-dimensional combinatorial inputs. CASMOPOLITAN [215], uses adaptive trust regions from continuous spaces [79] by replacing the standard Euclidean distance with Hamming distance for discrete (sub)spaces. Bounce [175] is a recent approach that uses nested-embeddings along with trust region search to scale BO to high-dimensional combinatorial sequences.

Hybrid structures: There is also related work on approaches for constructing surrogate models over hybrid spaces with both discrete and continuous variables. Tree-Parzen Estimators (TPEs) [29] are applicable to hybrid spaces and consider density estimation in the input space which is potentially challenging in high-dimensional settings. CoCaBO [184] employs a sum kernel (summing a Hamming kernel over discrete subspace and a RBF kernel over continuous subspace) to learn GP models and showed good results over SMAC and TPE. Unfortunately, the sum kernel captures limited interactions between discrete and continuous variables. Oh et al. [170] extends COMBO's graph representation and leverages distances on continuous variables to modulate the graph Fourier spectrum in order to couple the two types of sub-spaces.
Varying-size structures: Moss et al. [157] propose using Gaussian processes with string kernels [145] for BO that are naturally applicable to varying sized sequences. Most approaches for richer varying-sized structures (for e.g. graph inputs) rely on using deep generative models to create a latent space and apply continuous BO methods (often referred to as "latent space BO"): [91, 210, 77, 122, 165, 153]. One key assumption in all these methods is availability of a large dataset of unsupervised structures which is required to learn the deep generative model.

2.4.2 Acquisition Function Optimization (AFO) over Combinatorial Structures

As described earlier, optimizing the acquisition function (sometimes referred as the inner-loop problem) over combinatorial inputs is itself a hard challenge. A popular

approach for fixed-size discrete spaces is greedy-hill climbing local search. The base local search procedure is usually augmented with a combination of multiple random restarts or simulated annealing [169, 49]. This can be naturally extended to hybrid search spaces by performing alternating search over continuous and discrete subspaces [170, 215]. BOCS [16] linear model allow semi-definite formulation of the AFO problem which can be handled by specialized semi-definite programming (SDP) solvers. Papalexopoulos et al. [173] develop a mixed integer linear programming formulation exploiting piecewise-linear activation functions like RELU in the neural network surrogates. Most latent space BO approaches can naturally use gradient-based methods in the continuous embedding space of the deep generative model.

CHAPTER THREE

DICTIONARY-BASED SURROGATE MODEL FOR HIGH-DIMENSIONAL COMBINATORIAL SPACES

In this chapter, we consider the problem of optimizing expensive black-box functions over fixed-size **high-dimensional combinatorial spaces** which arises in many science, engineering, and ML applications. We propose a novel surrogate modeling approach for efficiently handling a large number of binary and categorical parameters. The key idea is to select a number of discrete structures from the input space (the dictionary) and use them to define an ordinal embedding for high-dimensional combinatorial structures. This allows us to use existing Gaussian process models for continuous spaces. We develop a principled approach based on binary wavelets to construct dictionaries for binary spaces, and propose a randomized construction method that generalizes to categorical spaces. We provide theoretical justification to support the effectiveness of the dictionary-based embeddings. Our experiments on diverse real-world benchmarks demonstrate the effectiveness of our proposed surrogate modeling approach over state-of-the-art BO methods.

3.1 Problem Setup

We are given a *high-dimensional* combinatorial space $\mathcal{X}$, i.e., the number of discrete variables d is large. We assume we are optimizing a black-box objective function

$f : \mathcal{X} \mapsto \mathbb{R}$, which we can evaluate on each structure $\mathbf{x} \in \mathcal{X}$. For example, in feature selection for Auto ML tasks, $\mathbf{x}$ is a binary structure corresponding to a subset of features and $f(\mathbf{x})$ is the performance of a trained ML model using the selected features. Our goal is to find a structure $\mathbf{x} \in \mathcal{X}$ that approximately optimizes f given a small number of function evaluations.

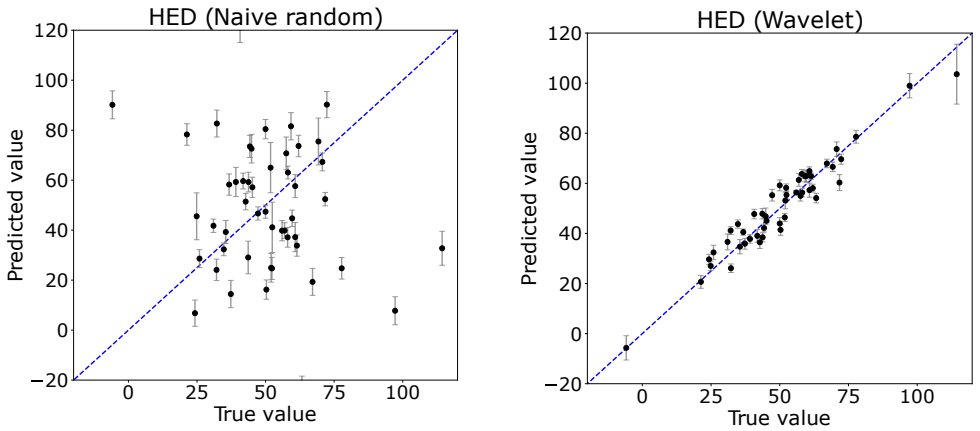
3.2 Dictionary Embeddings

In this section, we introduce the idea of a **H**amming **e**mbedding via **d**ictionaries (HED), a novel embedding for binary and categorical inputs that embeds the inputs into an ordinal feature space. In particular, we employ a GP over the embedding $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$ based on a dictionary $\mathbf{A}$ containing m discrete d-dimensional elements from the input space $\mathcal{X}$. The embedding $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$ of size m is obtained by computing the Hamming distance h between $\mathbf{x} \in \mathcal{X}$ and each element of the dictionary $\mathbf{a}_i \in \mathbf{A}$. That is,

$$[\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})]_i = h(\mathbf{a}_i, \mathbf{x}).$$

HED has several advantages. First, it allows us to transform the challenging task of building models over high-dimensional discrete spaces into an application of GPs to the well-understood continuous space settings. This subsequently allows us to perform inference of lengthscales associated with the embedding representations, in contrast to the original categorical space where one lengthscale models the effect

of a single category change. Further, the efficient inference of lengthscales due to the embedding enables Automatic Relevance Determination (ARD) to prune away redundant dimensions effectively, which we prove reduces the cardinality of the input space. We show theoretically that this improves the sample-efficiency of GP bandits (UCB), commonly used for BO, and produces state-of-the-art results for BO on high-dimensional combinatorial spaces. Further, while the core kernel is for binary spaces, it can easily be extended to hybrid spaces with both continuous and discrete parameters by using a product kernel.



(a) Naïve random dictionary **(b)** Binary wavelet dictionary

Figure 3.1: Mean predictions and associated 95% predictive intervals on a MaxSAT problem with 60 binary variables (see details in Sec. 3.6), comparing naïve random (left) and binary wavelet (right) dictionaries, using 50 training points and predicting on 50 test points.

3.3 Dictionary Construction Procedure

The effectiveness of HED depends on the dictionary construction. A naïve approach is to simply pick elements from the binary space uniformly at random. However, this naïve approach turns out to exhibit poor predictive or BO performance on the test problems considered in this work. For example, Fig. 3.1a illustrates the poor predictive performance of a GP using a dictionary kernel with a uniformly random binary dictionary on a MaxSAT test problem with 60 binary variables.

Another idea is to use deterministic dictionary construction methods, such as multi-resolution *wavelets* [147], effective and well-known tools for studying real-valued signals at different scales by applying a set of orthogonal transforms to the data. In the context of binary spaces, binary wavelet transforms [205] are highly related to the well-known orthogonal Hadamard matrices, and are applied in signal processing, spectroscopy, and cryptography [98, 103]. In contrast to the naïve random dictionary, sub-sampled binary wavelet dictionaries lead to great predictive performance on the same MaxSAT problem, as shown in Fig. 3.1b.

3.3.1 Binary Wavelet Dictionaries

Now, we describe the randomized dictionary construction approach based on Binary wavelet transform for binary spaces $\mathcal{X}=\{0,1\}^d$. At a high-level, this approach has two key steps. First, we employ a deterministic recursive procedure to construct

a pool of basis vectors over binary structures. Second, we randomly select a subset of k diverse vectors as our dictionary $\mathbf{A}$. We explain the details of these two steps below.

Recursive algorithm for binary wavelet design. The effectiveness of surrogate model critically depends on the dictionary employed to embed the discrete inputs. We define our dictionary matrix $\mathbf{A}_{[k\times d]}$ as a subsampled (k-sized) set of basis vectors over the binary space $\{0,1\}^d$ which is characterized by the constituent vectors varying over a range of sequencies. The notion of *sequency* is defined as the number of changes from 1 to 0 and vice versa (analogous to the notion of frequency in Fourier transforms).

Multi-resolution *wavelets* [147] are effective well-known techniques for studying real-valued signals at different scales by applying a set of orthogonal transforms to the data. Specifically, binary wavelet transforms [205] allow us to study data defined over binary spaces (concretely $\{0,1\}^d$ with *mod 2* arithmetic) at different scales. Hence, they are a natural choice for constructing our pool of basis vectors.

We construct the randomized dictionary $\mathbf{A}$ by randomly sampling from a deterministic binary wavelet transform matrix $\mathbf{B}_d$ generated by a recursive procedure as described in [205] (where such matrices were used for image compression). The key idea behind the procedure is to recursively generate binary matrices whose vectors are ordered in terms of increasing sequency. Algorithm 1 provides the pseudo-code of this recursive method.

Algorithm 1 BINARY WAVELET (n) Transform

requires: input dimension n

1: if $n == 2$: **return** $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

2: if $n == 4$: **return** $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

3: B_{n-4}= BINARY WAVELET (n-4)

4: Compute upper left $n-2 \times n-2$ matrix Γ

$$\Gamma = \begin{bmatrix} \mathbf{1}_{[2,2]} & \mathbf{1}_{[2,n-4]} \\ \mathbf{1}_{[n-4,2]} & \neg B_{n-4} \end{bmatrix}$$

5: Set lower left block $\Delta^T \leftarrow \begin{bmatrix} 1 & 0 & 1 & \cdots \\ 1 & 0 & 1 & \cdots \end{bmatrix}$

6: Set lower right block $\Lambda \leftarrow \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

7: **return** $B_n = \begin{bmatrix} \Gamma & \Delta \\ \Delta^T & \Lambda \end{bmatrix}$

Given $\mathbf{B}_d$, the dictionary $\mathbf{A}$ is constructed by subsampling row vectors from $\mathbf{B}_d$ i.e. $\mathbf{A} = \mathbf{P}\mathbf{B}_d$ where $\mathbf{P}$ randomly samples m vectors uniformly. The random sampling using $\mathbf{P}$ picks vectors that are spread over a range of sequencies in contrast to the alternative choice of picking top-m rows from $\mathbf{B}_d$ which restricts the chosen vectors to limited range of sequencies. Our experiments demonstrate the effectiveness of randomized dictionaries over the top-m alternative.

While binary wavelets constitute powerful dictionary designs for predictive and optimization problems in binary search spaces (for associated optimization results, see Fig. 3.4), their construction for non powers-of-two is non-trivial, and even their existence for arbitrary dimensions is an open problem [95, 19, 69]. For this reason,

we sub-sample the columns of the power-of-two dimensional binary wavelets for our experiments in non-power-of-two dimensions.

3.3.2 Diverse Random Dictionaries

To alleviate the difficulties around the general construction of binary wavelets, and to generalize our method to categorical spaces, we propose a randomized procedure that produces dictionary rows with a large range of sparsity levels. We refer to this randomized procedure as "diverse random."

Algorithm 2 provides pseudo-code for constructing diverse random dictionaries defined over binary input spaces $\{0,1\}^d$. The key principle of this construction procedure is to diversify the dictionary rows by generating binary vectors determined by different bias parameters (θ) of the Bernoulli distribution, unlike the naïve random where θ is always $1/2$. Therefore, the rows of the naïve random dictionaries tend to have close to $d/2$ non-zeros as d grows, whereas the diverse random dictionaries exhibit a large range of sparsity levels due to varying θ. This algorithm can easily be generalized to inputs with categorical variables of different sizes, see Algorithm 3 for details. To summarize, the diverse random dictionaries can be constructed for arbitrary dimensions, extends naturally to categorical inputs, and as we will show later exhibits strong optimization performance on a wide range of benchmark problems.

Algorithm 2 Dictionary design for binary input space $\{0,1\}^d$ with diversely sparse rows

requires: dictionary size m

Dictionary $\mathbf{A} \leftarrow$ empty
for i=1, 2, . . . , m **do**
  $\mathbf{a}_i \leftarrow$ empty
  Sample Bernoulli parameter $\theta \sim$ Uniform$(0, 1)$
  for j=1, 2, . . . , d **do**
    Sample binary number $a \sim$ Bernoulli(θ)
    $\mathbf{a}_i \leftarrow \mathbf{a}_i \cup a$
  end for
  Add $\mathbf{a}_i$ to dictionary: $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{a}_i$
end for
return the dictionary $\mathbf{A}$ of size $m \times d$

Algorithm 3 Dictionary design for discrete spaces with categorical variables via diverse parameters

Input: candidate sets $C(v_1), \ldots, C(v_d)$, dictionary size m **Output**: the dictionary $\mathbf{A}$ of size $m \times d$

Dictionary $\mathbf{A} \leftarrow$ empty
$\tau_{max} \leftarrow \max_j \tau_j$
for i=1, 2, . . . , m **do**
  $\mathbf{a}_i \leftarrow$ empty
  Sample $\boldsymbol{\theta} \sim \Delta^{\tau_{max}}$
  for j=1, 2, . . . , d **do**
    $\boldsymbol{\theta}_j \leftarrow$ sample (w/o repl.) τ_j elements from $\boldsymbol{\theta}$
    $\boldsymbol{\theta}_j \leftarrow \boldsymbol{\theta}_j / \|\boldsymbol{\theta}_j\|_1$ (Normalize to yield distribution)
    $a \leftarrow$ sample from $C(v_j)$ with probabilities $\boldsymbol{\theta}_j$
    $\mathbf{a}_i \leftarrow \mathbf{a}_i \cup a$
  end for
  Add $\mathbf{a}_i$ to dictionary: $\mathbf{A} \leftarrow \mathbf{A} \cup \mathbf{a}_i$
end for

Representation of hybrid input spaces. We have focused on a purely combinatorial input spaces $\mathcal{X}$, but can naturally extend our approach to hybrid search spaces consisting of both discrete and continuous parameters. In this setting, we aim

to model an input space $\mathcal{X}_d \times \mathcal{X}_c$ where each x in this space is represented by $x = (x_d \in \mathcal{X}_d, x_c \in \mathcal{X}_c)$ where x_d and x_c stands for the discrete and continuous parameters. For notational convenience, we will keep referring to the hybrid space $\mathcal{X}_d \times \mathcal{X}_c$ as $\mathcal{X}$. To extend our approach to this hybrid inputs setting, we use a product kernel leveraging the HED embedding for discrete parameters and a standard, e.g., Matérn-5/2 kernel with ARD for the continuous parameters.

3.4 BODi: Bayesian Optimization with Dictionary Embeddings

Our proposed BODi method is a straightforward instantiation of the generic BO framework. We use a GP with a standard Matérn-5/2 kernel with ARD on the HED embedding as the surrogate model, and we adopt the commonly used Expected Improvement (EI) acquisition function for single-objective problems. In our setting, EI takes as inputs the surrogate model $\mathcal{M}$ and the embedding $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$ to score the utility of evaluating the structure $\mathbf{x} \in \mathcal{X}$. In order to optimize the acquisition function over the discrete space $\mathcal{X}$, we employ local search from randomly generated initial conditions.

Algorithm 4 shows the pseudo-code of our method. We use a small random initial training set of elements in $\mathcal{X}$ and their function evaluations to construct an initial surrogate model $\mathcal{M}(\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}))$. We generate a new dictionary $\mathbf{A}$ in each BO iteration using a randomized procedure described in Alg. 2, and refit the GP model using the corresponding embedding $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$. For each BO iteration j, we select the next structure

$\mathbf{x}_j$ by optimizing the acquisition function. We add $\mathbf{x}_j$ and the corresponding function value $f(\mathbf{x}_j)$ to the training data D_j and train a new surrogate model $\mathcal{M}(\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}))$ using D_j. We repeat these steps until the query budget is exhausted and return the best input $\mathbf{x}_{\text{best}} \in \mathcal{X}$.

Algorithm 4 `BODi` (m) Algorithm

requires: black-box objective f, discrete space $\mathcal{X}$ with dimensionality d, dictionary size m

$D_0 \leftarrow$ small random initial training data
for $j = 1, 2, \ldots$ **do**
  Construct dictionary $\mathbf{A}$ of size m
  Compute low-dimensional embedding $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$ for
    each input structure $\mathbf{x} \in D_j$ using dictionary $\mathbf{A}$
  Fit a GP $\mathcal{M}$ on the embedded space $\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})$
  Maximize the acquisition function in the discrete space $\mathcal{X}$: $\mathbf{x}_j = \arg\max_{\mathbf{x} \in \mathcal{X}} \alpha(\mathcal{M}(\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x})))$
  Evaluate the selected structure $\mathbf{x}_j$ to get $f(\mathbf{x}_j)$
  Aggregate training data: $D_j \leftarrow D_{j-1} \cup \{\mathbf{x}_j, f(\mathbf{x}_j)\}$
end for
return $\mathbf{x}_{best} = \arg\min\{f(\mathbf{x}_1), f(\mathbf{x}_2) \cdots\}$

To optimize the acquisition function over hybrid search spaces, we perform alternating search over continuous and discrete subspaces, a common approach in BO over hybrid spaces [170, 65, 215]. We use local search for discrete parameters and gradient-based optimization for continuous parameters. While acquisition function optimization over discrete spaces is a challenging problem, local search with restarts has been shown to be effective in practice [169].

3.5 Theoretical Analysis of BODi

In the following, we derive a surprising relationship for the Hamming embedding with an affine transformation, explaining why canonical linear embeddings (e.g. Gaussian) do not perform well. We also provide a regret bound for BO with the dictionary kernel that crucially relies on a reduction in the cardinality – not the dimensionality – of the embedded search space. Our results are stated for binary search spaces, but can be readily generalized to categorical variables using a binary encoding, e.g., one-hot encoding, or more efficiently with $\lceil \log_2(c) \rceil$ bits for c categories.

Our first proposition shows that the Hamming embedding of vectors in $\{0,1\}^d$ is equivalent to an affine transformation of the $\{\pm 1\}$-encoding of the binary vector.

Proposition 1 (Affine Representation)**.** *Let* $\mathbf{A} \in \{0,1\}^{m \times d}$*,* $\mathbf{x} \in \{0,1\}^d$*. Then*

$$2\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) = d\mathbf{1}_m - \bar{\mathbf{A}}\bar{\mathbf{x}}, \tag{3.1}$$

where $\bar{a}_{ij} = 2a_{ij} - 1$ *and* $\bar{x}_i = 2x_i - 1 \in \{-1, 1\}$.

Proof. Our first proposition shows that the Hamming embedding of vectors in $\{0,1\}^d$ is equivalent to an affine transformation of the $\{-1,1\}$-encoding of the original binary vector.

Proposition 1 (Affine Representation)**.** *Let* $\mathbf{A} \in \{0,1\}^{n \times d}$*,* $\mathbf{x} \in \{0,1\}^d$*. Then*

$$2\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) = d\mathbf{1}_n - \bar{\mathbf{A}}\bar{\mathbf{x}}, \tag{3.2}$$

where $\bar{A}_{ij} = 2A_{ij} - 1$ *and* $\bar{x}_i = 2x_i - 1 \in \{-1, 1\}$*.*

Proof. Let $\mathbf{a}_i$ be the ith column in $\mathbf{A}$, and x_i the ith entry of $\mathbf{x}$. Then

$$\begin{aligned}
\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) &= \sum_i^d (\neg\mathbf{a}_i x_i + \mathbf{a}_i \neg x_i) \\
&= \sum_i^d ([\mathbf{1}_n - \mathbf{a}_i]x_i + \mathbf{a}_i[1 - x_i]) \\
&= \sum_i^d (\mathbf{1}_n x_i - 2\mathbf{a}_i x_i + \mathbf{a}_i) \\
&= \mathbf{1}_n(\mathbf{1}_d^\top \mathbf{x}) - \mathbf{A}(2\mathbf{x} - 1) \\
&= [(\mathbf{1}_n \mathbf{1}_d^\top)(2\mathbf{x} - 1) + d\mathbf{1}_n]/2 - \mathbf{A}(2\mathbf{x} - 1) \\
&= [d\mathbf{1}_n - (2\mathbf{A} - \mathbf{1}_{n,d})(2\mathbf{x} - \mathbf{1}_d)]/2 \\
&= (d\mathbf{1}_n - \bar{\mathbf{A}}\bar{\mathbf{x}})/2.
\end{aligned}$$

Multiplying both sides by two finishes the proof. □

Plugging the affine representation into the embedded distance formula yields

$$\begin{aligned} 2\|\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) - \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}')\| &= \|(d\mathbf{1}_n - \bar{\mathbf{A}}\bar{\mathbf{x}}) - (d\mathbf{1}_n - \bar{\mathbf{A}}\bar{\mathbf{x}}')\| \\ &= \|\bar{\mathbf{A}}\bar{\mathbf{x}} - \bar{\mathbf{A}}\bar{\mathbf{x}}'\| \\ &= \|\bar{\mathbf{A}}\bar{\mathbf{r}}\|, \end{aligned}$$

where $\bar{\mathbf{r}} = \bar{\mathbf{x}} - \bar{\mathbf{x}}'$. That is, the distance computation only relies on a *linear* projection of the difference vector $\bar{\mathbf{r}}$ of the $\{-1, 1\}$-encoding of the centered input vectors. As a further consequence, if the wavelet dictionary is chosen, the embedding is a sub-sampled Hadamard transform up to a constant shift, which we could implement by means of the Fast Hadamard Transform in $d \log d$ time. □

Plugging Eq. (3.1) into the embedded distance formula yields

$$2\|\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) - \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}')\|_2 = \|\bar{\mathbf{A}}\bar{\mathbf{r}}\|_2,$$

where $\bar{\mathbf{r}} = \bar{\mathbf{x}} - \bar{\mathbf{x}}'$. That is, the distance computation only relies on a *linear* projection of the difference vector $\bar{\mathbf{r}}$ of the $\{\pm 1\}$-encoding of the binary input vectors. Furthermore, the embedding associated with the wavelet dictionary described is thus equivalent up to a constant shift to a sub-sampled Hadamard transform, a type of Fourier transform on Boolean fields.

Proposition 1 proves the equivalence of the dictionary-based kernel to a canonical

kernel (e.g. Matérn) evaluated on linearly projected input data. Given the significant prior work on BO on subspaces [218, 142] and on properties of linear projections [139], one might assume that canonical linear embedding designs like Gaussian random matrices will perform well in our setting. However, this is not the case, as we demonstrate in the empirical evaluation.

To understand why, first note that BODi is effectively carrying out the optimization in the transformed search space

$$\mathcal{S}_{\mathbf{A}} = \left\{ \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) \mid \mathbf{x} \in \{0,1\}^d \right\}.$$

While linear embeddings generally reduce the *dimensionality* of the search space, they do not necessarily lead to a reduction in the *cardinality* $|\mathcal{S}_{\mathbf{A}}|$, a key quantity in regret bounds for BO in finite search spaces. Indeed, while Gaussian random projections satisfy many desirable properties, including approximate distance preservation and dimensionality reduction, our next result shows that even a one-dimensional Gaussian random projection preserves the full cardinality of the original search space almost surely.

Proposition 2. *Define* $\mathcal{S}_{\mathbf{a}} = \{\mathbf{a}^\top \mathbf{x} \mid \mathbf{x} \in \{\pm 1\}^d\}$*, and let* $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$*. Then* $|\mathcal{S}_{\mathbf{a}}| = 2^d$ *almost surely.*

Proof. Given $\mathbf{x}, \mathbf{x}' \in \{-1, 1\}^d$, suppose $\mathbf{x} \neq \mathbf{x}'$ and $\mathbf{a}^\top \mathbf{x} = \mathbf{a}^\top \mathbf{x}'$. Therefore, $\mathbf{a}^\top (\mathbf{x} -$

$\mathbf{x}') = 0$. Since $\mathbf{x} - \mathbf{x}' \neq 0$, this can only hold if $\mathbf{a} \perp (\mathbf{x} - \mathbf{x}')$. But $\{\mathbf{a} \mid \mathbf{a} \perp (\mathbf{x} - \mathbf{x}')\}$ is $(d-1)$-dimensional, and therefore a nullset under the Gaussian measure in d dimensions [186]. Therefore, $\mathbf{a}^\top(\mathbf{x} - \mathbf{x}') \neq 0$ almost surely. Since the set $\{-1, 1\}^d$ has finite cardinality 2^d, and by the subaddativity of any probability measure μ,

$$\mu\left(\bigcup_{\mathbf{x},\mathbf{x}' \in \{-1,1\}^d} \{\mathbf{a} \mid \mathbf{a} \perp (\mathbf{x} - \mathbf{x}') = 0\}\right) \leq \sum_{\mathbf{x},\mathbf{x}' \in \{-1,1\}^d} \mu\left(\{\mathbf{a} \mid \mathbf{a} \perp (\mathbf{x} - \mathbf{x}') = 0\}\right) = 0.$$

Thus, all distinct $\mathbf{x} \in \{-1, 1\}^d$ map to distinct values $\mathbf{a}^\top\mathbf{x}$ almost surely, so $|\mathcal{S}_\mathbf{a}| = |\{-1, 1\}|^d = 2^d$. □

In contrast, our next result presents a bound on the cardinality of $\mathcal{S}_\mathbf{A}$ that depends on a measure of the variability $\mu_\mathbf{A}$ of the dictionary rows and grows only polynomially with d.

Proposition 3 (Embedding Cardinality)**.** *Let $\mathbf{A} \in \{0, 1\}^{m \times d}$. Then the cardinality of the embedded search space $\mathcal{S}_\mathbf{A}$ can be bounded above by*

$$|\mathcal{S}_\mathbf{A}| \leq [(\mu_\mathbf{A} + 1)(d + 1 - \mu_\mathbf{A})]^{\lfloor m/2 \rfloor} (d + 1)^{m \bmod 2}$$

where $\mu_\mathbf{A} = \max_{i,j} \max(h(\mathbf{a}_i, \mathbf{a}_j), h(\neg\mathbf{a}_i, \mathbf{a}_j))$, and h is the Hamming distance.

Proof. First, we consider one anchor point. Let $d \in \mathbb{N}$, and $\mathbf{a} \in \mathcal{B}^d$. Then for any

$\mathbf{x} \in \mathcal{B}^d$, $\boldsymbol{\phi}(\mathbf{a}, \mathbf{x}) \in \mathbb{N}$ and

$$0 \leq \boldsymbol{\phi}_{\mathbf{a}}(\mathbf{x}) = h(\mathbf{a}, \mathbf{x}) = \sum_i \delta(a_i, x_i) \leq d,$$

so $\boldsymbol{\phi}_{\mathbf{a}}(\mathbf{x}) \in [d]$ and $|\mathcal{S}| = d + 1$. Naïvely generalizing this to n dimensions would yield $|\mathcal{S}| \leq (d+1)^n$. However, the true cardinality is much lower, because having certain elements in common with one anchor point will restrict the corresponding dimensions to be the same with another anchor point. The next paragraph will make this intuition precise.

Next, we consider two anchor points. Let $d \in \mathbb{N}$, and $\mathbf{a}_1, \mathbf{a}_2 \in \mathcal{B}^d$. Then for any $\mathbf{x} \in \mathcal{B}^d$, Suppose $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2]$, and let

$$s = \{i \in [d] \mid [\mathbf{a}_1]_i = [\mathbf{a}_2]_i\}$$

be the set of indices for which the anchors have take the same values, and $\neg s = [d] \backslash s$,

$|s| = k$. Then we can express the embedding as

$$\begin{aligned}
\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}) &= \boldsymbol{\phi}_{\mathbf{A}_s}(\mathbf{x}_s) + \boldsymbol{\phi}_{\mathbf{A}_{\neg s}}(\mathbf{x}_{\neg s}) \\
&= [h(\mathbf{a}_{1,s}, \mathbf{x}_s), h(\mathbf{a}_{2,s}, \mathbf{x}_s)] + [h(\mathbf{a}_{1,\neg s}, \mathbf{x}_{\neg s}), h(\mathbf{a}_{2,\neg s}, \mathbf{x}_{\neg s})] \\
&= [h(\mathbf{a}_{1,s}, \mathbf{x}_s), h(\mathbf{a}_{1,s}, \mathbf{x}_s)] + [h(\mathbf{a}_{1,\neg s}, \mathbf{x}_{\neg s}), h(\neg\mathbf{a}_{1,\neg s}, \mathbf{x}_{\neg s})] \\
&= [x_s, x_s] + [h(\mathbf{a}_{1,\neg s}, \mathbf{x}_{\neg s}), (d - h(\mathbf{a}_{1,\neg s}, \mathbf{x}_{\neg s}))] \\
&= [x_s, x_s] + [x_{\neg s}, (d - x_{\neg s})],
\end{aligned}$$

where $x_s = h(\mathbf{a}_{1,s}, \mathbf{x}_s)$. Now, $n_s = h(\mathbf{a}_s, \mathbf{x}_s) \in [k]$ and $n_{\neg s} \in [d-k]$. The cardinality of the embedding space is exactly $(k+1)(d+1-k)$, because a subset of $d-k$ variables always take the same values in both dimensions, and the remaining k move linearly independently to the first. Differentiating the cardinality with respect to k:

$$\frac{d}{dk}(k+1)(d+1-k) = d - 2k \leq 0 \qquad \text{for} \qquad \lceil d/2 \rceil \leq k \leq d,$$

we see that the cardinality is an even symmetric function around $k = \lceil d/2 \rceil$, where it achieves its maximum. This inspires the definition of the coherence-like quantity $\mu_{\mathbf{A}}$, whose value is monotonically related to the cardinality equation above, and satisfies $\lceil d/2 \rceil \leq \mu_{\mathbf{A}} \leq d$. Further, note that for two anchor points, $\mu_{\mathbf{A}} = \max(h(\neg\mathbf{a}_1, \mathbf{a}_2), h(\mathbf{a}_1, \mathbf{a}_2)) = \max(k, d-k)$. For m row, $\mu_{\mathbf{A}}$ is an upper bound on any pairwise similarity between all rows and their negations. Therefore, we can apply the

bound above to $\lfloor m/2 \rfloor$ pairs and have at most $(d+1)$ more values from the remaining dimension if m is odd. $\square$

The affine representation of Prop. 1 implies a strong similarity of $\mu_{\mathbf{A}}$ to the coherence of the dictionary rows:

$$2\mu_{\mathbf{A}} = d + \max_{i,j} \left| \bar{\mathbf{a}}_i^\top \bar{\mathbf{a}}_j \right| .$$

The mutual coherence of dictionary columns is a central quantity in the theory of compressed sensing [211]. Further, $\mu_{\mathbf{A}}$ provides a theoretical motivation for the dictionary designs. Indeed, the binary wavelet dictionary reaches the lowest possible coherence of $d/2$ in power-of-two dimensions and leads to great performance on a variety of benchmarks (see Fig. 3.4). Intuitively, we want to reduce the cardinality of the search space enough to accelerate optimization, but not so much that it fails to be a useful inductive bias. Note that $d/2 \leq \mu_{\mathbf{A}} \leq d$ and the bound attains its maximum for $\mu_{\mathbf{A}} = d/2$. For example, having duplicate elements in the dictionary would imply $\mu_{\mathbf{A}} = d$, and lead to a much larger drop in the cardinality for the same m than for the binary wavelet dictionary.

We now prepare to apply the bound of Prop. 3 in conjunction with the seminal result of [199] to provide an improved regret bound for BODi. Recall that the regret at iteration t is defined by $r_t = f(\mathbf{x}^*) - f(\mathbf{x}_t)$, where $\mathbf{x}^*$ is an optimal point and $\mathbf{x}_t$ is the

point chosen in the t^{th} iteration. The cumulative regret is $R_T = \sum_{t=1}^{T} f(\mathbf{x}^*) - f(\mathbf{x}_t)$ and is a key quantity in the theoretical study of BO algorithms. Many BO methods are no-regret (i.e. $\lim_{T\to\infty} R_T/T = 0$), though the rate with which R_T approaches zero varies significantly.

[199] prove a regret bound that is sub-linear in T for GP-based optimization with the upper confidence bound (UCB) acquisition function $\arg\max_{\mathbf{x}} \mu_{t-1}(\mathbf{x}) + \sqrt{\beta_t}\sigma_{t-1}(\mathbf{x})$, where μ_t (resp. σ_t^2) are the predictive mean (resp. variance) of the GP after t iterations. The bound mainly depends on two quantities: (1) The information gain after T iterations $\gamma_T = \log|\mathbf{I} + \sigma^{-2}\mathbf{K}_T|$, where $\mathbf{K}_T$ is the kernel matrix evaluated on the inputs $\{\mathbf{x}_t\}_{t=1}^{T}$ that were chosen in the first T iterations and σ is the standard deviation of the observations noise. (2) The cardinality of the search space $|\mathcal{S}|$, which we bound in Prop. 3 for BODi. Notably, γ_T depends on the kernel function and for the Matérn-ν kernel in our experiments, $\gamma_T = \mathcal{O}(T^{d(d+1)/(2\nu+d(d+1))}\log T)$. In the following, we use $\mathcal{O}^*$ to refer to $\mathcal{O}$ with log factors suppressed.

Theorem 4. *Let* $\mathbf{A}$ *have* m *rows,* $\delta \in (0,1)$*, and* $\beta_t = 2\log(|\mathcal{S}_{\mathbf{A}}|t^2\pi^2/6\delta)$*. Then the cumulative regret associated with running UCB for a sample* f *of a zero-mean GP with kernel function* $k_{BODi}(\mathbf{x},\mathbf{x}') = k_{base}(\boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\phi}_{\mathbf{A}}(\mathbf{x}'))$*, is upper-bounded by* $\mathcal{O}^*(\sqrt{T\gamma_T m})$ *with probability* $1-\delta$*, where* γ_T *is the maximum information gain of* k_{base}*.*

Theorem 4 exhibits a reduced dimensionality-dependent regret scaling of $\mathcal{O}^*(\sqrt{m})$,

compared to $\mathcal{O}^*(\sqrt{d})$ for non-embedded binary inputs, as long as m is not too large. We stress that this is due to the compressed cardinality of the search space, not the reduced dimensionality of the embedding. However, it is also important to note that not just the cardinality matters for optimization performance, since there are two main objectives that are usually at odds: (1) finding a model that is expressive enough and (2) reducing the complexity of fitting and optimizing this model. Simply reducing the cardinality of the search space will make it easier to fit the model, but potentially less likely to accurately model the underlying black-box objective function.

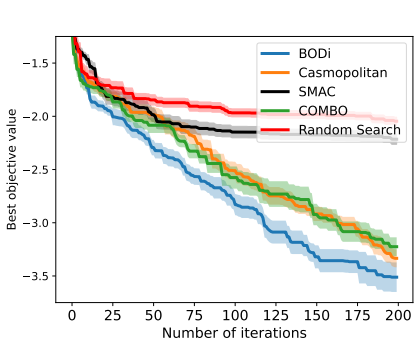
Starting with a large dictionary allows the model to choose from a large number of elements and adaptively prune redundant dimensions via ARD. In fact, our experiments confirm that larger embedding dimensions tend to improve performance and that ARD effectively prunes away the majority of embedding dimensions (see Sec. 3.6.2). The fact that the embedding values are ordinal, rather than binary, likely aids the inference of appropriate length scales. This results in the search space cardinality reduction shown by Prop. 3.

3.6 Experiments

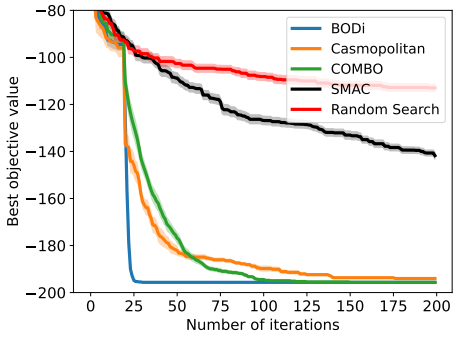
We evaluate BODi on wide range of challenging optimization problems for combinatorial and hybrid search spaces. We compare against several competitive baselines including CASMOPOLITAN, COMBO, CoCaBO, SMAC, and random search.

Experimental Setup. We use expected improvement as the acquisition function for all experiments. However, note that our approach is agnostic to this choice and any other acquisition function can be employed, which makes it easy to extend BODi to, e.g., multi-objective, multi-fidelity, and constrained settings. We employ a Matérn-5/2 kernel with ARD for both discrete and continuous variables. When considering combinatorial search spaces, we optimize the acquisition function using hill-climbing local search, similarly to the approach used by CASMOPOLITAN [215]. We follow Alg. 2 and $m = 128$ and the diverse random approach to construct dictionaries for all experiments. The choice $m = 128$ is investigated in an ablation study in Fig. 3.3c. Our code is built on top of the popular GPyTorch [87] and BoTorch [15] libraries. We use the open-source implementations for all the baselines: CASMOPOLITAN [1], COMBO [2], CoCaBO [3], and SMAC [4].

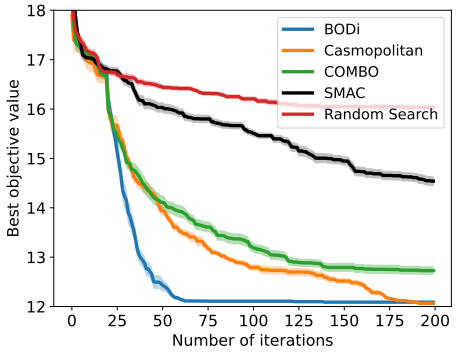
[1] https://github.com/xingchenwan/Casmopolitan
[2] https://github.com/QUVA-Lab/COMBO
[3] https://github.com/rubinxin/CoCaBO_code
[4] https://github.com/automl/SMAC3



(a) LABS (50 binary parameters)



(b) MaxSAT (60 binary parameters)



(c) Pest Control (25 categorical parameters with 5 possible values)

Figure 3.2: We compare BODi to CASMOPOLITAN, COMBO, SMAC, and random search on three high-dimensional combinatorial test problems. We find that BODi consistently performs the best followed by CASMOPOLITAN and COMBO.

3.6.1 Combinatorial Test Problems

LABS. The goal in the Low Auto-correlation Binary Sequences (LABS) problem is to find a binary sequence $\{1, -1\}$ of length n that maximizes the *Merit factor (MF)*:

$$\max_{\mathbf{x}\in\{1,-1\}^n} \mathrm{MF}(\mathbf{x}) = \frac{n^2}{E(\mathbf{x})}\,,$$
$$E(\mathbf{x}) = \sum_{k=1}^{n-1}\left(\sum_{i=1}^{n-k} x_i x_{i+k}\right)^2$$

This problem has diverse applications in multiple fields [30, 172], including communications where it is used in high-precision interplanetary radar measurements of space-time curvature [193]. We evaluate all methods on the 50-dimensional version of this problem. Fig. 3.2a plots the negative MF and shows that BODi finds significantly better solutions than the baselines. While COMBO and CASMOPOLITAN perform worse than BODi, they find better solutions than SMAC. Random search performs quite poorly, indicating the importance of employing model-guided search techniques for challenging problems (the combinatorial space for LABS has $2^{50} \approx 1.2 \times 10^{15}$ configurations). Note that Packebusch and Mertens [172] published the optimizer $\mathbf{x}_{\text{opt}}$ of the 50-dimensional LABS problem with $\mathrm{MF}(\mathbf{x}_{\text{opt}}) = 8.170$, which was computed with a branch-and-bound algorithm at exponential computational cost. We emphasize that our results here are not meant to advocate for the solution of this particular LABS problem using BO, but to serve as a comparison of the BO algorithms, which

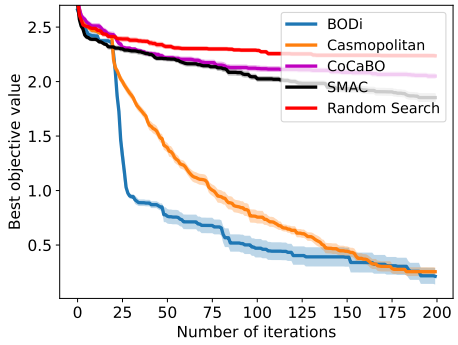
are designed to be sample efficient, on a challenging combinatorial optimization task.

Weighted maximum satisfiability. The goal of this problem is to find a 60-dimensional binary vector that maximizes the combined weights of satisfied clauses. We use the benchmark problem `frb-frb10-6-4.wcnf`[5] of the Maximum Satisfiability Competition 2018[6], similar to Oh et al. [169] and Wan et al. [215]. Satisfiability problems are ubiquitous and frequently arise in many fundamental areas of computer science [32]. Fig. 3.2b shows that BODi is quickly able to find a close-to-optimal solutions even though this combinatorial search space has as many as $2^{60} \approx 1.2 \times 10^{18}$ possible configurations. The strong performance of BODi on this problem is due to the superior model performance of the GP trained on the HED, see Sec. 3.6.2.

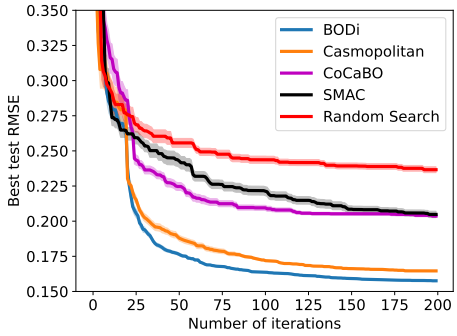
Pest control. This problem concerns the control of pest spread in a chain of 25 stations where a categorical choice of 5 possible options can be made at each station to use a pesticide differing in terms of their cost and effectiveness. This problem is challenging due to the $5^{25} \approx 3.0 \times 10^{17}$ total number of configurations. From Fig. 3.2c we observe that BODi quickly converges to a solution with objective value around ≈ 12 and substantially outperforms the other baselines on this problem.

[5] https://maxsat-evaluations.github.io/2018/index.html

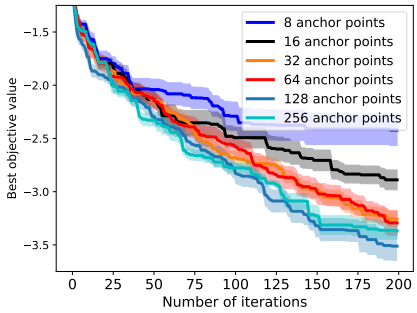
[6] http://sat2018.azurewebsites.net/competitions/



(a) Mixed Ackley (50 binary parameters, 3 continuous parameters)



(b) SVM (50 binary parameters, 3 continuous parameters)



(c) LABS ablation (50 binary parameters)

Figure 3.3: (Left, Middle) We compare BODi to CASMOPOLITAN, CoCaBO, SMAC, and random search and two high-dimensional problems with both discrete and continuous parameters. BODi converges faster than CASMOPOLITAN on the Ackley problem and performs better on the SVM problem. (Right) We study the sensitivity of BODi to the size of the dictionary (m) and observe consistent performance as long as we do not use dictionaries with too few elements.

3.6.2 Model Performance

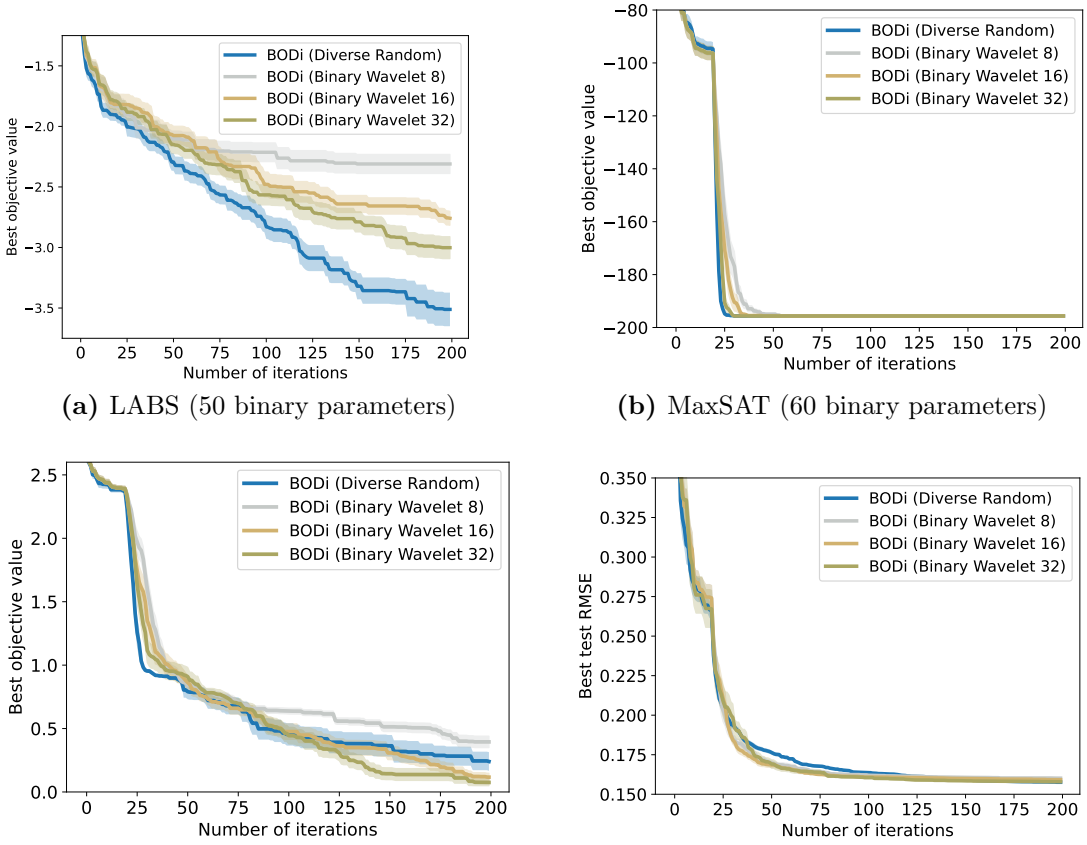
To validate that a GP using the HED provides accurate and well-calibrated estimates relative to categorical overlap kernels (used in CASMOPOLITAN, [215]), and the diffusion kernel (used in COMBO, [169]), we examine the predictive performance

of these different kernels on a 60-dimensional MaxSAT problem. We generate 50 training points and 50 test points and compare the test predictions of the dictionary-based kernel with the GP relative to the overlap kernel and diffusion kernel. The mean predictions on the test set with associated 95% predictive intervals are shown in Fig. 3.5.

The HED with diverse random dictionary elements gives rise to an accurate model of the unknown black-box function, while overlap and diffusion kernels fail to produce accurate test predictions. In addition, we also observe that HED with a Gaussian random dictionary – computed via the affine representation of Prop. 1 – performs poorly. Finally, even though we use dictionaries with $m = 128$ elements in Fig. 3.5, it turns out that only 4 of them have a lengthscale below 10 in the fitted GP model. This shows that ARD is able to effectively prune away the majority of dictionary elements and only use a small number of them, which leads to a tighter regret bound according to Thm. 4.

3.6.3 Hybrid Test Problems

Mixed Ackley. We consider a hybrid version of the standard Ackley problem from [215] with 50 binary and 3 continuous variables. We see that BODi makes quick progress and approaches the global optimal value of 0 (Fig. 3.3a). Except for CASMOPOLITAN, all other baselines perform poorly on this problem. Notably, the sub-sampled binary wavelet dictionary also performs particularly well on this prob-



(a) LABS (50 binary parameters)

(b) MaxSAT (60 binary parameters)

(c) Ackley (50 binary parameters, 3 continuous parameters)

(d) SVM (50 binary parameters, 3 continuous parameters)

Figure 3.4: Results comparing the two dictionary construction choices for BODi (i.e., diverse random and binary wavelet). Overall, we find that binary wavelet design performs reasonably well but diverse random is a more robust choice considering all the benchmarks. Moreover, the diverse random choice can also be employed for categorical parameters unlike the binary wavelet construction which is limited to binary parameters.

lem, see Fig. 3.4c.

Feature selection for SVM training. In this problem, we consider joint feature selection and hyperparameter optimization for training a support vector machine (SVM) model on the UCI slice dataset [75]. We optimize over the inclusion/exclusion

of 50 features, and additionally tune the C, ϵ, and γ hyperparameters of the SVM. The goal is to find the optimal subset of features and values of the continuous hyperparameters in order to minimize the RMSE on a held-out test set. Fig. 3.3b shows that BODi performs slightly better than CASMOPOLITAN on this real-world problem.

3.6.4 Ablation Study

We perform an ablation study on the sensitivity of BODi to the number of elements of the dictionary (dictionary size). We consider the 50-dimensional LABS problem. The results in Fig. 3.3c show that dictionaries with $m = 128$ or $m = 256$ elements perform the best (albeit differences in performance are relatively small, at least for larger m). We observe that using a small dictionary (with $m = 16$ or $m = 32$ elements) results in inferior performance. On the other hand, using a large number of elements increases the runtime of our method, which is why we opted for the choice of $m = 128$ for all experiments.

3.7 Summary

We introduced a novel dictionary kernel for GP models, which is suitable for high-dimensional combinatorial search spaces (and can be straightforwardly extended to hybrid search spaces). While we focused on using our dictionary-based modeling approach for BO, the implications of our contributions go far beyond BO alone and

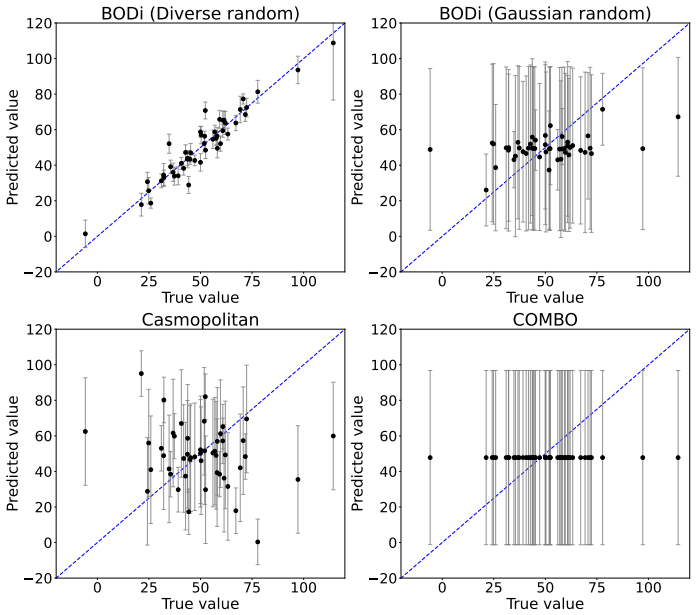


Figure 3.5: Mean predictions and associated 95% predictive intervals on then MaxSAT problem for BODi with diverse random dictionary (top left), BODi with a Gaussian random dictionary via the affine representation of Eq. (3.1) (top right), `Casmopolitan` (bottom left), and `COMBO` (bottom right). We use 50 training points and predict on 50 test points. BODi with the diverse random dictionary performs much better than with the Gaussian random embedding, validating our theoretical results in Sec. 3.5. Our kernel also outperforms the isotropic kernel used by CASMOPOLITAN and the diffusion kernel used by COMBO.

are relevant for kernel-based methods more generally. In the context of BO, our dictionary kernel is agnostic to the choice of acquisition function and can be easily applied to settings such as multi-objective and multi-fidelity optimization, and can also be combined with ideas such as trust region optimization. BODi showed strong performance on a diverse set of problems and outperformed several strong baselines such as CASMOPOLITAN and COMBO.

Our work has a few limitations and raises a number of interesting questions that

warrant further exploration. While BODi is agnostic to the choice of acquisition function, we only evaluated its performance on single-objective problems. In addition, rather than randomly generating a diverse set of dictionary elements, we may be able to further improve the dictionary-based GP model by optimizing the dictionary as part of the model fitting procedure. This may be particularly useful in cases where we have access to historical data that can help us discover suitable dictionaries. Alternatively, there may be ways of generating the dictionaries in a way that is more aligned with the goal of BO, which is not to fit a globally accurate model but rather identify the location of the global optimum. Finally, BODi may also benefit from recently proposed methods for efficient acquisition function optimization in hybrid search spaces [54].

CHAPTER FOUR

DIFFUSION KERNEL BASED SURROGATE MODEL FOR HYBRID SPACES

In this chapter, we address the problem of optimizing hybrid structures (mixture of discrete and continuous input variables) via expensive black-box function evaluations. Unlike the previous chapter, where we focused on high-dimensional input spaces, here we consider low/moderate dimensional input spaces. This problem arises in many real-world applications. For example, in materials design optimization via lab experiments, discrete and continuous variables correspond to the presence/absence of primitive elements and their relative concentrations respectively. The key challenge is to accurately model the complex interactions between discrete and continuous variables. We propose a novel approach referred as **Hy***brid* **B***ayesian* **O***ptimization (HyBO)* by utilizing diffusion kernels, which are naturally defined over continuous and discrete variables. We develop a principled approach for constructing diffusion kernels over hybrid spaces by utilizing the additive kernel formulation, which allows additive interactions of all orders in a tractable manner. We theoretically analyze the modeling strength of additive hybrid kernels and prove that it has the *universal approximation* property. Our experiments on synthetic and six diverse real-world benchmarks show that HyBO significantly outperforms the state-of-the-art methods.

4.1 Problem Setup

Let $\mathcal{X}$ be a hybrid space to be optimized over, where each element $x \in \mathcal{X}$ is a hybrid structure. Without loss of generality, let each hybrid structure $x = (x_d \in \mathcal{X}_d, x_c \in \mathcal{X}_c) \in \mathcal{X}$ be represented using m discrete variables and n continuous variables, where x_d and x_c stands for the discrete and continuous sub-space of $\mathcal{X}$. Let each discrete variable v_d from x_d take candidate values from a set $C(v_d)$ and each continuous variable v_c from x_c take values from a compact subset of $\mathbb{R}$. We assume an unknown, expensive real-valued objective function $f : \mathcal{X} \mapsto \mathbb{R}$, which can evaluate each hybrid structure x (also called an experiment) and produces an output $y = f(x)$. For example, in high-entropy alloys optimization application, x_d corresponds to the presence/absence of metals and x_c corresponds to their relative concentrations, and $f(x)$ corresponds to running a physical lab experiment using additive manufacturing techniques. The main goal is to find a hybrid structure $x \in \mathcal{X}$ that approximately optimizes f by conducting a limited number of evaluations and observing their outcomes.

Algorithm 5 HyBO Approach

Input: $\mathcal{X}$ = Hybrid input space, $\mathcal{K}(x, x')$ = Kernel over hybrid structures, $\mathcal{AF}(\mathcal{M}, x)$ = Acquisition function parametrized by model $\mathcal{M}$ and input x, $\mathcal{F}(x)$ = expensive objective function

Output: $\hat{x}_{best}$, the best structure

Initialize $\mathcal{D}_0 \leftarrow$ initial training data; and $t \leftarrow 0$

repeat

  Learn statistical model: $\mathcal{M}_t \leftarrow$ GP-Learn($\mathcal{D}_t$, $\mathcal{K}$)

  Compute the next structure to evaluate:

  $x_{t+1} \leftarrow \arg max_{x \in \mathcal{X}} \mathcal{AF}(\mathcal{M}_t, x)$

  $x_c \leftarrow$ Optimize continuous subspace conditioned on assignment to discrete variables x_d

  $x_d \leftarrow$ Optimize discrete subspace conditioned on assignment to continuous variables x_c

  Evaluate objective function $\mathcal{F}(x)$ at x_{t+1} to get y_{t+1}

  Aggregate the data: $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$

  $t \leftarrow t + 1$

until convergence or maximum iterations

$\hat{x}_{best} \leftarrow \arg max_{x_t \in \mathcal{D}} y_t$

return the best uncovered hybrid structure $\hat{x}_{best}$

4.2 Diffusion Kernels over Hybrid Structures

We first provide the details of key mathematical and computational tools that are needed to construct hybrid diffusion kernels. Next, we describe the algorithm to automatically construct additive diffusion kernels over hybrid structures. Finally, we present theoretical analysis to show that hybrid diffusion kernels satisfy universal

approximation property.

4.2.1 Key Mathematical and Computational Tools

Diffusion kernels [136, 137] are inspired from the diffusion processes occurring in physical systems like heat and gases. The mathematical formulation of these processes naturally lends to kernels over both continuous and discrete spaces(e.g., sequences, trees, and graphs).

Diffusion kernel over continuous spaces. The popular radial basis function (RBF) kernel (also known as Gaussian kernel) [136] is defined as follows:

$$k(x, x') = \frac{1}{2\pi\sigma^2} e^{-\|x-x'\|^2/2\sigma^2} \tag{4.1}$$

where σ is the length scale hyper-parameter. This is the solution of the below continuous diffusion (heat) equation:

$$\frac{\partial}{\partial t} k_{x_0}(x, t) = \Delta k_{x_0}(x, t) \tag{4.2}$$

where $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \cdots \frac{\partial^2}{\partial x_D^2}$ is the second-order differential operator known as the *Laplacian operator*, and $k_{x_0}(x, t) = k(x, x')$ with $x' = x_0$ and $t = \sigma^2/2$.

4.2.2 *Diffusion Kernel over Discrete Spaces*

The idea of diffusion kernels for continuous spaces is extended to discrete structures (e.g., sequences, graphs) [135] by utilizing the spectral properties of a graph representation of the discrete space. A discrete analogue of the Equation 4.2 can be constructed by employing the matrix exponential of a graph and the *graph Laplacian operator* L as given below:

$$\frac{\partial}{\partial \beta} e^{\beta L} = L e^{\beta L} \tag{4.3}$$

where L is the graph Laplacian of a suitable graph representation of the discrete input space and β is a hyper-parameter of the resulting diffusion kernel similar to the length scale parameter σ of the RBF kernel. The solution of Equation 4.3 defines a positive-definite kernel for discrete spaces known as the discrete diffusion kernel.

According to Equation 4.3, one important ingredient required for defining diffusion kernels on discrete spaces is a suitable graph representation for discrete spaces. One such representation was proposed in a recent work [168]. In this case, the entire discrete space is represented by a combinatorial graph G. Each node in the vertex set V of the graph corresponds to one candidate assignment of all the discrete variables. Two nodes are connected by an edge if the Hamming distance between the corresponding assignments for all discrete variables is exactly one. The diffusion kernel

over this representation is defined as follows:

$$k(V, V) = \exp(-\beta L(G)) \tag{4.4}$$

$$k(V, V) = \Phi \exp(-\beta \Pi) \Phi^T \tag{4.5}$$

where $\Phi = [\phi_1, \cdots, \phi_{|V|}]$ is the eigenvector matrix and $\Pi = [\pi_1, \cdots, \pi_{|V|}]$ is the eigenvalue matrix, where ϕ_i's and π_i's are the eigenvectors and eigenvalues of the graph Laplacian $L(G)$ respectively. Although this graph representation contains an exponential number of nodes, [168] computes the graph Laplacian $L(G)$ by decomposing it over the Cartesian product ($\diamond$) of m (number of discrete variables) sub-graphs $(G_1, G_2 \cdots, G_m)$ with each sub-graph G_i representing one variable individually. This algorithmic approach has time-complexity $O(\sum_{i=1}^{m}(C(v_i))^3)$, where $C(v_i)$ is the number of candidate values (arity) for the ith discrete variable. However, this method is computationally expensive, especially, for problems with large-sized arity.

To avoid this computational challenge, we leverage prior observation in [135] which provides a *closed-form* of the discrete diffusion kernel by exploiting the structure of the above combinatorial graph representation. We explain this observation for binary variables $\{0, 1\}$. From its definition in Equation 4.4, the discrete diffusion kernel over single-dimensional input will be:

$$k(x_d, x'_d) = \begin{cases} (1 - e^{-2\beta}) & \text{if } x_d \neq x'_d \\ (1 + e^{-2\beta}) & \text{if } x_d = x'_d \end{cases} \tag{4.6}$$

Since the kernel over $m > 1$ dimensions is defined using the Kronecker product over m dimensions, the above expression is easily extended to multiple dimensions setting giving:

$$k(x_d, x'_d) = \prod_{i=1}^{m} \frac{(1 - e^{-2\beta_i})}{(1 + e^{-2\beta_i})}^{\delta(x_d^i, x_d'^i)} \tag{4.7}$$

where $\delta(x_d^i, x_d'^i) = 0$ if x_d^i is equal to $x_d'^i$ and 1 otherwise. The subscript d denotes that the variables are discrete and the superscript refers to the ith dimension of the discrete subspace. For general (discrete spaces with arbitray categories), we follow the same observation [135] and use the following constant-time expression of the discrete diffusion kernel in our method:

$$k(x_d, x'_d) = \prod_{i=1}^{m} \left(\frac{1 - e^{-C(v_i)\beta_i}}{1 + (C(v_i) - 1)e^{-C(v_i)\beta_i}} \right)^{\delta(x_d^i, x_d'^i)} \tag{4.8}$$

4.2.3 Diffusion Kernels over Hybrid Spaces

Unifying view of diffusion kernels. Our choice of diffusion kernels is motivated by the fact that they can be naturally defined for both discrete and continuous spaces. In fact, there is a nice transition of the diffusion kernel from discrete to continuous space achieved by continuous space limit operation. More generally, both discrete and continuous diffusion kernel can be seen as continuous limit operation on two parameters of random walks: *time* and *space*. For illustration, consider a random walk on an evenly spaced grid where mean time of jump is t and mean gap between two points is s. If $t \to 0$, the resulting continuous time and discrete space random walk generates the diffusion kernel on discrete spaces. Additionally, in the limit of the grid spacing s going to zero, the kernel will approach the continuous diffusion kernel.

Algorithm to construct hybrid diffusion kernels. We exploit the general formulation of additive Gaussian process kernels [76] to define an *additive hybrid diffusion* kernel over hybrid spaces. The key idea is to assign a base kernel *for each input dimension* $i \in \{1, 2, \cdots, m+n\}$, where m and n stand for the number of discrete and continuous variables in hybrid space $\mathcal{X}$; and construct an overall kernel by summing all possible orders of interactions (upto $m + n$) between these base kernels. In our case, the RBF kernel and the discrete diffusion kernel acts as the base kernel for continuous and discrete input dimensions respectively. The p^{th} order of interaction

(called p^{th} *additive kernel*) is defined as given below:

$$\mathcal{K}_p = \theta_p^2 \sum_{1 \leq i_1 < i_2 < \cdots, i_p \leq m+n} \left(\prod_{d=1}^{p} k_{i_d}(x_{i_d}, x'_{i_d}) \right)$$

where θ_p is a hyper-parameter associated with each additive kernel and k_{i_d} is the base kernel for the input dimension i_d. In words, the pth additive kernel is a sum of $\binom{m+n}{p}$ terms, where each term is a product of p distinct base kernels. Estimation of θ_p hyper-parameter from data allows automatic identification of important orders of interaction for a given application. The overall *additive hybrid diffusion kernel* $\mathcal{K}_{HYB}(x, x')$ over hybrid spaces is defined as the sum of all orders of interactions as given below:

$$\mathcal{K}_{HYB} = \sum_{p=1}^{m+n} \mathcal{K}_p \tag{4.9}$$

$$\mathcal{K}_{HYB} = \sum_{p=1}^{m+n} (\theta_p^2 \sum_{i_1, \cdots, i_p} \prod_{d=1}^{p} k_{i_d}(x_{i_d}, x'_{i_d})) \tag{4.10}$$

It should be noted that the RHS in Equation 4.10 requires computing a sum over exponential number of terms. However, this sum can be computed in polynomial time using Newton-Girard formula for elementary symmetric polynomials [76]. It is

an efficient formula to compute the p^{th} additive kernel recursively as given below:

$$\mathcal{K}_p = \theta_p^2 \cdot \left(\frac{1}{p} \sum_{j=1}^{p} (-1)^{(j-1)} \mathcal{K}_{p-j} S_j \right) \tag{4.11}$$

where $S_j = \sum_{i=1}^{m+n} k_i^j$ is the jth power sum of all base kernels k_j and the base case for the recursion can be taken as 1 (i.e., $\mathcal{K}_0 = 1$). This recursive algorithm for computing additive hybrid diffusion kernel has the time complexity of $\mathcal{O}((n+m)^2)$.

Data-driven specialization of kernel for a given application. In real-world applications, the importance of different orders of interaction can vary for optimizing the overall performance of BO approach (i.e., minimizing the number of expensive function evaluations to uncover high-quality hybrid structures). For example, in some applications, we may not require all orders of interactions and only few will suffice. The θ_p hyper-parameters in the additive hybrid diffusion kernel formulation allows us to identify the strength/contribution of the pth order of interaction for a given application in a *data-driven* manner. We can compute these parameters (along with the hyper-parameters for each base kernel) by maximizing the marginal log-likelihood, but we consider a fully Bayesian treatment by defining a prior distribution for each of them. This is important to account for the uncertainty of the hyper-parameters across BO iterations. The acquisition function $\mathcal{AF}(x)$ is computed by marginalizing

the hyper-parameters as given below:

$$\mathcal{AF}(x;\mathcal{D}) = \int \mathcal{AF}(x;D,\Theta)p(\Theta|D)d\Theta \tag{4.12}$$

where Θ is a variable representing all the hyperparameters (σ for continuous diffusion kernel, β for discrete diffusion kernel, and θ for strengths of different orders of interaction in hybrid diffusion kernel) and $\mathcal{D}$ represents the aggregate dataset containing the hybrid structure and function evaluation pairs. The posterior distribution over the hyper-parameters is computed using slice sampling [164].

4.2.4 Theoretical Analysis

Intuitively, a natural question to ask about the modeling power of a kernel is whether (given enough data) it can approximate (with respect to a suitable metric) any black-box function defined over hybrid spaces. This is a minimum requirement that should guide our choice of kernel in the given problem setting. This question has been studied widely in the form of a key property called *universality* of a kernel [201, 154, 200, 151]. In this section, we prove the universality of the *additive hybrid diffusion kernel* by combining the existing result on the universality of RBF (Gaussian) kernel with a novel result proving the universality of discrete diffusion kernels.

Proposition 5. *[201, 154] Let $\mathcal{X}_c$ be a compact and non-empty subset of $\mathbb{R}^n$. The RBF kernel in Equation 4.1 is a universal kernel on $\mathcal{X}_c$.*

A kernel k defined on an input space $\mathcal{X}_c$ has a unique correspondence with an associated Reproducing Kernel Hilbert Space (RKHS) of functions $\mathcal{H}_k$ defined on $\mathcal{X}_c$ [202]. For compact metric input spaces $\mathcal{X}_c$, a kernel k is called universal if the RKHS $\mathcal{H}_k$ defined by it is dense in the space of continuous functions $C(\mathcal{X}_c)$. [201] proved the universality of the RBF (Gaussian) kernel with respect to the uniform norm. [154] established universality for a larger class of translation invariant kernels. [200] discussed various notions of universality and connected to the concept of *characteristic kernels*.

Proposition 6. *Let $\mathcal{X}_d$ be the discrete space $\{0,1\}^m$ and a psuedo-boolean function on $\mathcal{X}_d$ is defined as $f : \mathcal{X}_d \mapsto \mathbb{R}$. The discrete diffusion kernel is a universal kernel on $\mathcal{X}_d$.*

Proof. A Reproducing Kernel Hilbert Space $\mathcal{H}_k$ associated with a kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is defined as:

$$\mathcal{H}_k = cl(span\{k(x,\cdot), \forall x \in \mathcal{X}\}) \tag{4.13}$$

where cl represents the closure and $k(x,\cdot)$ is called as the feature map of x [202].

In our setting, a kernel k defined on discrete input space $\mathcal{X}_d$ is universal if and only if any pseudo-Boolean function f can be written as a linear combination of functions

$(k(x_{i_d}, \cdot), \forall x_{i_d} \in \mathcal{X}_d)$ in the RKHS $\mathcal{H}_k$ [151, 93], i.e.

$$\forall f : \mathcal{X}_d \mapsto \mathbb{R}; \quad \exists a_i \in \mathbb{R}; f = \sum_i a_i k(x_{i_d}, \cdot); \tag{4.14}$$

We prove that this is true by computing the explicit form of functions $(k(x_{i_d}, \cdot), \forall x_{i_d} \in \mathcal{X}_d)$ existing in the RKHS $\mathcal{H}_k$ of the discrete diffusion kernel. To see this, we exploit the structure of the combinatorial graph representation of the discrete space discussed in Section 4.2.1. The discrete diffusion kernel is defined in terms of the eigenvectors ϕ_i and eigenvalues π_i of the graph Laplacian $L(G)$ as follows:

$$k(x_d, x'_d) = \sum_{i=1}^{2^n} \phi_i[x_d] \exp(-\beta\pi_i)\phi_i[x'_d] \tag{4.15}$$

Since the combinatorial graph G is generated by the Cartesian product over subgraphs G_i (one for each discrete variable), the eigenvectors term $\phi_i[x_d]$ can be calculated via an explicit formula, i.e., $\phi_i[x_d] = -1^{w^T x_d}$, where w is a binary vector of size n [48] (number of discrete variables).

$$k(x_d, x'_d) = \sum_{i=1}^{2^n} -1^{w^T x_d} \exp(-\beta\pi_i) - 1^{w^T x'_d} \tag{4.16}$$

$$< k(x_d, \cdot), k(x'_d, \cdot) >= \sum_{i=1}^{2^n} -1^{w^T x_d} \exp(-\beta\pi_i) - 1^{w^T x'_d} \tag{4.17}$$

where the inner product in LHS follows from the reproducing property [202] of a

kernel k. Therefore, the functions $k(x_d, \cdot)$ in the RKHS $\mathcal{H}_k$ of the discrete diffusion kernel are of the form $\{-1^{{w_j}^T x_d}; w_j \in [0, 2^n - 1]\}$, which is the well-known *Walsh Basis* [214] for pseudo-Boolean functions. Therefore, any pseudo-Boolean function f can be represented by a linear combination of functions in $\mathcal{H}_k$ since they form a basis.

Theorem 7. *Let $\mathcal{X}_c$ be a compact and non-empty subset of $\mathbb{R}^n$ and κ_c be RBF kernel on $\mathcal{X}_c$. Let $\mathcal{X}_d$ be the discrete space $\{0,1\}^m$ and κ_d be discrete diffusion kernel on $\mathcal{X}_d$. The additive hybrid diffusion kernel defined in Eqn 4.10, instantiated with k_c and k_d for continuous and discrete spaces respectively, is a universal kernel for the hybrid space $\mathcal{X}_c \times \mathcal{X}_d$.*

According to Equation 4.9, any pth order of interaction term in the additive hybrid diffusion kernel is defined as $\left(\prod_{d=1}^{p} k_{i_d}(x_{i_d}, x'_{i_d})\right)$. Therefore, if each k_{i_d} is universal over its corresponding dimension X_{i_d} (which is true from Propositions 1 and 2), we need to show that the product $\left(\prod_{d=1}^{p} k_{i_d}(x_{i_d}, x'_{i_d})\right)$ is universal over the union of dimensions $\mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \cdots \times \mathcal{X}_{i_p}$. This was proven by Lemma A.5 in [203]. We provide the lemma here for completeness.

Lemma 8. *From [203] Let $\mathcal{X} \subset \mathbb{R}^m$ be a compact and non-empty subset, $I, J \subset \{1, \ldots, m\}$ be non-empty, and k_I and k_J be universal kernels on $\mathcal{X}_I \times \mathcal{X}_J$, respectively. Then $k_I \otimes k_J$ defined by*

$$k_I \otimes k_J(x, x') := k_I(x_I, x'_I) \cdot k_J(x_J, x'_J)$$

for all $x, x' \in \mathcal{X}_I \times \mathcal{X}_J$ *is a universal kernel on* $\mathcal{X}_I \times \mathcal{X}_J$.

Since both continuous and discrete spaces are compact and Lemma 8 is true for arbitrary compact spaces, each order of interaction is universal with respect to its corresponding ambient dimension $\mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \cdots \times \mathcal{X}_{i_p}$. In particular, it is true for $m + n$th order of interaction which is defined over the entire hybrid space $\mathcal{X}_c \times \mathcal{X}_d$ which proves the theorem.

4.3 Experiments and Results

We first describe our experimental setup. Next, we discuss experimental results along different dimensions.

4.3.1 Benchmark Domains

Synthetic benchmark suite. `bbox-mixint` is a challenging mixed-integer blackbox optimization benchmark suite [212] that contains problems of varying difficulty. This benchmark suite is available via COCO platform[1]. We ran experiments with multiple problems from this benchmark, but for brevity, we present canonical results on four benchmarks (shown in Table 4.1) noting that all the results show similar trends.

Real world benchmarks. We employ six diverse real-world domains.

1) Pressure vessel design optimization. This mechanical design problem [127, 207] involves minimizing the total cost of a cylindrical pressure vessel. There

[1] `https://github.com/numbbo/coco`

Name	Name in the suite	Dimension
Function 1	f001_i01_d10	10 (8d, 2c)
Function 2	f001_i02_d10	10 (8d, 2c)
Function 3	f001_i01_d20	20 (16d, 4c)
Function 4	f001_i02_d20	20 (16d, 4c)

Table 4.1: Benchmark problems from bbox-mixint suite.

are two discrete (thickness of shell and head of pressure vessel) and two continuous (inner radius and length of cylindrical section) variables.

2) Welded beam design optimization. The goal in this material engineering domain [57, 182] is to design a welded beam while minimizing the overall cost of the fabrication. There are six variables: two discrete (type of welding configuration and bulk material of the beam) and four continuous (weld thickness, welded joint length, beam width and thickness).

3) Speed reducer design optimization. In this domain from NASA [39], the goal is to minimize the weight of a speed reducer defined over seven input variables: one discrete (number of teeth on pinion) and six continuous (face width, teeth module, lengths of shafts between bearings, and diameters of the shafts)

4) Optimizing control for robot pushing. This is a 14 dimensional control parameter tuning problem, where a robot is trying to push objects toward a goal location [217]. We consider a hybrid version of this problem by discretizing ten input variables corresponding to location of the robot and number of simulation steps. The remaining four parameters corresponding to rotation are kept as continuous.

5) Calibration of environmental model. The problem of calibration and uncertainty analysis of expensive environmental models is very important in scientific domains [33, 13]. There are four input variables (one discrete and three continuous).

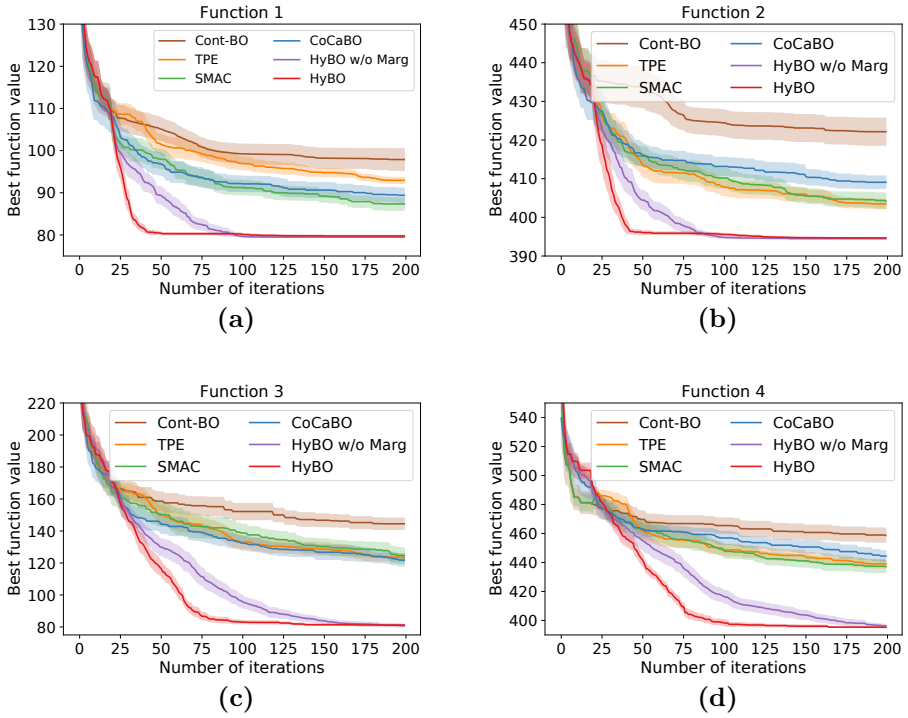


Figure 4.1: Results of HyBO and state-of-the-art baselines on bbob-mixint benchmark suite for functions shown in Table 4.1.

6) Hyper-parameter optimization. We consider hyper-parameter tuning of a neural network model on a diverse set of benchmarks [90]: five discrete (hidden layer size, activation type, batch size, type of learning rate, and whether to use early stopping or not) and three continuous (learning rate initialization, momentum parameter, and regularization coefficient) hyper-parameters.

4.3.2 Experimental Setup

Baseline methods. We compare HyBO with four strong baselines: 1) `CoCaBO`, a state-of-the-art method [184]; 2) `SMAC` [105]; 3) `TPE` [29]; 4) `HyBO w/o Marg` is a special case of HyBO, where we do not perform marginalization over the hyper-parameters of the hybrid diffusion kernel; and 5) `Cont-BO` treats discrete variables as continuous and performs standard BO over continuous spaces (both modeling and acquisition function optimization). We did not include `MiVaBO` [56] as there was no publicly available implementation [55] [2].

Configuration of algorithms and baselines. We configure `HyBO` as follows. We employ uniform prior for the length scale hyperparameter (σ) of the RBF kernel. Horse-shoe prior is used for β hyper-parameter of the discrete diffusion kernel (Equation 4.8) and hyper-parameters θ of the additive diffusion kernel (Equation 4.9). We employ expected improvement [155] as the acquisition function. For acquisition function optimization, we perform iterative search over continuous and discrete sub-spaces as shown in Algorithm 5. For optimizing discrete subspace, we run local search with 20 restarts. We normalize each continuous variable to be in the range $[-1, 1]$ and employed CMA-ES algorithm [3] for optimizing the continuous subspace. We found that the results obtained by CMA-ES were not sensitive to its hyper-parameters. Specifically, we fixed the population size to 50 and initial standard deviation to 0.1 in all

[2] Personal communication with the lead author.
[3] `https://github.com/CMA-ES/pycma`

our experiments. We employed the open-source python implementation of CoCaBO [4], SMAC [5], and TPE [6].

All the methods are initialized with same random hybrid structures. We replicated all experiments for 25 different random seeds and report the mean and two times the standard error in all our figures.

Evaluation metric. We use the best function value achieved after a given number of iterations (function evaluations) as a metric to evaluate all methods. The method that uncovers high-performing hybrid structures with less number of function evaluations is considered better.

4.3.3 Results and Discussion

Dataset	Cont-BO	TPE	SMAC	CoCaBO	HyBO
blood	76.09 (0.33)	76.71 (0.43)	76.66 (0.42)	76.98 (0.46)	**77.82 (0.46)**
kc1	85.19 (0.13)	85.64 (0.07)	85.45 (0.09)	85.42 (0.10)	85.47 (0.12)
vehicle	80.50 (1.12)	80.91 (1.05)	83.67 (1.01)	82.88 (1.22)	**86.10 (0.89)**
segment	87.25 (1.00)	87.79 (0.54)	89.99 (0.69)	89.64 (0.73)	**91.43 (0.28)**
cnae	95.37 (0.10)	95.69 (0.08)	95.61 (0.06)	95.68 (0.11)	95.64 (0.14)
jasmine	77.32 (0.22)	77.89 (0.07)	77.46 (0.19)	77.51 (0.20)	77.12 (0.17)

Table 4.2: Results on the task of hyper-parameter tuning of neural network models. Bold numbers signify statistical significance.

Results on mixed integer benchmark suite. Figure 4.1 shows the canonical results on four benchmarks from `bbox-mixint` listed in Table 4.1 noting that all re-

[4] `https://github.com/rubinxin/CoCaBO_code`
[5] `https://github.com/automl/SMAC3`
[6] `https://github.com/hyperopt/hyperopt`

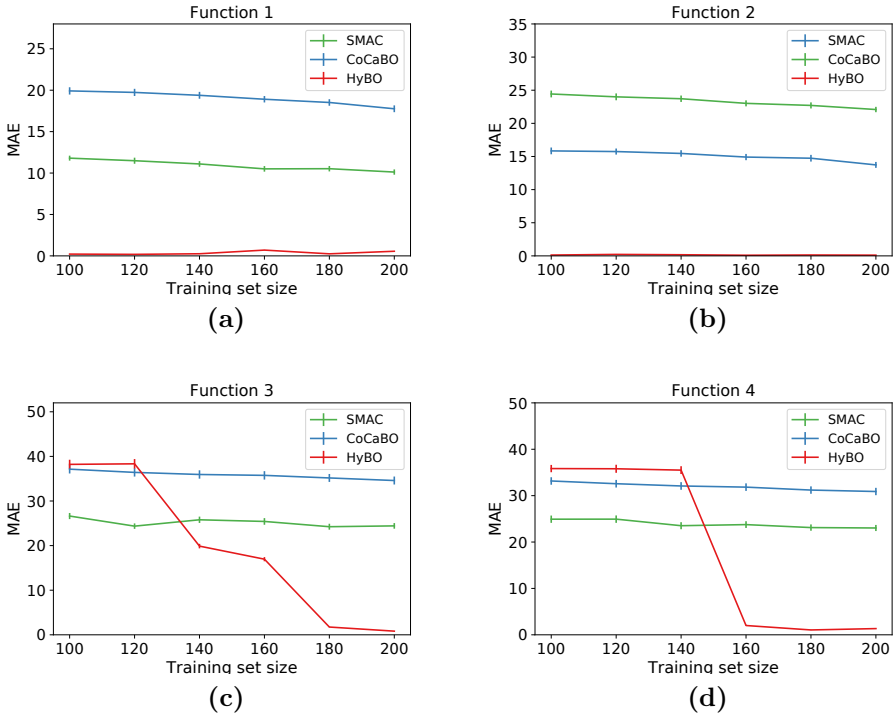


Figure 4.2: Results showing mean absolute test error with increasing size of training set on the bbob-mixint synthetic benchmarks.

sults show similar trends. HyBO and its variant HyBO-Round performs significantly better and converges much faster than all the other baselines. One key reason for this behavior is that hybrid diffusion kernel accounts for higher-order interactions between variables. Cont-BO performs the worst among all the methods. This shows that simply treating discrete variables as continuous is sub-optimal and emphasizes the importance of modeling the structure in discrete variables.

Ablation results for statistical models. To understand the reasons for the better performance of HyBO, we compare the performance of its surrogate model based on

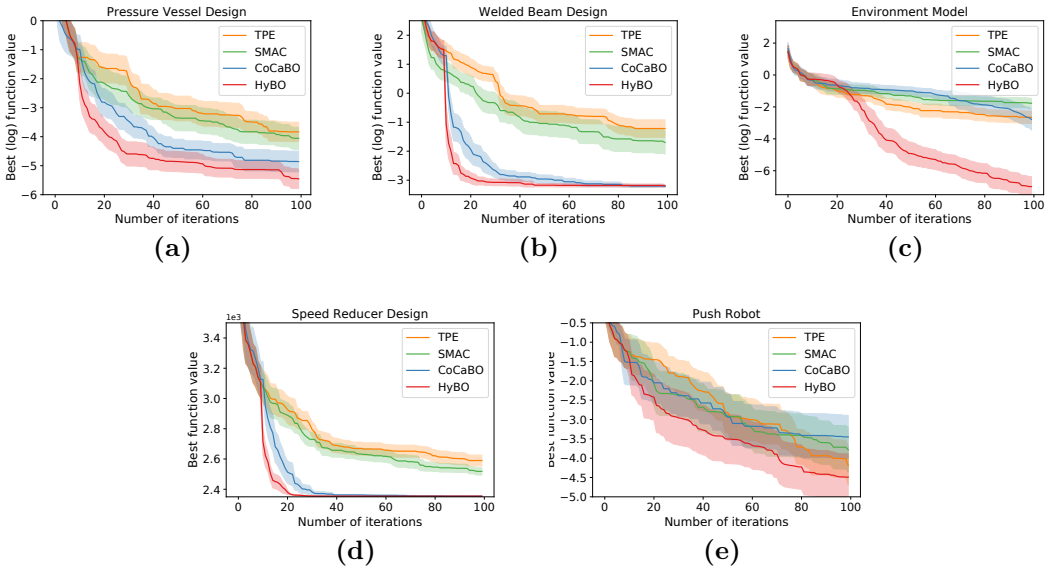


Figure 4.3: Results comparing the proposed HyBO approach with state-of-the-art baselines on multiple real world benchmarks.

Benchmark	**TPE**	**SMAC**	**CoCaBO**	**HyBO**
Synthetic Function 1	0.012	2.34	2.30	50
Synthetic Function 2	0.012	0.98	1.31	50
Synthetic Function 3	0.026	2.99	3.18	180
Synthetic Function 4	0.026	1.98	2.96	180
Pressure Vessel Design	0.003	0.34	0.85	20
Welded Beam Design	0.004	0.64	1.02	40
Speed Reducer Design	0.006	1.38	0.94	40
Push Robot	0.017	1.94	1.70	90
Environment model	0.005	0.31	0.50	40

Table 4.3: Computational cost in average wall-clock time (seconds) per BO iteration.

hybrid diffusion kernels with those of CoCaBO and SMAC. We perform the following experiment. We constructed testing dataset (pairs of hybrid structures and their function evaluations) of size 200 via uniform random sampling. We compute the mean absolute error (MAE) of the three surrogate models as a function of training

set size. The results are shown in Figure 4.2 which depicts the mean and two times standard error of the MAE on 25 random testing datasets. HyBO clearly has very low error compared to CoCaBO and SMAC on Function 1 and 2. Although HyBO has similar MAE to CoCaBO in the beginning on Function 3 and 4, it rapidly decreases as the training set size increases which is not the case for other two methods. This experiment provides strong empirical evidence for the fact that the proposed surrogate model in HyBO can model hybrid spaces more accurately when compared to CoCaBO and SMAC.

Ablation results for marginalization in HyBO. Bayesian treatment of hyper-parameters (marginalization) is one key component of our proposed HyBO method. However, to decouple the efficacy of additive diffusion kernel from the usage of marginalization, we performed experiments using HyBO without marginalization (HyBO w/o Marg in Figures). As evident from Figure 4.1, HyBO w/o Marg finds better solutions than all the baselines albeit with slower convergence which is improved by adding marginalization.

Results for real-world domains. Figure 4.3 shows comparison of HyBO approach with baseline methods on all real-world domains except hyper-parameter optimization. We make the following observations. 1) HyBO consistently performs better than all the baselines on all these benchmarks. 2) Even on benchmarks such as speed reducer design and welded beam design where HyBO finds a similar solution as Co-

CaBO, it does so with much faster convergence. 3) CoCaBO performs reasonably well on these benchmarks but its performance is worse than HyBO demonstrating that its sum kernel (along with Hamming kernel for discrete spaces) is less powerful than hybrid diffusion kernel of HyBO. 4). TPE has the worst performance on most benchmarks possibly a direct result of its drawback of not modeling the interactions between input dimensions. 5) SMAC performs poorly on all the benchmarks potentially due to poor uncertainty estimates from random forest surrogate model.

Table 4.2 shows the final accuracy (mean and standard error) obtained by all methods including HyBO on the task of tuning neural network models for six different datasets (BO curves are similar for all methods). HyBO produces comparable or better results than baseline methods.

Computational cost analysis. We compare the runtime of different algorithms including HyBO. All experiments were run on a AMD EPYC 7451 24-Core machine. Table **4.3** shows the average wall-clock time (in seconds) per BO iteration. We can see that HyBO is relatively expensive when compared to baseline methods. However, for real-world science and engineering applications, minimizing the cost of physical resources to perform evaluation (e.g., conducting an additive manufacturing experiment for designing materials such as alloys) is the most important metric. The computational cost for selecting inputs for evaluation is a secondary concern. HyBO uses more time to select inputs for evaluation to minimize the number of function

evaluations to uncover better structures. We provide a finer-analysis of the HyBO runtime in Table 4.4. Each kernel evaluation time with all orders of interactions is very small. The overall runtime is spent on two major things: a) Sampling from posterior distributions of hyperparameters using slice sampling; and b) AFO using CMA-ES + local search. We can reduce the sampling time by considering HyBO without marginalization which shows slightly worse performance, but takes only 10 percent of the sampling time in HyBO.

Orders of interaction	**HyBO iteration**	**AFO**	**Sampling**	**Kernel eval.**
2	62	46	16	0.005
5	68	50	18	0.006
10	102	68	34	0.010
20 (HyBO)	180	114	66	0.020

Table 4.4: Average runtime (seconds) for different orders of interaction within hybrid kernel for synthetic Function 3.

4.4 Summary

We studied a novel Bayesian optimization approach referred as HyBO for optimizing hybrid spaces using Gaussian process based surrogate models. We presented a principled approach to construct hybrid diffusion kernels by combining diffusion kernels defined over continuous and discrete sub-spaces in a tractable and flexible manner to capture the interactions between discrete and continuous variables. We proved that additive hybrid kernels have the universal approximation property. Our

experimental results on diverse synthetic and real-world benchmarks show that HyBO performs significantly better than state-of-the-art methods.

CHAPTER FIVE

BAYESIAN OPTIMIZATION OVER PERMUTATION SPACES

In this chapter, we consider the problem of optimizing expensive to evaluate black-box functions over an input space consisting of all permutations of d objects which is an important problem with many real-world applications. For example, placement of functional blocks in hardware design to optimize performance via simulations. The overall goal is to minimize the number of function evaluations to find high-performing permutations. The key challenge in solving this problem using the Bayesian optimization (BO) framework is to trade-off the complexity of statistical model and tractability of acquisition function optimization. We propose and evaluate two algorithms for **BO** *over* **P***ermutation* **S***paces (BOPS)*. First, BOPS-T employs Gaussian process (GP) surrogate model with Kendall kernels and a **T**ractable acquisition function optimization approach based on Thompson sampling to select the sequence of permutations for evaluation. Second, BOPS-H employs GP surrogate model with Mallow kernels and a **H**euristic search approach to optimize expected improvement acquisition function. We theoretically analyze the performance of BOPS-T to show that their regret grows sub-linearly. Our experiments on multiple synthetic and real-world benchmarks show that both BOPS-T and BOPS-H perform better than the state-of-the-art BO algorithm for combinatorial spaces. To drive future research on this important problem, we make new resources and real-world benchmarks available to the community.

5.1 Problem Setup

In this chapter, we consider optimization problems with the input space consisting of all permutations over d objects. Given $[1, d] := \{1, 2, \cdots, d\}$, indexing the d objects, a permutation is defined as a bijective mapping $\pi : [1, d] \mapsto [1, d]$. The set of all permutations along with the composition binary operation $((\pi_1 \circ \pi_2)(x) = \pi_1(\pi_2(x))$ is known as the Symmetric group $\mathcal{S}_d$ which has a cardinality $|\mathcal{S}_d| = d!$.

Let $f : \mathcal{S}_d \mapsto \mathbb{R}$ be a black-box objective function that is expensive to evaluate. Our goal is to optimize f while *minimizing the number of function evaluations*:

$$\pi^* = arg\min_{\pi \in \mathcal{S}_d} f(\pi) \tag{5.1}$$

For a concrete example problem, consider the domain of design and optimization of integrated circuits (ICs). There are many applications in IC design, where we need to optimize over permutations of functional blocks of different granularity (small cells to processing cores). Some example objectives include performance and manufacturing cost. We need to perform expensive computational simulations to evaluate each candidate permutation.

5.2 BO Algorithms for Permutation Spaces

In this section, we provide two algorithms for BO over permutation spaces that make varying trade-offs between the complexity of statistical model and tractability of acquisition function optimization. First, BOPS-T employs a simple statistical model with an efficient Semi-definite programming (SDP) relaxation based optimization method. Second, BOPS-H employs a complex statistical model and performs heuristic search for optimizing the acquisition function. We employ Gaussian processes (GPs) [180] as the surrogate model in both algorithms. GPs are effective statistical models commonly used for BO as they provide a principled framework for uncertainty estimation. They are fully characterized by a kernel k [123] which intuitively captures the similarity between two candidate inputs from the same input space.

5.2.1 BOPS-T Algorithm

Surrogate model. The similarity between any two permutations (π, π') can be naturally defined by considering the number of pairs of objects ordered in the same way or in opposite ways. This is captured by the notion of the number of discordant pairs $n_d(\pi, \pi')$, also known as Kendall-tau distance [128]. $n_d(\pi, \pi')$ counts the number

of pairs of objects ordered oppositely by π and π' as defined below:

$$n_d(\pi, \pi') = \sum_{i<j} [1_{\pi(i)>\pi(j)} 1_{\pi'(i)<\pi'(j)} + 1_{\pi(i)<\pi(j)} 1_{\pi'(i)>\pi'(j)}] \tag{5.2}$$

A related notion of concordant pairs $n_c(\pi, \pi')$ counts the number of object pairs ordered similarly by π and π':

$$n_c(\pi, \pi') = \binom{d}{2} - n_d(\pi, \pi') \tag{5.3}$$

Kendall kernels [113] are positive-definite kernels defined over permutations using the notion of discordant and concordant pairs as follows:

$$k(\pi, \pi') = \frac{n_c(\pi, \pi') - n_d(\pi, \pi')}{\binom{d}{2}} \tag{5.4}$$

Because of their proven effectiveness over permutations [113, 114], we propose using Kendall kernels with GPs as surrogate model in our BOPS-T algorithm.

For our surrogate model, we consider the weight-space formulation of the GP. This weight-space formulation is essential for the SDP based acquisition function optimization approach described in the next section. In the *weight-space* view, we

can reason about GPs as a weighted sum of basis functions $\phi = \{\phi_i(\cdot)\}$, i.e.,

$$w^T \phi(\cdot); \quad w \sim N(0, I) \tag{5.5}$$

where $N(\cdot)$ represents multi-variate Gaussian distribution and I is the identity matrix. Every kernel has a canonical feature map (as per the Moore-Aronszajn theorem [12]) $\phi : \mathcal{S}_d \mapsto H_k$, H_k being its associated Reproducing Kernel Hilbert Space (RKHS), that is employed as the basis function in 5.5. The feature map expression for Kendall kernel (constructed by [113]) is given below:

$$\phi(\pi) = \{\sqrt{\binom{d}{2}^{-1}} \left(1_{\pi(i)>\pi(j)} - 1_{\pi(i)<\pi(j)}\right)\}_{(1 \leq i < j \leq d)} \tag{5.6}$$

Acquisition function and optimizer. In order to sequentially select the next permutation for evaluation guided by the learned surrogate model, we employ Thompson sampling as our acquisition function. Thompson sampling is a powerful, practitioner-friendly, and parameter-free approach for appropriately balancing the exploration vs. exploitation dilemma [187] in sequential bandit optimization. The key idea is to sample a function from the surrogate model's posterior and select its optimizer as the next permutation for evaluation. In the weight-space view of GPs, this corresponds to sampling a weight vector $\hat{w}$ from its posterior and solving the following optimization

problem:

$$\pi_{next} = arg \min_{\pi \in \mathcal{S}_d} \hat{w}^T \phi(\pi) \tag{5.7}$$

It should be noted that the sampled weight vector $\hat{w}$ is an exact function defined by GP (with Kendall kernel) over permutation spaces and has no approximation error when compared to the function space approach. This is in contrast to the common practice of using Thompson sampling over continuous spaces, where random Fourier features based weight-space representation of GPs is used which inevitably results in approximation error because of sampling a finite number of features from an infinite feature space.

We now show that the above acquisition function optimization problem (5.7) is a Quadratic Assignment Problem (QAP) [38]. To observe that, the objective in 5.7 is written in an equivalent form in terms of P_d, the set of all possible permutation matrices P of size $d \times d$, as follows:

$$\min_{P \in P_d} Tr(WPAP^T) \tag{5.8}$$

where Tr is the matrix trace operation and A is a $d \times d$ matrix defined as follows:

$$A = \begin{cases} 1 & \text{if } i < j \\ -1 & \text{if } i > j \\ 0 & \text{if } i == j \end{cases} \quad \forall i, j \in [1, d]$$

and W is another $d \times d$ matrix given as follows:

$$W = \begin{cases} w_{\frac{(i-1)}{2}(2d-i)+(j-i)} & \text{if } i < j \\ 0 & \text{if } i \geq j \end{cases} \quad \forall i, j \in [1, d]$$

Concretely, the equivalence of objectives in 5.7 and 5.8 can be seen as follows:

$$Tr(WPAP^T) = \sum_{i=1}^{d} (WPAP^T)_{ii} \tag{5.9}$$

Equation 5.9 is the definition of the trace of a matrix. Now, considering each entry

$(WPAP^T)_{ii}$ in 5.9:

$$(WPAP^T)_{ii} = \sum_{j=1}^{d} W_{ij} \cdot (PAP^T)_{ji} \tag{5.10}$$

$$= \sum_{j>i}^{d} w_{\frac{(i-1)}{2}(2d-i)+(j-i)} \cdot (PAP^T)_{ji} \tag{5.11}$$

$$= \sum_{j>i}^{d} w_{\frac{(i-1)}{2}(2d-i)+(j-i)} \cdot A_{\pi(j)\pi(i)} \tag{5.12}$$

where 5.11 follows from the definition of W and 5.12 follows from the fact that pre-multiplying (post-multiplying) by a permutation matrix permutes the rows (columns) of A. Using 5.12 in 5.9:

$$Tr(WPAP^T) = \sum_{i=1}^{d} \sum_{j>i}^{d} w_{\frac{(i-1)}{2}(2d-i)+(j-i)} \cdot A_{\pi(j)\pi(i)} \tag{5.13}$$

By noting that $A_{\pi(j)\pi(i)}$ is exactly the feature map in 5.6 (upto multiplication by a constant $\sqrt{\binom{d}{2}^{-1}}$ which doesn't change the optimal solution), the equivalence between 5.7 and 5.8 is established.

Although, in general, Quadratic assignment probem is NP-hard [188], we leverage existing Semi-definite programming (SDP) based strong relaxations [227] to obtain good approximate solutions to the acquisition function optimization problem. Using the invariance of the trace under cyclic permutations and vectorization identity

$(vec(APW) = (W^T \otimes A)vec(P))$, objective in 5.8 is standardized as:

$$\min_{P,Q}((W^T \otimes A)Q) \tag{5.14}$$

$$P \in P_n$$

$$Q = vec(P)vec(P)^T$$

where $vec(P)$ is the column-wise vectorization of P. We leverage the clique-based SDP relaxation approach of [80] which can exploit matrix sparsity (e.g., zeros in the upper-triangular matrix W) for solving 5.14. The key idea is to enforce semi-definiteness only over groups of Q's entries (i.e., cliques) to get a relaxation that can be solved using fast and accurate algorithms.

5.2.2 BOPS-H Algorithm

Surrogate model. We propose to employ Mallows kernel which plays a role on the symmetric group $\mathcal{S}_d$ similar to the Gaussian (RBF) kernel on the Euclidean space. Given a pair of permutations π and π', the Mallows kernel is defined as the exponentiated negative of the number of discordant pairs $n_d(\pi, \pi')$ between π and π' i.e.

$$k_m\pi, \pi' = \exp(-ln_d(\pi, \pi')) \tag{5.15}$$

where $l \geq 0$ is a hyper-parameter of the kernel similar to the length-scale hyper-parameter of the Gaussian kernels on Euclidean space. A key measure of the expressivity of a kernel is based on a property called *universality* which captures the notion of whether the RKHS of the kernel is rich enough to approximate any function on a given input space arbitrary well. It was recently shown [150] that Mallows kernel is *universal* over the space of permutations in contrast to the Kendall kernel discussed in the previous section. Therefore, Mallows kernels are more powerful than Kendall kernels and allows us to capture richer structure in permutations when used to learn GP based surrogate models. Indeed, our experiments also demonstrate empirically the superior modeling capability of Mallows Kernel.

Acquisition function and optimizer. Unlike Kendall kernel, the feature space of Mallows Kernel is exponentially large [150] making it practically inefficient to sample functions from the GP posterior (in the weight-space style as described earlier).Therefore, we propose to employ expected improvement (EI) as our acquisition function. The additional complexity of GP based statistical model with Mallows kernel makes the acquisition function optimization problem $\pi_{next} = arg\min_{\pi \in \mathcal{S}_d} AF(\pi)$ is intractable for EI. Therfore, we propose to perform **H**euristic search in the form of local search with multiple restarts that has been shown to be very effective in practice for solving combinatorial optimization problems. To search over only valid permutations $\pi \in \mathcal{S}_d$, at each local search step, we consider only those neighbors

which are permutations of the current state. Otherwise, we will be searching over a huge combinatorial space with both valid (permutations) and invalid structures (non-permutations) as done by COMBO: may not result in producing a permutation from its acquisition function optimization procedure. Indeed, we observed this behavior in our experiments with COMBO. We use the modified local search procedure over permutations for both COMBO and our BOPS-H algorithm in experiments.

5.3 Theoretical Analysis for BOPS-T

In this section, we analyze the theoretical properties of our BOPS-T algorithm in terms of regret metric [197], which is a commonly used measure for analyzing BO algorithms. Note that there is no prior regret bound analysis for BO algorithms for EI even in continuous spaces. Hence, we leave the analysis of BOPS-H algorithm for future work. Let simple regret R be defined as follows:

$$R = \sum_{t=1}^{T} (f(\pi_t) - f(\pi^*)) \tag{5.16}$$

where π_t is the permutation picked by the BO algorithm at time (iteration) t. In our case of using Thompson sampling as an acquisition function, it is natural to consider the expected form of this regret [187] where the expectation is taken over the distribution of functions as given by the GP prior with Kendall kernel. We analyze

this expected form of regret, also known as Bayesian regret:

$$\mathcal{BR} = \sum_{t=1}^{T} \mathbb{E}(f(\pi_t) - f(\pi^*)) \tag{5.17}$$

where the expectation is over the distribution of functions $f \sim GP(0, k)$. The below theorem bounds the Bayesian regret of our BOPS-T algorithm:

Theorem 9. *Let $f \sim GP(0, k)$ with Kendall kernel k (5.4), the Bayesian regret of the BOPS-T algorithm after T observations $y_i = f(\pi_i) + \epsilon_i, i \in \{1, 2, \cdots T\}$ with ϵ_i being Gaussian distributed i.i.d. noise $\epsilon_i \sim N(0, \sigma^2)$ is : $\mathcal{BR} = \mathcal{O}^*(d^{3/2}\sqrt{T})$, where $\mathcal{O}^*$ denotes upto log factors.*

Proof. The key quantity in bounding the regret of Bayesian optimization with Gaussian processes (also known as GP bandits) is an information-theoretic quantity called as *maximum information gain* γ_T [197] that depends on the kernel k and intuitively captures the maximum information that can be gained about f after T observations, i.e.,

$$\gamma_T = \max_{A \subset \mathcal{S}_d, |A|=T} I(\mathbf{y_A}; f) \tag{5.18}$$

where I is the mutual information and A is a subset of permutations with corresponding function evaluations $\mathbf{y_A}$.

[187] proved the Bayesian regret for Thompson sampling by characterizing it in terms of upper confidence bound based results from [197]:

Proposition 10. *(Proposition 5 [187]). If $|X| < \infty$, $\{f(x) : x \in X\}$ follows a multivariate Gaussian distribution with marginal variances bounded by 1, the Bayesian regret for Thompson sampling based bandit policy is given as:*

$$\mathcal{BR} = 1 + 2\sqrt{T\gamma_T \ln(1+\sigma^2)^{-1} \ln\left(\frac{(T^2+1)|X|}{\sqrt{2\pi}}\right)} \tag{5.19}$$

where X is the action space.

This proposition is directly applicable in our setting because the action space, being the cardinality of the symmetric group $\mathcal{S}_d$, is finite (i.e., $|\mathcal{S}_d| = d!$) and the function $\{f(\pi) : \pi \in \mathcal{S}_d\}$ follows a multivariate Gaussian distribution (by definition of Gaussian process with Kendall kernel). We compute the specific terms in the right-hand side of 5.19 that are applicable in our setting to prove the regret bound.

The maximum information gain for kernels with finite feature maps can be computed in the weight-space form (Sec 5.2.1) as a special case of linear kernel [197].

$$\gamma_T \leq C \log |I + \sigma^{-2}K| \tag{5.20}$$

where $C = 1/2 \cdot (1 - 1/e)^{-1}$ is a constant, K is a $T \times T$ matrix with each entry K_{ij}

$= k(\pi_i, \pi_j)$. As per kernel trick,

$$K = \Phi^T \Phi \tag{5.21}$$

where Φ is a matrix with $\Sigma^{1/2}\phi(\pi_i), i \in \{1, 2, \cdots, T\}$ as the columns (5.6). Therefore,

$$\gamma_T \leq C \ln |I + \sigma^{-2} \Phi^T \Phi| \tag{5.22}$$

By Schur's complement:

$$\gamma_T \leq C \ln |I + \sigma^{-2} \Phi^T \Phi| \leq C \ln |I + \sigma^{-2} \Phi \Phi^T| \tag{5.23}$$

By Hadamard's inequality:

$$\gamma_T \leq C \ln |I + \sigma^{-2} \Phi \Phi^T| \tag{5.24}$$

$$\leq C \sum_{i=1}^{\binom{d}{2}} \ln(1 + \sigma^{-2} \lambda_i) \tag{5.25}$$

where $\{\lambda_1, \lambda_2, \cdots\}$ is the eigenvalue set of the matrix $\Phi\Phi^T$.

By Gershgorin circle theorem [213], all the eigenvalues of a matrix is upper bounded by the maximum absolute sum of rows, i.e. $\lambda_i \leq d^2 T$ with the assump-

tion that $\|\Sigma^{1/2}\phi(\pi)\| \leq 1$.

$$\gamma_T = O(d^2 \ln(d^2 T)) \tag{5.26}$$

Now, using Stirling's approximation, we can bound the $\ln(|X|)$ term in 5.19, where $|X| = |\mathcal{S}_d|$ in our case:

$$\ln(|\mathcal{S}_d|) = O(d \ln d) \tag{5.27}$$

Plugging 5.26 and 5.27 in 5.19 and ignoring constants, we get the following expression:

$$\mathcal{BR} = O(\sqrt{T d^2 \ln d^2 T (\ln T^2 + d \ln d)}) \tag{5.28}$$

$$\mathcal{BR} = O(\sqrt{(T d^2 \ln d^2 T \ln T^2 + T d^3 \ln d^2 T \ln d)}) \tag{5.29}$$

$$\mathcal{BR} = \mathcal{O}^*(d^{3/2}\sqrt{T}) \tag{5.30}$$

Hence, ignoring log factors, our proposed BOPS-T algorithm achieves sublinear (time) regret.

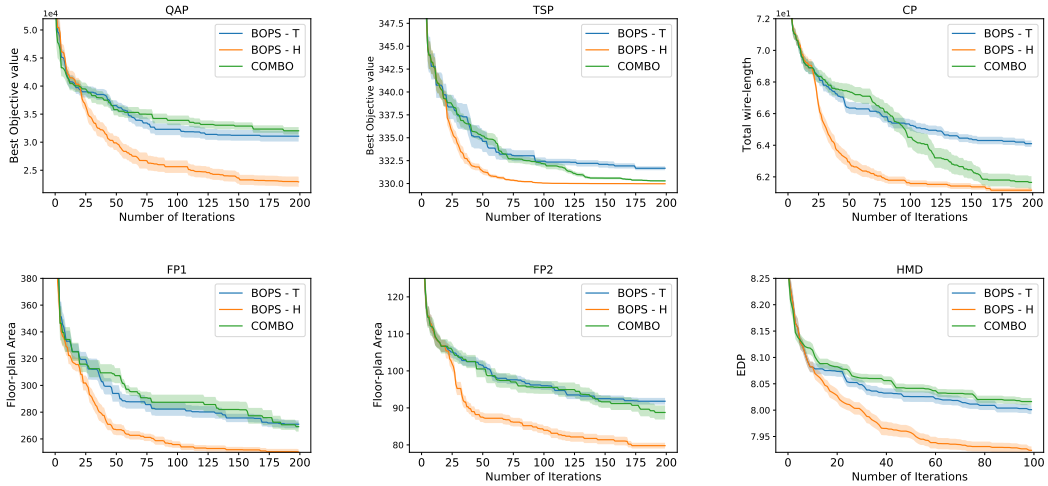


Figure 5.1: Results comparing BOPS-T, BOPS-H, and COMBO (best objective function value vs. number of BO iterations) on both synthetic and real-world benchmarks: (Top row) QAP, TSP, CP; and (Bottom row) FP1, FP2, and HMD.

5.4 Experiments and Results

In this section, we describe the benchmarks and experimental setup followed by results and discussion.

5.4.1 Benchmarks

We employ diverse and challenging benchmarks for black-box optimization over permutations for our experiments. We have the following two synthetic benchmarks.

1) Quadratic assignment problem (QAP). QAPLIB [37] is a popular library that contains multiple QAP instances. Each QAP instance contains a cost matrix (A) and distance matrix (B) sized $n \times n$, where n is the number of input dimensions.

The goal is to find the best permutation that minimizes the quadratic assignment objective $Tr(APBP^T)$, where P is an $n \times n$ permutation matrix. We use input space with $n = 15$ dimensions in our experiments.

2) Traveling salesman problem (TSP). TSP problems are derived from low-dimensional variants of the printed circuit board (PCB) problems from the TSPLIB library [181]. The overall goal is to find the route of drilling holes in the PCB that minimizes the time taken to complete the job. We use input space with $d = 10$ dimensions from the data provided in the library.

We perform experiments on three important real-world applications from the domain of computer-aided design of integrated circuits (ICs). These applications are characterized by permutations over functional blocks at different levels of granularity that arise in different stages of design and optimization of ICs. Importantly, even tiny improvements in solution has huge impact (e.g., improved performance over the lifespan of the IC or reduced cost for manufacturing large samples of the same IC). A big challenge in the combinatorial BO literature is the availability of challenging real-world problems to evaluate new approaches. Hence, we provide our three real-world benchmarks as a new resource to allow rapid development of the field.

3) Floor planning (FP). We are given k rectangular blocks with varying width and height, where each block represents a functional module performing certain task. Each placement of the given blocks is called a *floor-plan*. Our goal is to find the

floor plan that minimizes the manufacturing cost per chip. We use two variants of this benchmark with 10 blocks (FP1 and FP2) that differ in the functionality of the blocks.

4) Cell placement (CP). We are given 10 rectangular cells with same height and a netlist that contains the connection information among the cells. The goal is to place the 10 rectangular cells for optimizing the performance of the circuit. Intuitively, shorter nets have shorter delays, so placements with shorter wire-length will result in higher performance.

5) Heterogeneous manycore design (HMD). This is a manycore architecture optimization problem from the rodinia benchmark [43]. We are given 16 cores of three types: 2 CPUs, 10 GPUs, and 4 memory units. They are connected by a mesh network (each core is connected to its four neighboring cores) to facilitate data transfer. The goal is to place the given 16 cores to optimize the energy delay product (EDP) objective that captures both latency and energy, two key attributes of a manycore chip.

5.4.2 Experimental Setup

Configuration of algorithms. We compare our proposed BOPS-T and BOPS-H algorithms with the state-of-the-art combinatorial BO algorithm COMBO [**?**]. COMBO employs a diffusion kernel based GP surrogate model and optimizes expected improvement acquisition function using local search with restarts to select inputs for

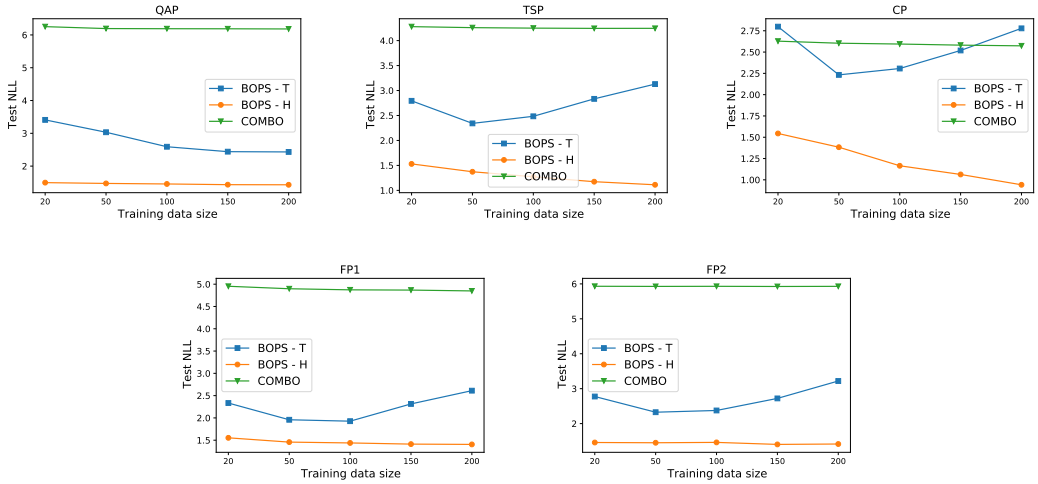


Figure 5.2: Results comparing the three surrogate models of BOPS-T, BOPS-H and COMBO on negative log-likelihood (NLL) metric computed on a test set on five benchmarks: (Top row) QAP, TSP, CP; and (Bottom row) FP1, FP2.

evaluation. Each local search step considers all neighbors of the current structure in the combinatorial graph (i.e., structures with Hamming distance one). We modify COMBO's local search procedure (`https://github.com/QUVA-Lab/COMBO`) to consider only those neighbors which are permutations of the current state thereby helping COMBO to avoid searching a large combinatorial space with huge number of invalid structures (non-permutations).

We used the SDP relaxation based QAP solver code from (`https://github.com/fsbravo/csdp`) for implementing BOPS-T. BOPS-H is built using popular GPyTorch [87] and BoTorch [15] libraries. We used 10 restarts for local search based EI optimization for BOPS-H. BOPS-T, BOPS-H, and COMBO are initialized with the same 20 random permutations in each experiment.

Evaluation metric. We plot the objective function value of the best permutation over different BO iterations. Each experiment is repeated 20 times and we plot the mean of the best objective value plus and minus the standard error.

5.4.3 Results and Discussion

In this section, we present and discuss our experimental results along different dimensions.

Figure 5.1 shows the results for BO performance (best objective value vs. number of function evaluations / BO iterations) of BOPS-T, BOPS-H, and COMBO on all six benchmarks. Below we discuss these results in detail.

BOPS-T vs. BOPS-H. Recall that BOPS-T and BOPS-H makes varying trade-offs between the complexity of statistical model and tractability of acquisition function optimization: BOPS-T uses simple model and tractable search; and BOPS-H employs complex model and heuristic search. From Figure 5.1, we can observe that BOPS-H performs significantly better than BOPS-T on all six benchmarks.

BOPS vs. COMBO. From the results shown in Figure 5.1, we make the following observations: 1) BOPS-H performs significantly better than both BOPS-T and COMBO on all six benchmarks; and 2) BOPS-T is comparable or slightly better than COMBO on all benchmarks except TSP and CP.

We hypothesize that the performance of different BO algorithms, namely, BOPS-H, BOPS-T, and COMBO is proportional to the quality of their surrogate models in

terms of making predictions on unknown permutations and their uncertainty quantification ability. To verify this hypothesis, we compare the three surrogate models quantitatively in terms of their performance on the log-likelihood metric.

Comparison of surrogate models. We compare the three surrogate models on the log-likelihood metric [158] because it captures both the prediction and uncertainty quantification of a model which are essential for the effectiveness of BO. We plot the negative log-likelihood (NLL) of the three surrogate models on a testing set of 50 instances as a function of the increasing size of training data. Each experiment is replicated with 10 different training sets and each method is evaluated using the median of the NLL metric on 10 different test sets of 50 permutations each. Figure 5.2 shows the results on all benchmarks except HMD. We do not show results on HMD since each function evaluation is much more expensive when compared to all other benchmarks, and we are generating multiple replications of the training and testing sets ($10 \times 10 = 100$ runs). We make the following observations from Figure 5.2: 1) BOPS-H shows the best performance among the three surrogate models; 2) BOPS-T does better than COMBO on all benchmarks other than cell-placement. Since both COMBO and BOPS-H employ the same acquisition function (EI) and optimizer (local search), it is evident that the gains in the BO performance comes from the superior surrogate model of BOPS-H.

5.5 Summary

We proposed and evaluated two effective Bayesian optimization algorithms with varying trade-offs for optimizing expensive black-box functions over the challenging input space of permutations. The results point to a key conclusion that it is important to use an appropriate model that exploits the specific structure of permutation spaces, which is different than the generic combinatorial space over categorical variables. We characterized the importance of this problem setting by describing three important real-world applications from the domain of computer-aided design of integrated circuits. Furthermore, we make all these benchmarks available to drive future research in this problem space.

CHAPTER SIX

SURROGATE MODELS COMBINING STRENGTHS OF DEEP GENERATIVE MODELS AND STRUCTURED KERNELS

In this chapter, we consider the problem of optimizing expensive black-box function defined over richer varying-sized combinatorial spaces (e.g., sequences, trees, and graphs). For example, optimizing molecules for drug design using physical lab experiments. A recent BO approach for combinatorial spaces is through a reduction to BO over continuous spaces by learning a latent representation of structures using deep generative models (DGMs). The selected input from the continuous space is decoded into a discrete structure for performing function evaluation. However, the surrogate model over the latent space only uses the information learned by the DGM, which may not have the desired inductive bias to approximate the target black-box function. To overcome this drawback, this chapter proposes a principled approach referred as LADDER. The key idea is to define a novel structure-coupled kernel that explicitly integrates the structural information from decoded structures with the learned latent space representation for better surrogate modeling. Our experiments on real-world benchmarks show that LADDER significantly improves over the BO over latent space method, and performs better or similar to state-of-the-art methods.

6.1 Problem Setup and Background

Let $\mathcal{X}$ be a space of combinatorial structures (e.g., sequences, trees, and graphs). We assume the availability of a black-box objective function $f : \mathcal{X} \mapsto \mathbb{R}$ defined over the combinatorial space $\mathcal{X}$. Evaluating each candidate structure $\mathbf{x} \in \mathcal{X}$ using function f (also called an experiment) is *expensive* in terms of the resources consumed and produces an output $y = f(\mathbf{x})$. For example, in the drug design application, each $\mathbf{x} \in \mathcal{X}$ is a molecule, and $f(\mathbf{x})$ corresponds to running a physical lab experiment. Our overall goal is to find a structure $\mathbf{x} \in \mathcal{X}$ that approximately optimizes f by minimizing the number of experiments and observing their outcomes.

We are also provided with a database of *unsupervised* structures $\mathcal{X}_u \subset \mathcal{X}$. Unsupervised means that we do not know the function evaluations $f(x)$ for structures in $\mathcal{X}_u$. This assumption is satisfied by many scientific applications including chemical design and material design. We assume the availability of a *latent space* $\mathcal{Z}$ learned from unsupervised structures $\mathcal{X}_u$ using a encoder-decoder style deep generative model, e.g., variational autoencoders (VAEs) for structured data such as junction tree VAE [115] and grammer VAE [121]. Formally, the encoder denoted by Υ embeds a given combinatorial structure $\mathbf{x} \in \mathcal{X}$ into a point in the latent space $\mathbf{z} \in \mathbb{R}^d = \Upsilon(x)$ where d is the number of dimensions of the latent space $\mathcal{Z}$ and the decoder denoted by Φ converts a given point from latent space $\mathbf{z}' \in \mathcal{Z}$ into a structured object $\mathbf{x}' \in \mathcal{X} = \Phi(\mathbf{z}')$. Encoder Υ and decoder Φ are typically realized by neural networks.

6.2 LADDER: Latent Space BO guided by Decoded Structures

In this section, we first discuss the challenges with the Naïve latent space BO approach. Next, we describe our proposed LADDER approach with a focus on the novel surrogate model by combining kernels over structured data and latent space representation, which is our key technical contribution.

6.2.1 Challenges with the Naïve Latent Space BO approach

As mentioned above, the Naïve latent space BO approach builds a surrogate model over the latent space using kernels for continuous spaces (e.g., Matern or Squared Exponential kernel) and performs acquisition function optimization in the latent space using optimizers for continuous spaces (e.g., gradient-based methods) to select point $\mathbf{z} \in \mathcal{Z}$ for evaluation. However, the expensive objective function $f(\mathbf{x})$ is defined over the space of combinatorial structures $\mathcal{X}$ and *not* the latent space $\mathcal{Z}$. Therefore, we need to decode this point $\mathbf{z}$ using the decoder Φ to get the corresponding combinatorial structure $\mathbf{x}{=}\Phi(z)$ for function evaluation $f(\mathbf{x})$. All the existing work on BO over latent space do not account for this decoding process. As a consequence, we need to deal with two inter-related challenges, which are especially significant for small-data settings.

- **Challenge #1:** The kernel over the latent space only uses the information learned by the deep generative model. It doesn't explicitly incorporate infor-

mation about the decoded structure. This means the corresponding Gaussian process surrogate model may not have the desired inductive bias to approximate the black-box objective function. Therefore, we are not able to leverage this potentially useful inductive bias and rich structural information available in decoded structures.

- **Challenge #2:** The surrogate statistical model itself might not generalize well beyond the training examples from the latent space (set of points in the latent space and their corresponding function evaluations). This is especially true in the small-data setting for latent spaces learned using DGMs in real-world scientific applications, where the number of dimensions of latent space can be large when compared to the standard BO setup.

Indeed, we provide empirical evidence to demonstrate these challenges and our key hypothesis in Figure 6.2. We show that by incorporating the rich structural information from the decoded output, we can address both these challenges to improve the overall BO performance.

6.2.2 Overview of LADDER Algorithm and Key Advantages

LADDER is an instantiation of the latent space BO framework that employs a novel surrogate statistical model to address the two challenges with the Naïve method. The surrogate model is a Gaussian process that combines the complementary

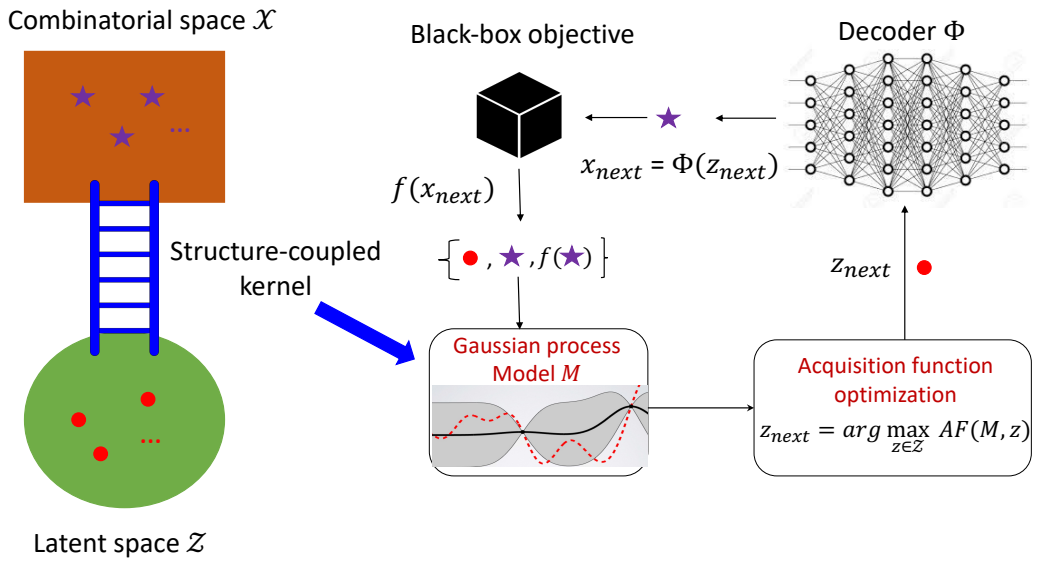


Figure 6.1: High-level conceptual illustration of our proposed LADDER approach, which acts as a "ladder" in connecting the rich structural information of each structure in the combinatorial space with its corresponding latent space representation. Structure-coupled kernel is the key element that enables this connection to build an effective Gaussian process based surrogate model.

strengths of the latent space representation with the rich structural information from decoded outputs using structured kernels (e.g., string kernels and graph kernels). The key idea is to define a *structure-coupled kernel* that extends the continuous kernel on the evaluated points in the latent space to unknown points using the rich information from structured kernels. We employ expected improvement (EI) as the acquisition function. For optimizing the acquisition function to select high utility inputs from the latent space, we employ evolutionary search due to its recent successes [146] including policy search in high-dimensional spaces [189].

Figure 6.1 shows a high-level illustration of LADDER and Algorithm 1 provides the complete pseudo-code. We use a small set of initial training data in the form of points in the latent space and their corresponding function evaluations to bootstrap the GP based surrogate model using the structure-coupled kernel. In each iteration t, we optimize the acquisition function to select a point $\mathbf{z}_t$ from the latent space $\mathcal{Z}$ for evaluation. The corresponding decoded structure $\mathbf{x}_t{=}\Phi(\mathbf{z}_t)$ is evaluated to measure the outcome $f(\mathbf{x}_t)$. The GP model with structure-coupled kernel is updated using the new 3-tuple training example $\{\mathbf{z}_t, \mathbf{x}_t, f(\mathbf{x}_t)\}$. We repeat these sequence of steps until covergence or maximum query budget and then return the best uncovered combinatorial structure $\hat{\mathbf{x}} \in \mathcal{X}$ as the output.

Advantages of LADDER. Some of the key advantages of our proposed approach are listed below.

- We are allowed to employ any existing trained deep generative model for structured data in a *plug-and-play* manner with LADDER. Therefore, any advances in the latent-space generative modeling technology will directly improve the overall BO performance.

- LADDER is a generic approach that is applicable to any combinatorial space of structures (e.g., sequences, trees, graphs, sets, and permutations). This method just requires an appropriate structured kernel over the given combinatorial space. Therefore, we can leverage a large body of research on generic kernels over structured data (e.g., string kernels and graph kernels) and hand-designed kernels based on domain knowledge for specific applications. For example, in our experiments, we employ sub-sequence string kernel (generic) and fingerprint kernel (domain-specific) to concretely instantiate the LADDER approach for strings and molecules respectively to demonstrate its flexibility.

- Combines the complementary strengths of latent space representations and structured kernels in a principled manner to create highly-effective surrogate statistical models.

6.2.3 Novel Surrogate Statistical Model via Structure-coupled Kernel

We consider *Gaussian process (GP)* [180] as the surrogate model of the expensive black-box objective function $f(\mathbf{x} \in \mathcal{X})$. GPs are known to have excellent statistical

Algorithm 6 Latent Space Bayesian Optimization guided by Decoded Structures (LADDER)

Input: Objective function $f(\mathbf{x})$, Encoder (Υ) - Decoder (Φ) style model for latent space $\mathcal{Z}$, Kernel for latent space $l(\mathbf{z_i} \in \mathcal{Z}, \mathbf{z_j} \in \mathcal{Z})$, Kernel for combinatorial space $k(\mathbf{x_i} \in \mathcal{X}, \mathbf{x_j} \in \mathcal{X})$
Initialize dataset $\mathcal{D}_0$ by evaluating few random points: $\mathcal{D}_0 \leftarrow \{\mathbf{Z}_0, \mathbf{X}_0, f(\mathbf{X}_0)\}$; $t \leftarrow 0$
// slight abuse of notation here since $\mathbf{Z}_0$ is the set of initial points from latent space $\mathcal{Z}$ and $\mathbf{X}_0$ is the set of corresponding decoded structures from $\mathcal{X}$ with function evaluations $f(\mathbf{X}_0)$
repeat
Learn Gaussian process model on the dataset $\mathcal{D}_t$ with the proposed kernel in Equation 6.4
Optimize acquisition function over the latent space $\mathcal{Z}$ to find the next point $\mathbf{z}_t$ for evaluation
Compute the decoded structure $\mathbf{x}_t$ for point $\mathbf{z}_t$ using decoder Φ
Evaluate the combinatorial structure $\mathbf{x}_t$ to get $f(\mathbf{x}_t)$
Add new training triple: $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\mathbf{z}_t, \mathbf{x}_t, f(\mathbf{x}_t)\}$; increment the iteration $t \leftarrow t + 1$
UNTIL convergence or maximum iterations
Output: best uncovered structure and the corresponding function value

properties including principled uncertainty quantification, which is critical for the effectiveness of BO algorithms. A GP model defines a prior distribution on functions which is entirely characterized by a kernel defined over a pair of inputs. Most of the existing work on latent space BO employs a standard continuous space kernel represented by $l(\mathbf{z_i} \in \mathcal{Z}, \mathbf{z_j} \in \mathcal{Z})$, e.g., Radial Basis Function (RBF) / Gaussian and Matern kernels, over points in the latent space to create the GP model. However, this surrogate model is highly-ineffective for small data settings, especially when the number of dimensions of latent space is large, which is the common case for deep generative models for scientific applications.

Structure-coupled kernel. We propose to utilize the rich structural information that is available from the decoded combinatorial structure $\mathbf{x}$ corresponding to each point $\mathbf{z}$ from the latent space $\mathcal{Z}$. The key idea behind our approach is to integrate the structure information from the decoded outputs with the learned representation of inputs from the latent space to achieve better surrogate modeling performance. To include this structure information in a principled manner within a GP model, we leverage the Generalized Nystrom extension idea [191, 219] to *extrapolate* the eigenfunctions of the kernel matrix over latent space ($\mathbf{L} = \{L_{ij} = l(\mathbf{z_i}, \mathbf{z_j}) | \mathbf{z_i}, \mathbf{z_j} \in \mathcal{Z}\}$) with basis functions from a kernel $k(\mathbf{x}_i \in \mathcal{X}, \mathbf{x}_j \in \mathcal{X})$ defined over the decoded combinatorial structures.

Without loss of generality, let m be the number of evaluated inputs from the latent space, which are denoted as $\mathbf{Z} = \{\mathbf{z_1}, \mathbf{z_2}, \cdots, \mathbf{z_m}\}$. For example, these are the points accumulated after m iterations of the latent space BO approach. Let the corresponding set of decoded structures be $\mathbf{X} = \{\mathbf{x_1} = \Phi(\mathbf{z_1}), \mathbf{x_2} = \Phi(\mathbf{z_2}), \cdots, \mathbf{x_m} = \Phi(\mathbf{z_m})\}$ with their function evaluations $\{f(\mathbf{x_1}), f(\mathbf{x_2}), \cdots, f(\mathbf{x_m})\}$. Given kernels $l : \mathbf{z} \times \mathbf{z} \to \Re$ and $k : \mathbf{x} \times \mathbf{x} \to \Re$ with $\mathbf{L}$ and $\mathbf{K}$ representing their corresponding kernel matrices, Generalized Nystrom extension generates an m-dimensional feature vector $\xi(\mathbf{z})$ for a point $\mathbf{z}$ in the latent space. The ith component of $\xi(\mathbf{z})$ is given as

follows:

$$\xi_i(\mathbf{z}) = \mathbf{k_z}^T \mathbf{K}^{-1} v_i \quad i \in \{1, 2, \cdots, m\} \tag{6.1}$$

where $\mathbf{k_z}$ is an m-dimensional vector evaluated between $\mathbf{x}$ (decoded output of latent space input $\mathbf{z}$) and other combinatorial structures from $\mathbf{X}$:

$$\mathbf{k_z} = \left[k\left(\Phi(\mathbf{z}), \Phi(\mathbf{z_1})\right), \cdots, k\left(\Phi(\mathbf{z}), \Phi(\mathbf{z_m})\right)\right] = \left[k\left(\mathbf{x}, \mathbf{x_1}\right), \cdots, k\left(\mathbf{x}, \mathbf{x_m}\right)\right] \tag{6.2}$$

$\mathbf{K}$ is an $m \times m$ kernel matrix for combinatorial structures in the set $\mathbf{X}$, i.e., $K_{ij} = k(\Phi(\mathbf{z_i}), \Phi(\mathbf{z_j})) = k(\mathbf{x_i}, \mathbf{x_j})$, and v_i is the eigenvalue-scaled eigenvector of the kernel matrix $\mathbf{L}$ defined over the latent space inputs in the set $\mathbf{Z}$, i.e. $V = [v_1, v_2, \cdots v_m] = U\Sigma^{1/2}$, where U and Σ are eigenvectors and eigenvalues of $\mathbf{L}$ respectively.

The reader should note that the term $k(\Phi(\mathbf{z_i}), \Phi(\mathbf{z_j}))$ in (6.2) means that the structured kernel is applied to the decoded structures $\mathbf{x_i} = \Phi(\mathbf{z_i})$ and $\mathbf{x_j} = \Phi(\mathbf{z_j})$ of latent space inputs $\mathbf{z_i}$ and $\mathbf{z_j}$ respectively. This is a key term in the above expression because it integrates each input from latent space with its decoded structure in the new feature map $\xi(\mathbf{z})$. For any two inputs $\mathbf{z}$ and $\mathbf{z}'$ in the latent space, the resulting structure-coupled kernel denoted as $c(\mathbf{z}, \mathbf{z}')$ is defined as the dot product of their

corresponding feature vectors $\xi(\mathbf{z})$ and $\xi(\mathbf{z}')$:

$$c(\mathbf{z}, \mathbf{z}') = \xi(\mathbf{z})^T \xi(\mathbf{z}') \tag{6.3}$$

$$c(\mathbf{z}, \mathbf{z}') = \mathbf{k_z}^T \mathbf{K}^{-1} \mathbf{L} \mathbf{K}^{-1} \mathbf{k_{z'}} \tag{6.4}$$

Intuitively, by this construction, we are extending the kernel over the latent space l on the evaluated points $\mathbf{Z}$ to non-evaluated points in the latent space by utilizing the rich structural features from kernel k defined over combinatorial spaces. For training points (candidate inputs from latent space along with their function evaluations), the resulting kernel matrix will be $\mathbf{L}$ since Equation 6.4 becomes $\mathbf{K}\mathbf{K}^{-1}\mathbf{L}\mathbf{K}^{-1}\mathbf{K} = \mathbf{L}$. For latent space points not in the training set, the structured kernel k acts like a smooth extrapolating kernel. It can also be seen by interpreting the Equation 6.4 through the definition of a kernel in terms of the empirical kernel map [1](with respect to the decoded structured outputs) endowed with a dot product induced by the positive [2] definite matrix $\mathbf{K}^{-1}\mathbf{L}\mathbf{K}^{-1}$, i.e.,

$$c(\mathbf{z}, \mathbf{z}') = < \mathbf{k_z}, \mathbf{k_{z'}} >_{\mathbf{K}^{-1}\mathbf{L}\mathbf{K}^{-1}} = < \mathbf{k_z}, \mathbf{K}^{-1}\mathbf{L}\mathbf{K}^{-1}\mathbf{k_{z'}} > \tag{6.5}$$

Importantly, the above general construction of structure-coupled kernel allows us

[1]We refer to empirical kernel map as commonly defined in [190] (Definition 3)

[2]Positive definiteness of $\mathbf{K}^{-1}\mathbf{L}\mathbf{K}^{-1}$ can be easily seen as a consequence of the following three facts: i) $\mathbf{K}$ and $\mathbf{L}$, being kernel matrices, are positive definite; ii) inverse of a psd matrix is psd; and iii) MNM is psd if M and N are two psd matrices.

to leverage extensive research on kernel methods for highly-structured data, which try to exploit structural features of combinatorial objects. For example, string kernels [144] count the number of common sub-strings in string inputs, fingerprint kernels [179] capture neighborhood-aggregated properties of molecules, and features such as number of random walks and shortest paths are utilized by graph kernels [34].

6.3 Experiments and Results

In this section, we empirically evaluate the effectiveness of the proposed LADDER approach on real-world benchmarks, and perform comparison with baseline methods.

6.3.1 Real-world Benchmarks

We employ two widely used real-world benchmarks for combinatorial Bayesian optimization.

Arithmetic expressions optimization. In this benchmark, the goal is to search in the space of uni-variate arithmetic expressions (generated from a given grammar) to find the best expression that fits a given target expression [121]. As described in [121], the latent space model is trained on 100K randomly generated expressions from

the following grammar:

```
S → S '+' T | S '*' T | S '/' T | T

T → '(' S ')' | 'sin(' S ')' | 'exp(' S ')'

T → 'v' | '1' | '2' | '3'
```

We follow the same setup as discussed in the state-of-the-art paper for this benchmark [210]. We consider the log mean-squared error between an expression $\mathbf{x}$ and the target expression $\mathbf{x}^* = 1/3 \cdot v \cdot \sin(v^2)$ (computed over 1000 evenly-spaced values of v in the interval $[-10, 10]$) as the objective function which should be minimized.

Chemical design optimization. This benchmark considers finding molecules with best drug-like properties [121] and is similar in prototype for many scientific applications. Specifically, the goal is to maximize the water-octanol partition coefficient (logP) over the space of molecules. The latent space model is trained on the Zinc molecule dataset of 250K molecules. For consistency purposes, all our results are shown as minimization obtained by taking a negation of the logP objective.

6.3.2 Experimental Setup

Configuration of algorithms. The BO part of the source code is written using the popular GpyTorch [85] and BoTorch [14] libraries for all BO methods including LADDER. We employ ARD (automatic relevance determination) Matern kernel for

the latent space inputs in all our experiments. Matern kernel is commonly advocated as a better choice than RBF kernel for BO algorithms since the sample functions from the latter are impractically smooth [196]. Hyperparameters of Gaussian process models are fitted by marginal likelihood maximization after every BO iteration. We employed Junction tree VAE [115] and Grammar VAE [121] as the latent-space model for chemical design and arithmetic expression optimization benchmarks respectively. Both pretrained encoder-decoder models are taken from the source code provided by the authors' of [210] [3]. We employed expected improvement (EI) as the acquisition function for all the BO methods. All experiments are performed on a machine with the following configuration: Intel(R) Core(TM) i9-7960X CPU @ 2.80GHz with 128 GB RAM.

LADDER instantiations. In addition to the encoder-decoder style latent space model, which is same as the Naïve latent space BO (LSBO), LADDER also requires an appropriate structured kernel for the given combinatorial space. To demonstrate the generality of our proposed approach, we instantiate LADDER with two different kernels for our two optimization benchmarks. We employed sub-sequence string kernel for the arithmetic expressions task and fingerprints kernel for the chemical design task. We briefly describe both kernels below.

- *Sub-sequence string kernel.* This kernel captures the similarity between two

[3] `https://github.com/cambridge-mlg/weighted-retraining/`

strings by counting the number of matching substrings, where the substrings can be non-contiguous [144, 40]. Following the notation in [157], given an alphabet Π, the kernel between two strings s_1 and s_2 is given as: $k(s_1, s_2) = \sum_{u \in \Pi^n} \rho(s_1)\rho(s_2)$ where $\rho(s) = \lambda_m^{|u|} \cdot \sum_{1<i_1<\cdots<i_{|u|}<|s|} \left(\lambda_g^{i_{|u|}-i_1} \mathbf{1}_u(s_{i_1}, \cdots, s_{i_{|u|}})\right)$ for any string s where λ_g and λ_m are gap decay and match decay hyperparameters. Since arithmetic expressions are naturally represented as strings, we use this kernel for the arithmetic expressions task.

- *Fingerprints kernel.* There is a large body of work in the chemical informatics literature for designing structural features for molecular inputs. These hand-engineered features by domain experts are commonly known as molecular fingerprints [152]. We employ Morgan fingerprints [183] which are high-dimensional binary features to capture different substructures in a molecule while being invariant to atom relabeling. Since combinatorial structures in the chemical design task are molecules, given two molecules m_1 and m_2, we consider the dot product of their Morgan fingerprints as the structured kernel for the chemical design task. Hence, we refer to this kernel as the Fingerprints kernel. We employed 2048-bit fingerprints with a bond radius of 3 [156].

In all our results and figures, the corresponding structured kernel for LADDER is denoted in the parenthesis, e.g., LADDER (String). We employed the evolutionary search algorithm CMA-ES [97] as the acquisition function optimizer for LADDER.

The parameters of CMA-ES[4] were fixed with sigma = 0.2 and a population size of 50. The performance of CMA-ES was found to be highly robust to different choice of these parameters. We ran CMA-ES from 10 different starting inputs for 10 iterations each and picked the best optimizer found. As discussed later, we consider one instance of Naïve LSBO with the same configuration of CMA-ES optimizer for fair comparison and to test our key hypothesis that surrogate modeling within LADDER is better. We use 10 random points (uniformly picked from the dataset) to initialize the GP models.

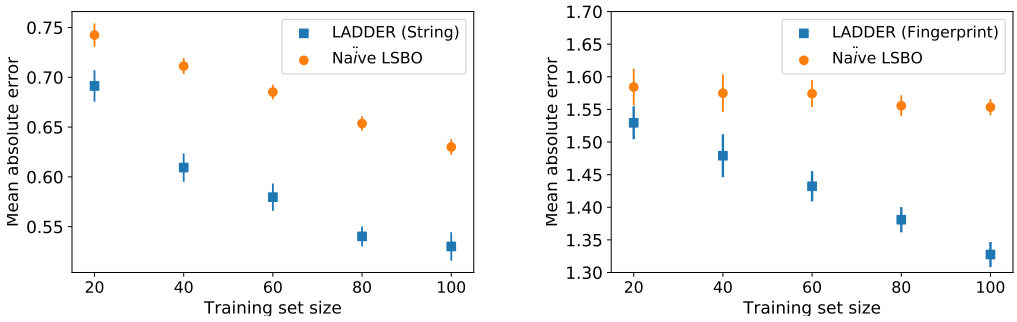
Evaluation metric. We evaluate all methods on the best objective (log MSE for the arithmetic expressions task and logP for the chemical design task) value uncovered as a function of the number of experiments (expensive function evaluations). The method that finds high-performing structures with less number of function evaluations is considered better.

We ran each method on all benchmarks for 10 different runs with the same initialization (small number of randomly selected structures and function evaluations). We plot the mean and two times the standard error for all our experimental results.

6.3.3 Results and Discussion

Comparison of surrogate models. Recall that the key hypothesis of this chapter is that surrogate model employed by the Naïve LSBO approach (GP model with Matern

[4]https://github.com/CMA-ES/pycma



(a) Arithmetic expression optimization **(b)** Chemical design optimization

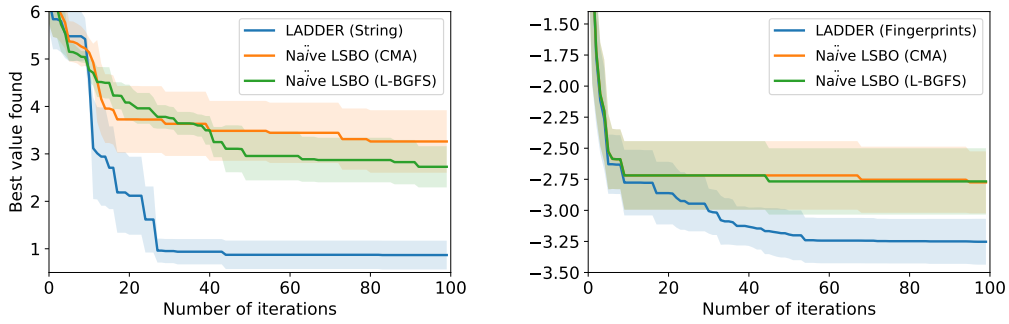
Figure 6.2: Mean absolute error results comparing the quality of model fit for varying sizes of training sets with two models: GP model with Matern kernel (Naïve LSBO) and GP model with the proposed structure-coupled kernel (LADDER). Lower MAE values mean better surrogate model.

kernel in our case) is ineffective and GP model with our proposed structure-coupled kernel for small-data settings. To test this hypothesis, we compare the quality of the model fit for these two GP models (Naïve LSBO and LADDER) by evaluating the mean absolute error (MAE) of their predictions on a testing set. To perform this experiment, we generate 50 (uniformly) random training sets of different sizes and evaluate the models on 20 random testing sets. The averaged MAE results over these 1000 training and testing set pairs are shown in Figure 6.2. We make following observations. **1)** The standard GP model on latent space denoted by Naïve LSBO shows some improvement in MAE for the arithmetic expressions task while the improvement is minimal for the chemical design task. **2)** Our proposed surrogate model, i.e., GP with structure-coupled kernel, denoted by LADDER has significantly lower MAE than Naïve LSBO on both the benchmarks and continuously decreases as

a function of the training set size. These strong results corroborate our key hypothesis and demonstrate the effectiveness of our proposed surrogate model with the structure-coupled kernel.

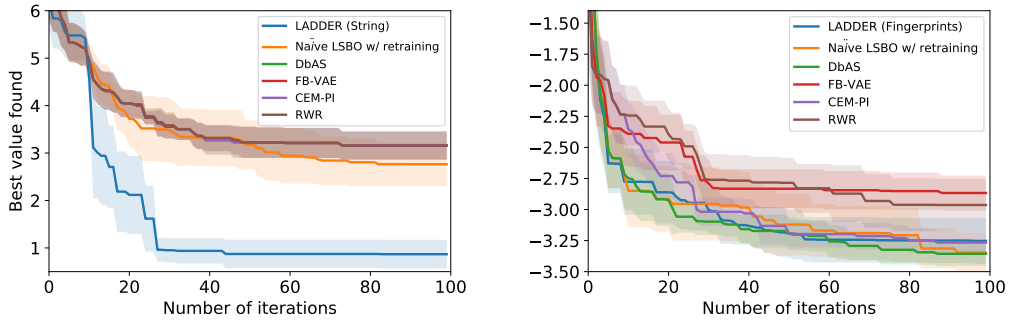
Naïve latent space BO vs. LADDER. A natural question based on the above results is whether improved surrogate modeling results in better overall BO performance. To answer this question, we compare the BO performance (best uncovered or incumbent objective value vs. number of function evaluations or iterations) of Naïve LSBO and LADDER. We consider two choices for acquisition function optimizer within the Naïve LSBO approach: zeroth order CMA based optimizer (same as LADDER) and second-order gradient based optimizer (L-BFGS). We make the following observations from the results shown in Figure 6.3. **1)** LADDER consistently uncovers significantly better structures than those from the Naïve LSBO approach on both benchmarks. This is a direct consequence of the better surrogate model of LADDER since the acquisition function optimizer is kept the same for both methods, i.e., CMA. **2)** L-BFGS based acquisition function optimizer slightly improves the BO performance of Naïve LSBO but still cannot match the performance of LADDER.

LADDER vs. State-of-the-art. To further analyze LADDER, we compare its performance with a set of state-of-the-art methods. We include *weighted retraining* approach which was recently proposed [210] to update the latent space model as the BO algorithm progresses (Naïve LSBO w/ retraining). We also consider design



(a) Arithmetic expression dataset **(b)** Chemical design dataset

Figure 6.3: Results comparing the BO performance of LADDER and Naïve latent space BO.



(a) Arithmetic expression dataset **(b)** Chemical design dataset

Figure 6.4: Results comparing the BO performance of LADDER and different state-of-the-art methods.

by adaptive sampling (DbAS) [35], the cross-entropy method with probability of improvement (CEM-PI) [185], the feedback VAE (FB-VAE) [94], and reward-weighted regression (RWR) [177]. We employ the publicly available implementation of these baseline methods. We make the following observations from the results shown in Figure 6.4. **1)** LADDER performs significantly better on the arithmetic expressions task and similar to the best method on the chemical design task. We verify the

similar performance of LADDER on the chemical design task by performing a two-sided paired Wilcoxon test at 1% significance. The performance of LADDER and Naïve LSBO w/ retraining is statistically similar on the chemical design task (p-value = 0.0489). **2)** Retraining the latent space model helps in improving the performance of Naïve LSBO on both tasks. **3)** DbAS and CEM-PI perform similar to Naïve LSBO w/ retraining since they are special case of retraining method as described in [210]. The discrepancy in LADDER's performance on the chemical design benchmark can be attributed to the low-flexibility of fingerprint kernel which doesn't include any hyper-parameters (as opposed to string kernel) to tune it for a specific dataset.

6.4 Summary

We introduced a sample-efficient Bayesian optimization (BO) approach for combinatorial spaces called LADDER. The key idea behind LADDER is a Gaussian process based surrogate model that combines the complementary strengths of latent space representation with rich information about decoded outputs using structured kernels. We showed that the BO performance of LADDER is better or similar than state-of-the-art methods and significantly better than the Naïve latent space BO method. Since LADDER's key contribution is in the surrogate model part of the BO procedure, it provides the flexibility to use any acquisition function, which opens up an avenue for various type of extensions including multi-objective [20, 53, 176, 204], multi-fidelity [206, 125, 226], and constrained BO [100, 86].

CHAPTER SEVEN

MERCER FEATURES TO SAMPLE FUNCTIONS FROM GAUSSIAN PROCESS POSTERIOR

This chapter addresses the BO problem setting for fixed-size combinatorial spaces (e.g., sequences and graphs) similar to Chapters 3 and 4. The key challenge in this problem setting is to balance the complexity of statistical models and tractability of search to select combinatorial structures for evaluation. In this chapter, we propose an efficient approach referred as *Mercer Features for Combinatorial Bayesian Optimization (MerCBO)*. The key idea behind MerCBO is to provide explicit feature maps for diffusion kernels over discrete objects by exploiting the structure of their combinatorial graph representation. These Mercer features combined with Thompson sampling as the acquisition function allows us to employ tractable solvers to find next structures for evaluation. Mercer features allow sampling functions from Gaussian process posteriors which is a key step that enables the use of information-theoretic acquisition functions [21, 23, 26, 22, 28, 2, 27, 25]. Experiments on diverse real-world benchmarks demonstrate that MerCBO performs similarly or better than prior methods.

7.1 MerCBO Algorithm

Algorithm 7 MerCBO Algorithm

Require: Input space $\mathcal{X}$, Black-box objective f, Order of Mercer features MAX, Query budget B

Initialize an empty list $\phi = []$

for $i = 1$ to MAX **do**

  Compute the features for the ith order: $\{\sqrt{e^{-\beta\lambda_i}} \cdot (-1)^{\langle r_i, \mathbf{x}\rangle}\}$ and append to ϕ

end for

Return Mercer features ϕ

Initialize a small-sized training set TRAIN with Mercer features computed for input structures

while Query budget does not exceed B **do**

  Learn Gaussian Process model M using TRAIN

  Construct Thompson sampling acquisition function by sampling from a parametric approximation of the GP posterior

  Find $\mathbf{x}_{next}$ by optimizing Thompson sampling based acquisition function over model M

  Construct submodular relaxation based lower bound of AFO with parameters

  while convergence or maximum iterations **do**

    Step 2: Solve the relaxed problem using graph cut algorithms

    Step 3: Update the relaxation parameters to obtain a tighter bound

  end while

  Evaluate f at $\mathbf{x}_{next}$ and add to TRAIN: $\mathbf{x}_{next}$, Mercer_Features($\mathbf{x}_{next}$, MAX), and $O(\mathbf{x}_{next})$

end while

Return Best input $\mathbf{x}_{best}$ and its objective value $O(\mathbf{x}_{best})$

We first provide an overview of MerCBO and its advantages.

Overview of MerCBO algorithm. MerCBO is an instantiation of the generic BO framework. 1) Gaussian Process with diffusion kernels is the surrogate statistical model. 2) Thompson sampling is used as the acquisition function α. The proposed Mercer features are used to sample from a parametric approximation of the GP posterior to construct the Thompson sampling objective. 3) Acquisition function optimization problem is shown to be a Binary Quadratic Program which is solved using an efficient and scalable submodular-relaxation method. The key idea is to construct a lower bound of the AFO problem in terms of some unknown relaxation parameters and iteratively optimize those parameters to obtain a tighter bound. Algorithm 7 shows the complete pseduo-code of MerCBO. We use a small set of input structures and their function evaluations (denoted TRAIN) to bootstrap the surrogate model. In each iteration, we select the next structure for evaluation $\mathbf{x}_{next}$ by solving the AFO problem; add $\mathbf{x}_{next}$, its function evaluation $O(x_{next})$, and mercer features of $\mathbf{x}_{next}$ to TRAIN. We repeat these sequence of steps until the query budget is exhausted and then return the best found structure $\mathbf{x}_{best}$ as the output.

Advantages of MerCBO over COMBO and SMAC. **1)** Mercer features allow us to leverage a large number of acquisition functions from the continuous BO literature including Thompson sampling (TS), PES [99] and MES [216] to improve the BO performance for combinatorial spaces. **2)** Retains the modeling strength of complex GP-based model using diffusion kernels and still allows a tractable and scalable

AFO procedure with Thompson sampling. **3)** Mercer features combined with TS has many desiderata required for several scientific applications: *diversity* in explored structures and selection of *large batch* of structures for parallel evaluation. Indeed, our experiments on biological sequence design demonstrate the effectiveness of TS (embarrassingly parallel) over COMBO with EI.

7.1.1 Preliminaries

Graph Laplacian. Given a graph $G = (V, E)$, its Laplacian matrix $L(G)$ is defined as $D - A$, where D is the degree matrix and A is the adjacency matrix of G.

Graph Cartesian Product. Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their graph Cartesian product $(G_1 \square G_2)$ is another graph with vertex set $V(G_1 \square G_2) = V_1 \times V_2$ consisting of the set Cartesian product of V_1 and V_2. Two vertices (v_1, u_1) and (v_2, u_2) of $G_1 \square G_2$ (where $\{v_1, v_2\} \in V_1$ and $\{u_1, u_2\} \in V_2$) are connected if either, $v_1 = v_2$ and $(u_1, u_2) \in E_2$, or $u_1 = u_2$ and $v_1, v_2 \in E_2$.

Combinatorial Graph Representation of Discrete Space. Recall that we need a graph representation of the discrete space $\mathcal{X}$ to employ diffusion kernels for learning GP models. We consider the combinatorial graph representation (say $G{=}(V, E)$) proposed in a recent work [168]. There is one vertex for each candidate assignment of n discrete variables $x_1, x_2, \cdots, x_n$. There is an edge between two vertices if and only if the Hamming distance between the corresponding binary structures is one. This

graph representation was shown to be effective in building GP models for BO over discrete spaces [168].

7.1.2 Efficient Mercer features for Diffusion Kernel

We are interested in computing explicit feature maps for diffusion kernel over discrete spaces [135, 168], which is defined using the above combinatorial graph representation as follows:

$$K(V, V) = U \exp(-\beta\Lambda)U^T \tag{7.1}$$

where $U = [u_0, u_1, \cdots, u_{2^n-1}]$ is the eigenvector and $\Lambda = [\lambda_0, \lambda_1, \cdots, \lambda_{2^n-1}]$ is the eigenvalue matrix of the graph Laplacian $L(G)$ and β is a hyper-parameter. For any two given structures $\mathbf{x}_1, \mathbf{x}_2 \in \{0, 1\}^n$, the kernel definition is:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} u_i([\mathbf{x}_1]) u_i([\mathbf{x}_2]) \tag{7.2}$$

where $u_i([\mathbf{x}_1])$ denotes the value of the ith eigenvector indexed at the integer value represented by $\mathbf{x}_1$ in base-2 number system. From the above equation, we can see that the eigenspace of the graph Laplacian $L(G)$ is the central object of study for diffusion kernels. The key insight behind our proposed approach is to exploit the structure of the combinatorial graph G and its Laplacian $L(G)$ for computing its eigenspace in a

closed-form.

Since G is an exponential-sized graph with 2^n vertices, computing the eigenspace of $L(G)$ seems intractable at first sight. However, G has special structure: G has an equivalent representation in terms of the graph Cartesian product of n sub-graphs $G_1, G_2, \cdots, G_n$:

$$G = (((G_1 \square G_2) \square G_3 \cdots) \cdots) \square G_n) \tag{7.3}$$

where each sub-graph G_i represents the ith binary input variable and is defined as a graph with two vertices (labeled 0 and 1) and an edge between them. Therefore, $L(G)$ is equivalently given by:

$$L(G) = L((((G_1 \square G_2) \square G_3 \cdots) \cdots) \square G_n)) \tag{7.4}$$

$$L(G) = L(G_1) \oplus L(G_2) \oplus L(G_3) \cdots \oplus L(G_n) \tag{7.5}$$

where Equation 7.5 is due to distributive property of the Laplacian operator over graph Cartesian product via Kronecker sum operator ($\oplus$) [96] and associative property of the $\oplus$ operator.

Proposition 11. *[96] Given two graphs G_1 and G_2 with the eigenspace of their Laplacians being $\{\Lambda^1, U^1\}$ and $\{\Lambda^2, U^2\}$ respectively, the eigenspace of $L(G_1 \square G_2)$ is given by $\{\Lambda^1 \bowtie \Lambda^2, U^1 \otimes U^2\}$ where $\Lambda^1 \bowtie \Lambda^2 = \{\lambda_i^1 + \lambda_j^2 : \lambda_i^1 \in \Lambda^1, \lambda_i^2 \in \Lambda^2\}$ and $\otimes$ is the*

Kronecker product operation.

Property 11 gives a recursive procedure to compute the eigenspace of $L(G)$ based on its decomposition in Equation 7.5 provided the eigenspace of each of its constituent $L(G_i)$ is easily computable. Fortunately, in our special case, where each G_i is a simple graph of two vertices (labeled 0 and 1) with an edge between them, it has two eigenvalues $\{0, 2\}$ with corresponding eigenvectors $[1, 1]$ and $[1, -1]$ respectively. The eigenvector matrix $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is called as a Hadamard matrix of order 2 (2^1, where 1 in the exponent is the number of input variables) [81, 221]. Applying the $\bowtie$ and $\otimes$ operation recursively for n inputs (represented by $\{G_i : i \in \{1, 2, \cdots, n\}\}$) as described in property 11, it can be seen that the eigenspace of L(G) has an explicit form given as:

1. $L(G)$ has n distinct eigenvalues $\{0, 2, 4, \cdots, 2n\}$ where jth eigenvalue occurs with multiplicity $\binom{n}{j}, j \in \{0, 1, 2, \cdots n\}$.

2. The eigenvectors of $L(G)$ are given by the columns of Hadamard matrix of order 2^n.

Hadamard matrix of order 2^n is equivalently defined as:

$$H_{ij} = (-1)^{<r_i, r_j>} \tag{7.6}$$

where r_i is the n-bit representation of the integer i in base-2 system. Using this definition of Hadamard matrix, each entry of the eigenvectors of $L(G)$ can be computed in a closed form. Therefore, from Equation 7.2, the kernel value for any pair of structures $\mathbf{x}_1$ and $\mathbf{x}_2$, $K(\mathbf{x}_1, \mathbf{x}_2)$, can be written in terms of an equivalent sum over binary vectors:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} \cdot -1^{<r_i, \mathbf{x}_1+\mathbf{x}_2>} \tag{7.7}$$

where r_i is the base-2 representation of integer i ranging from 0 to 2^n-1. We rearrange the RHS of Equation 7.7 to delineate the dependency on each input in the pair.

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=0}^{2^n-1} e^{-\beta\lambda_i} \cdot -1^{<r_i, \mathbf{x}_1>} \cdot -1^{<r_i, \mathbf{x}_2>} \tag{7.8}$$

$$K(\mathbf{x}_1, \mathbf{x}_2) =< \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) > \tag{7.9}$$

where $\phi(\mathbf{x})$ corresponds to the proposed (explicit) *Mercer feature maps* of any input structure $\mathbf{x}$ and is given as follows:

$$\phi(\mathbf{x}) = \{i \in [0, 2^n - 1] : \sqrt{e^{-\beta\lambda_i}} \cdot -1^{<r_i, \mathbf{x}>}\} \tag{7.10}$$

As mentioned earlier, the eigenvalues of $L(G)$ are explicitly given by the set $\{2j : j \in \{0, 1, 2, \cdots n\}\}$, where jth eigenvalue occurs with multiplicity $\binom{n}{j}$. Based on this

observation, an elegant way of interpreting the feature maps given in 7.10 is based on the number of 1s in the r_i vector (binary expansion of integer i). There are exactly $\binom{n}{j}$ r-vectors with j bits set to 1. Hence, we refer to j as the *order of the Mercer feature maps*. The order variable controls the trade-off between the computational cost and approximation quality of the feature map. We found that the second-order feature maps[1] maintain the right balance as they can be computed efficiently and also allows to perform a tractable and scalable search for acquisition function optimization as described in the next section. Moreover, choosing second order is also prudent from the viewpoint of the definition of diffusion kernels itself, which requires suppressing higher frequency elements of the eigenspace [168] to define a smooth function over discrete spaces.

7.1.3 Tractable Acquisition Function Optimization

In this section, we describe a tractable and scalable acquisition function optimization procedure using the proposed Mercer features and Thompson sampling. Thompson sampling [47, 101, 126] selects the next point for evaluation by maximizing a sampled function from the GP posterior. We approximate the non-parametric GP model using a parametric Bayesian linear model $f(\mathbf{X}) = \theta^T \phi(\mathbf{x})$ using our proposed Mercer features $\phi(\mathbf{x})$. Given a Gaussian prior, i.e., $\theta \sim \mathcal{N}(0, I)$, the posterior

[1]Slight abuse of notation. Second-order, from here-on, means concatenation of both first and second-order features.

distribution over θ is also a Gaussian with the following mean and covariance:

$$\mu = (\mathbf{\Phi^T \Phi} + \sigma^{\mathbf{2}}\mathbf{I})^{-\mathbf{1}}\mathbf{\Phi^T y} \tag{7.11}$$

$$\mathbf{\Sigma} = (\mathbf{\Phi^T \Phi} + \sigma^{\mathbf{2}}\mathbf{I})^{-\mathbf{1}}\sigma^2 \tag{7.12}$$

with $\mathbf{\Phi}$ is the feature matrix with ith row corresponding to $\phi(\mathbf{x}_i)$ and $\mathbf{y}$ is the output vector with $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$.

Acquisition function optimization problem. We sample θ^* from the posterior parametrized by μ and $\mathbf{\Sigma}$ defined in Equation 7.12 and minimize the objective $f(\mathbf{x})$ $= \theta^*\phi(\mathbf{x})$ with respect to $\mathbf{x} \in \{0,1\}^n$. Suppose $\mathbf{x}$=$(x_1, x_2, \cdots, x_n)$ is a candidate structure with values assigned to n binary variables. We now show how this AFO problem is an instance of Binary quadratic programming (BQP) problem by using second-order feature maps from 7.10.

$$\phi(\mathbf{x}) = \left\{ 0 \leq i \leq \binom{n}{2} : \sqrt{e^{-\beta\lambda_i}} \cdot -1^{<r_i, \mathbf{x}>} \right\} \tag{7.13}$$

The second-order feature maps are composed of two major parts: features constructed by order-1 r-vectors and features constructed by order-2 r-vectors. For order-1 r-vectors, the set $\{0 \leq i \leq n : -1^{<r_i, \mathbf{x}>}\}$ is equivalent to $\{0 \leq i \leq n : -1^{x_i}\}$. Similarly, the order-2 feature maps can also be written as $\{1 \leq i \leq n, i+1 \leq j \leq n : -1^{(x_i + x_j)}\}$. Therefore, ignoring the constant corresponding to the 0th index of θ^*,

the AFO problem $\min_{\mathbf{x}\in\{0,1\}^n} \theta^*\phi(\mathbf{x})$ becomes:

$$\begin{aligned}\min_{\mathbf{x}\in\{0,1\}^n} &\sum_{i=1}^{n} \theta_i^* \sqrt{e^{-\beta\lambda_i}} \cdot -1^{x_i} \\ &+ \sum_{i=1}^{n}\sum_{j=i+1}^{n} \theta_{ij}^* \sqrt{e^{-\beta\lambda_{n\cdot i+j}}} \cdot -1^{(x_i+x_j)}\end{aligned} \tag{7.14}$$

By replacing the -1^{x_i} terms in 7.14 with an equivalent term $(1-2x_i)$, we get:

$$\begin{aligned}\min_{\mathbf{x}\in\{0,1\}^n} &\sum_{i=1}^{n} \theta_i^* \sqrt{e^{-\beta\lambda_i}}(1-2x_i) \\ &+ \sum_{i=1}^{n}\sum_{j=i+1}^{n} \theta_{ij}^* \sqrt{e^{-\beta\lambda_{n\cdot i+j}}}(1-2x_i)(1-2x_j)\end{aligned} \tag{7.15}$$

Rearranging and combining terms with the same degree, we get the following final expression for AFO which is clearly a Binary quadratic programming (BQP) problem:

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{b}^T\mathbf{x} + x^T\mathbf{A}\mathbf{x} \tag{7.16}$$

where $\mathbf{b}$ and $\mathbf{A}$ are defined as given below:

$$b_i = -2\left(\theta_i^*\sqrt{e^{-\beta\lambda_i}} + \sum_{j=1}^{n}\theta_{ij}^*\sqrt{e^{-\beta\lambda_{n\cdot i+j}}}\right), 1 \le i \le n \tag{7.17}$$

$$A_{ij} = \delta_{ij}\cdot 4\left(\theta_{ij}^*\sqrt{e^{-\beta\lambda_{n\cdot i+j}}}\right), 1 \le i,j \le n \tag{7.18}$$

where $\delta_{ij} = 1$ if $j > i$ and 0 otherwise.

Efficient submodular-relaxation solver. BQP is a well-studied problem in multiple areas including computer vision [92]. Motivated by the prior success of submodular-relaxation methods in the structured prediction area [208], we study a fast and scalable approach for solving AFO problems based on recent advances in submodular-relaxation [107]. The key idea is to construct an efficient relaxed problem with some unknown parameters and optimize those parameters iteratively to improve the accuracy of solutions. We provide a high-level sketch of the overall algorithm [60] below.

The objective in 7.16 is called as *submodular* if $A_{ij} \leq 0 \quad \forall i, j$. Submodular functions can be exactly minimized by fast strongly polynomial-time graph-cut algorithms [82]. However, in general, the objective might not be submodular. Therefore, a submodular relaxation to the objective in Equation 7.16 [107] is constructed by lower bounding the positive terms A^+ of A:

$$\begin{aligned} \mathbf{x}^T(\mathbf{A}^+ \circ \Gamma)\mathbf{1} + \mathbf{1}^T(\mathbf{A}^+ \circ \Gamma)\mathbf{x} - \mathbf{1}^T(\mathbf{A}^+ \circ \Gamma)\mathbf{1} \\ \leq \mathbf{x}^T\mathbf{A}^+\mathbf{x} \end{aligned} \tag{7.19}$$

where $\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-$,

$$\mathbf{A}^+ = A_{ij} \text{ if} A_{ij} \geq 0 \text{ and } 0 \text{ otherwise} \tag{7.20}$$

$$\mathbf{A}^- = A_{ij} \text{ if} A_{ij} \leq 0 \text{ and } 0 \text{ otherwise} \tag{7.21}$$

and Γ stands for unknown relaxation parameters. The accuracy of this relaxed problem critically depends on Γ. Therefore, we perform optimization over Γ parameters after initializing them by repeating the following two steps.

1. Solve submodular relaxation lower bound of the BQP objective in 7.16 using minimum graph-cut algorithms.

2. Update relaxation parameters Γ via gradient descent

This submodular-relaxation based AFO solver scales gracefully with the increasing input dimension n because of the strongly polynomial complexity $O(n^3)$ of minimum graph cut algorithms [4].

7.2 Experiments and Results

Experimental setting. The source code for state-of-the-art baseline methods was taken from their github links: COMBO (`https://github.com/QUVA-Lab/COMBO`), BOCS (`https://github.com/baptistar/BOCS`), and SMAC (`https://github.com/automl/SMAC3`). For a fair comparison, the priors for all GP hyper-parameters and their posterior computation were kept the same for both COMBO and MerCBO. COMBO employs a separate hyper-parameter β_i for each dimension to enforce sparsity, which is important in certain applications. For MerCBO also, we include sparsity in sampling θ for Thompson Sampling for all benchmarks other than Ising: $\mathcal{N}\left((\mathbf{\Phi^T\Phi} + \sigma^2\mathbf{\Upsilon^{-1}})^{-1}\mathbf{\Phi^T y}, (\mathbf{\Phi^T\Phi} + \sigma^2\mathbf{\Upsilon^{-1}})^{-1}\sigma^2\right)$. It should be noted that we are

not introducing any more hyper-parameters than COMBO, but use them in a *strong heirarchical* sense [31].

We ran five iterations of submodular relaxation approach for solving AFO problems and observed convergence. We ran each method on all the benchmarks for 25 random runs and report mean and two times the standard error for results.

7.2.1 Sequential Design Optimization Benchmarks

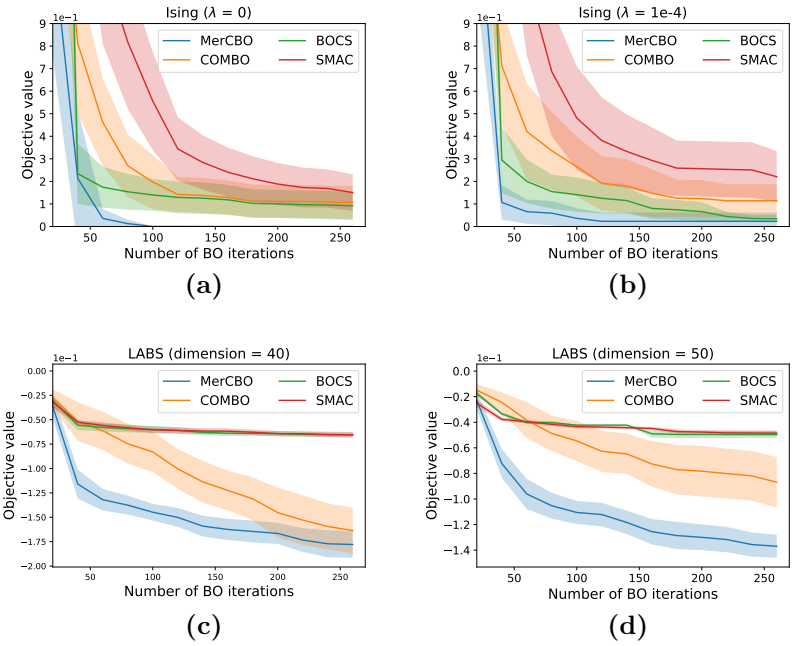


Figure 7.1: Results for Ising and LABS domains.

We employed four diverse benchmarks for sequential design: *function evaluations are performed sequentially.*

Ising sparsification. The probability distribution $p(z)$ [16, 168] defined by Zero-field

Ising model I_p is parametrized by a symmetric interaction matrix J^p whose support is represented as a graph G^p. The goal in this problem is to approximate $p(z)$ with another distribution $q(z)$ such that the number of edges in G^q are minimized. The objective function is defined as:

$$\min_{\mathbf{x}\in\{0,1\}^n} D_{KL}(p||q) + \lambda\|\mathbf{x}\|_1$$

where D_{KL} is the KL-divergence between p and q, and λ is a regularization parameter. The results for this 24-dimensional domain are shown in Figure 7.1a and 7.1b. In COMBO [168], BOCS was shown to perform better than COMBO on this domain. However, MerCBO shows the best performance among all methods signifying that the proposed approach augments the performance of GP surrogate model.

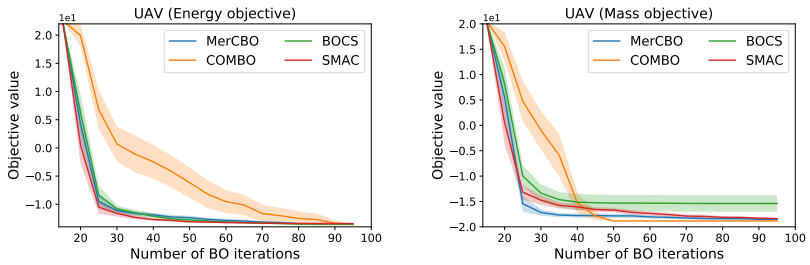


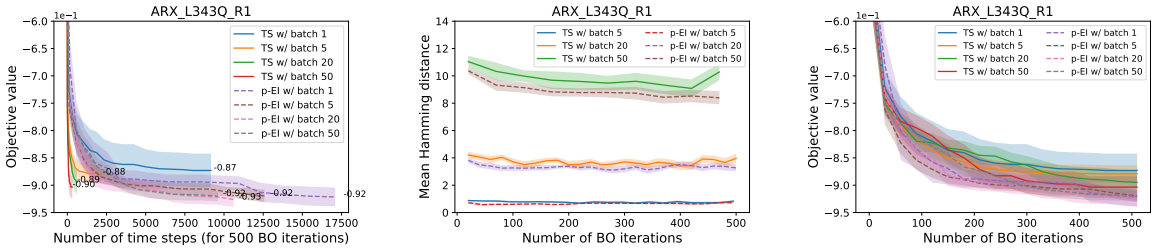
Figure 7.2: Results for power system design of UAVs.

Low auto-correlation binary sequences (LABS). This problem has diverse applications in multiple fields [30, 172] including communications where it is used in high-precision interplanetary radar measurements [193]. The goal is to find a binary

sequence $\{1, -1\}$ of length n that maximizes the *Merit factor (MF)* defined as follows:

$$\max_{\mathbf{x}\in\{1,-1\}^n} \mathrm{MF}(\mathbf{x}) = \frac{n^2}{E(\mathbf{x})}\,,$$
$$E(\mathbf{x}) = \sum_{k=1}^{n-1}\left(\sum_{i=1}^{n-k} x_i x_{i+k}\right)^2$$

where $E(\mathbf{x})$ is the energy of the sequence. This domain allows us to evaluate all methods on large input dimensions. Results with 40 and 50 dimensions are shown in Figure 7.1c and 7.1d respectively. MerCBO shows best performance among all methods showing its effectiveness. COMBO shows poor convergence and matches MerCBO only on 40 dimensional case at the end of allocated budget. SMAC and BOCS show poor performance resulting from their inability to model the underlying structure properly.



(a) **(b)** **(c)**

Figure 7.3: Representative results on biological sequence design problem for one transcription factor.

Electrified aviation power system design. We consider the design of power system for unmanned aerial vehicles (UAVs) via a time-based static simulator [209, 25].

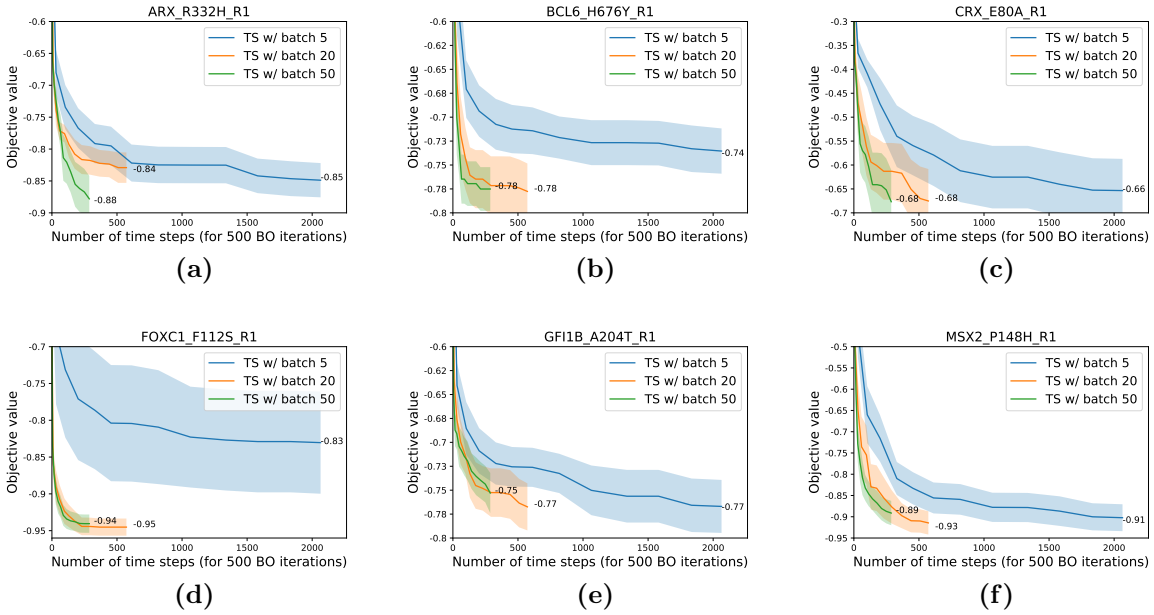


Figure 7.4: Results on biological sequence design with Thompson sampling for six transcription factors.

Each structure is specified using five design variables such as the battery pack configuration and motor size. Evaluation of each design requires performing a computationally-expensive simulation. We consider two variants of this design task: (1) Optimize *total energy*; and (2) Optimize *mass*. We employed a dataset of 250,000 candidate designs for our experiments. The categorical variables are encoded as 16 bit binary variables.

To make the benchmark challenging, we initialized the surrogate models by randomly selecting from worst (in terms of objective) 10% structures. Results for this benchmark are shown in Figure 7.2. Interestingly, both BOCS and SMAC shows good performance on both mass and total energy objectives of this benchmark. Since EI has the tendency to be relatively more exploitative, COMBO shows poor convergence

but reaches the best value at the end of allocated budget. MerCBO converges much faster on both problems, but the performance plateaus out on the mass objective. We attribute this behavior to the naturally exploratory behavior of the Thompson sampling acquisition function.

7.2.2 Parallel Biological Sequence Design

Motivation. Design of optimized biological sequences such as DNA and proteins is a fundamental problem with many important applications [223, 5, 222]. These design problems have the following requirements: uncover a diverse set of structures (*diversity*); select a large batch of structures in each round to perform parallel evaluations (*large-scale parallel experiments*); and use parallel experimental resources to accelerate optimization (*real-time accelerated design*).

Benefits of Thompson sampling (TS). TS is a powerful approach that meets these requirements [101]. Any acquisition function is defined as the expectation of a utility function under the posterior predictive distribution $p(y|\mathbf{x}, D) = \int p(y|\theta)p(\theta|\mathbf{x}, D)$. TS approximates this posterior by a single sample $\theta^* \sim p(\theta|\mathbf{x}, D)$ which inherently enforces exploration due to the variance of this Monte-Carlo approximation. In MerCBO, we can sample as many θ (via Mercer features) as required and can employ scalable AFO solvers for each sample in parallel. However, parallel version of EI (acquisition function employed in COMBO) does not meet most of the requirements as

it selects the batch of structures for evaluation sequentially.

Experimental setup. We evaluate TS and EI with GP based models on diverse DNA sequence design problems. The goal is to find sequences that maximize the binding activity between a variety of human transcription factors and every possible length-8 DNA sequence [18, 5]. Categorical variables are encoded as 2 bit binary variables. We multiply objective values by -1 to convert the problem into minimization for consistency. We compare the parallel version of EI as proposed in [195] and used in multiple earlier works [101, 126] with parallel TS. For a batch of B evaluations in each iteration, parallel-EI works by picking the first input in the same way as in the sequential setting and selects the remaining inputs $j = 2$ to B by maximizing the expected EI acquisition function under all possible outcomes of the first $j - 1$ pending evaluations (aka fantasies [195]). On the other hand, TS is easily scalable and parallelizable by sampling B θ's from the GP posterior and optimizing each of them independently with our MerCBO approach. We run both TS and EI experiments on a 32 core Intel(R) Core(TM) i9-7960X CPU @ 2.80GHz machine. All reported time units are in seconds on the same machine.

Discussion of results. Figure 7.3 shows the canonical comparison of parallel TS with parallel EI (p-EI) on one transcription factor from the DNA-binding affinity benchmark. The numbers within the plots show the mean objective value for a budget of 500 evaluations. Although parallel-EI is slightly better in terms of optimization

performance, TS is extremely fast and is useful for practitioners in time-constrained applications including drug and vaccine design. The diversity of best batch of sequences is equally important to hedge against the possibility of some chosen candidate designs failing on downstream objectives [5]. Figure 7.3b shows the results comparing the diversity of sequences (on mean Hamming distance metric) found by TS versus parallel-EI. There are two key observations that can be made from this figure. First, increasing the batch size increases the diversity of sequences. Second, TS finds comparatively more diverse sequences than parallel-EI.

Interestingly, performance of parallel TS improves with increasing batch size. To justify this observation, we further evaluated parallel TS on six other transcription factors as shown in Figure 7.4. As evident from the figure, performance of parallel TS improves with increasing batch size on five out of six benchmarks. This shows that the exploratory behavior of TS, which can be bad in some sequential settings, helps in better performance for the parallel setting.

7.3 Summary

We introduced an efficient approach called MerCBO for optimizing expensive black-box functions over discrete spaces. MerCBO is based on computing Mercer features for diffusion kernels and fast submodular relaxation based acquisition function optimization. We showed that MerCBO produces similar or better performance than state-of-the-art methods on multiple real-world problems.

CHAPTER EIGHT

L2S-DISCO: A GENERIC LEARNING-TO-SEARCH FRAMEWORK FOR ACQUISITION FUNCTION OPTIMIZATION

This chapter addresses the BO problem setting for fixed-size combinatorial spaces (e.g., sequences and graphs) similar to Chapters 3, 4 and 7. The key challenge is to select a sequence of combinatorial structures to evaluate, in order to identify high-performing structures as quickly as possible. Our main contribution is to introduce and evaluate a new learning-to-search framework for this problem called L2S-DISCO. The key insight is to employ search procedures guided by control knowledge at each step to select the next structure and to improve the control knowledge as new function evaluations are observed. This framework is inspired by the prior success of integrating learning and search for solving structured prediction problems [71, 72, 138, 73, 160, 46]. We provide a concrete instantiation of L2S-DISCO for local search procedure and empirically evaluate it on diverse real-world benchmarks. Results show the efficacy of L2S-DISCO over state-of-the-art algorithms in solving complex optimization problems.

8.1 Learning to Search Framework

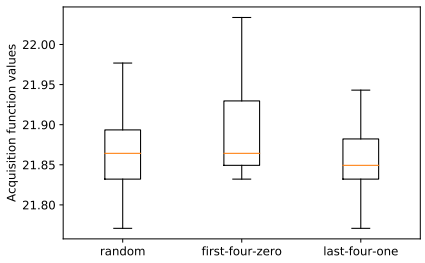
In this section, we first motivate learning-to-search (L2S) methods for solving acquisition function optimization (AFO) problems. Subsequently, we describe our

proposed learning to search framework, L2S-DISCO, and provide a concrete instantiation for local-search based AFO.

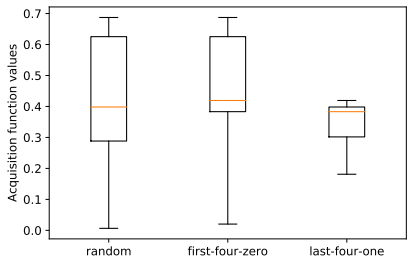
8.1.1 Motivation

Search-based AFO solvers. In a search-based optimizer, the overall problem-solving can be modeled as a computational search process defined in terms of an appropriate *search space* over candidate solutions, *search procedure* to uncover solutions, and *search control knowledge* to guide the search. For example, a solver, based on local search with multiple restarts, may use control knowledge that biases the restart distribution. Similarly, a solver, based on branch-and-bound search, may use control knowledge corresponding to policies for node expansion and pruning based on the current state of the solver. An important aspect of search-based optimization is that we can potentially improve the search control knowledge during a search by feeding the information about the search progress to machine learning techniques.

Relation between AFO problems. We now give the intuition for why it may be useful to learn control knowledge across the sequence of AFO problems encountered during BO. Recall that the change in acquisition function $\mathcal{AF}(\mathcal{M}, x)$ from iteration i to $i+1$ is due to *only one* new training example (x_i, y_i), where x_i is the selected structure in iteration i and y_i is its function evaluation. Intuitively, even if the acquisition function scores of candidate structures in $\mathcal{X}$ are changing, the search



(a) Contamination domain with UCB acquisition function.



(b) Ising domain with EI acquisition function.

Figure 8.1: Empirical evidence to show how learning can be useful to solve acquisition function optimization. Boxplot shows final acquisition function values resulting from 100 runs of local search based optimization with three different restart strategies.

control knowledge can still guide the search towards promising structures and only require small modifications to account for the slight change in the AFO problem from previous BO iteration. This motivates using machine learning to adapt the knowledge in a way that generalizes from prior iterations of AFO to future AFO iterations.

Empirical evidence for the utility of learning. We now provide some empirical evidence on real-world problems to show how machine learning can be potentially useful to improve the accuracy of solving AFO problems. We consider local search with multiple restarts as the AFO solver. In this case, the AFO solver takes as input the objective function $\mathcal{AF}(\mathcal{M}, x)$ and restarting strategy, and returns the local optima $\hat{x} \in \mathcal{X}$ with associated acquisition function value $\mathcal{AF}(\mathcal{M}, \hat{x})$. The accuracy of local search based AFO solver critically depends on the restart strategy. We performed local search based AF optimization using three different restart strategies on

optimization problems with binary discrete variables: 1. Completely random (*random*); 2. Assigning the first four discrete variables as zero and remaining randomly (*first-four-zero*); and 3. Assigning the last four discrete variables as one and remaining randomly (*last-four-one*). In figure 8.1, we show the results of solving a single AFO problem using these three different restart strategies over 100 runs. We plot the distribution of $\mathcal{AF}(\mathcal{M}, \hat{x})$ for these strategies. We can see that different restart strategies give varied solutions (empirically). This observation can be leveraged to learn a search heuristic to select promising starting states for local search using the training data from local search trajectories.

8.1.2 L2S-DISCO and Key Elements

L2S-DISCO integrates machine learning techniques and combinatorial search in a principled manner for accurately solving AFO problems to select combinatorial structures for evaluation. This framework allows us to employ surrogate statistical models of arbitrary complexity and can work with any acquisition function. The key insight behind L2S-DISCO is to directly tune the search via learning during the optimization process to select the next structure for evaluation. The search-based perspective has several advantages: 1) High flexibility in defining search spaces over structures; 2) Easily handles domain constraints that determine which structures are "valid". For example, when designing an optimized network on the chip to facilitate data transfer between multiple cores, we need to make sure that there is a viable

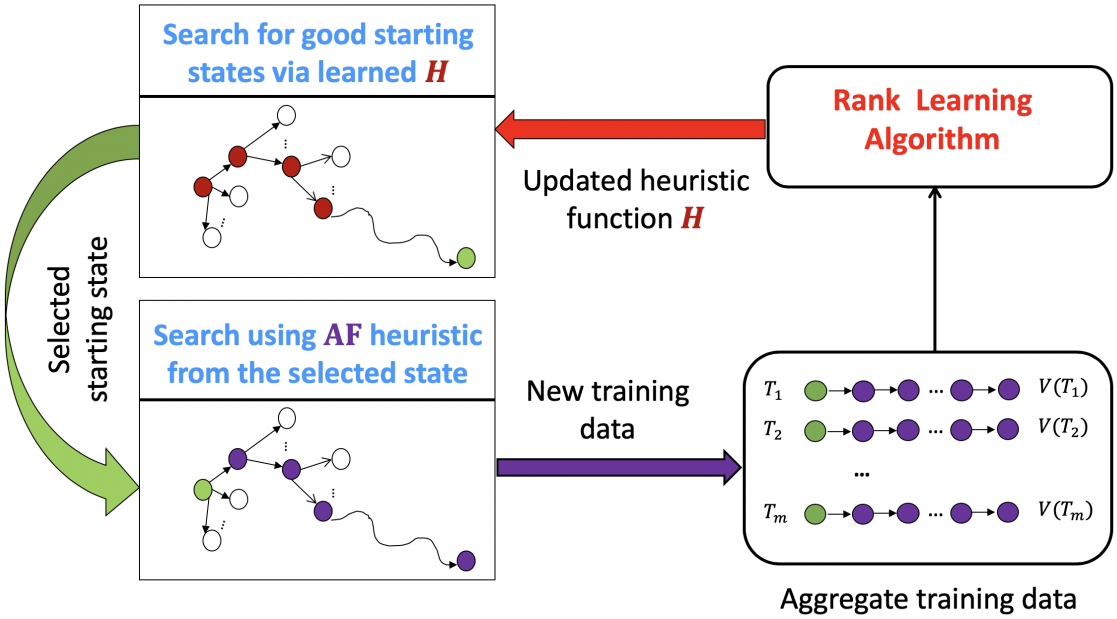


Figure 8.2: High-level overview of L2S-DISCO instantiation for local search. It repeatedly performs three steps. First, run local search from a random state guided by current heuristic $\mathcal{H}$ to select a good starting state. Second, run local search from this selected starting state guided by acquisition function ($\mathcal{AF}$). Third, use new training data in the form of local search trajectory T and acquisition function value of the local optima $V(T)$ to update the heuristic $\mathcal{H}$ via rank learning.

path between any pair of cores; 3) Allows to incorporate prior knowledge in the form of heuristic rules to explore promising regions of the search space; and 4) Provides additional points for learning within the search framework to improve the effectiveness of search in uncovering better structures.

Overview of L2S-DISCO. We build a surrogate model $\mathcal{M}$ using a small number of experiments and their outcomes to guide our search process to select the sequence of combinatorial structures to perform experiments. L2S-DISCO is parameterized by a

search space $\mathcal{S}$ over structures, a learned function $\mathcal{AF}(\mathcal{M}, x \in \mathcal{X})$ to score the utility of structures for evaluation, a search strategy $\mathcal{A}$ (e.g., local search), and a learned search control knowledge $\mathcal{H}$ to guide the search towards high-scoring structures. In each BO iteration, we perform the following two steps repeatedly until the maximum time-bound is exceeded or a termination criteria is met. **Step 1:** Execute search strategy $\mathcal{A}$ guided by the current search control knowledge to uncover promising structures. **Step 2:** Update the parameters of search control knowledge $\mathcal{H}$ using the online training data generated from the recent search experience. Fig 8.2 illustrates the instantiation of L2S-DISCO for local search. Each structure $x \in \mathcal{X}$ uncovered during the entire search is scored according to $\mathcal{AF}(\mathcal{M}, x)$ and we select the highest scoring structure x_{next} for function evaluation. We perform experiment using the selected structure x_{next} and observe the outcome $f(x_{next})$. The statistical model $\mathcal{M}$ is updated using the new training example $(x_{next}, f(x_{next}))$. We repeat the next iteration of BO via L2S-DISCO initialized with the current search control knowledge.

Key Elements. There are two key elements in L2S-DISCO that need to be specified to instantiate it for a given search procedure. **1)** The form of training data to learn search control knowledge $\mathcal{H}$; and **2)** The learning formulation and associate learning algorithm to update the parameters of search control knowledge $\mathcal{H}$ using online training data. These elements vary for different search procedures and forms of search control knowledge. We provide a high-level example to illustrate these elements for

branch-and-bound search.

Branch-and-bound search is a widely used search procedure to solve combinatorial optimization problems. It employs a search space over partial structures, where each state corresponds to partial assignment of variables. The states with complete assignment for all variables are referred as terminals. Variable selection strategy for successive assignment is one of the main components of branch-and-bound search. Therefore, $\mathcal{H}$ corresponds to the policy that selects the variable on which to branch on for the next assignment. In this case, the training data is generated by the trajectories obtained by a strong branching (SB) strategy [129] which exhaustively tests each variable for assignment. A learning-to-rank formulation is natural for inducing the variable selection policy, since the reference strategy (SB) effectively ranks variables at a node by a score, and picks the highest-scoring variable, i.e., the score itself is not important.

Below we provide a concrete instantiation of L2S-DISCO for local search based acquistion function optimization that will be employed for our empirical evaluation.

8.1.3 Instantiation of L2S-DISCO for Local Search

Recall that local search based AFO solver performs multiple runs of local search guided by the acquisition function $\mathcal{AF}(\mathcal{M}, x)$ from different random starting states. The search space is defined over complete structures, where each state corresponds to a complete structure $x \in \mathcal{X}$. The successors of a state with structure x referred

as $\mathcal{N}(x)$, is the set of all structures $x' \in \mathcal{X}$ such that the hamming distance between x and x' is one. The effectiveness of local search depends critically on the quality of starting states. Therefore, we instantiate L2S-DISCO for local search and learn a search heuristic $\mathcal{H}(\theta, x)$ to select good starting states that will allow local search to uncover high-scoring structures from $\mathcal{X}$ according to $\mathcal{AF}(\mathcal{M}, x)$.

To instantiate L2S-DISCO for local search, we need to specify the two key elements: 1) The training data for learning the heuristic $\mathcal{H}(\theta, x)$?; and 2) The learning formulation to induce $\mathcal{H}(\theta, x)$ from online training data.

Algorithm 8 L2S-DISCO for local search

Input: $\mathcal{X}$= space of combinatorial structures, $\mathcal{AF}(\mathcal{M}, x)$= acquisition function, $\mathcal{H}(\theta, x)$= search heuristic from previous BO iteration, RankLearn= rank learner
Output: $\hat{x}_{next}$, the selected structure for function evaluation

Initialization: $\mathcal{T} \leftarrow \emptyset$ (training data of local search trajectories) and $\mathcal{S}_{start} \leftarrow \emptyset$ (set of starting states)
repeat
  Perform local search from a random state $x \in \mathcal{X}$ guided by heuristic $\mathcal{H}(\theta, x)$ to reach a local optima $x_{restart}$
  if $x_{restart} \in \mathcal{S}_{start}$ **then**
    $x_{start} \leftarrow$ random structure from $\mathcal{X}$
  else
    $x_{start} \leftarrow x_{restart}$
  end if
  Perform local search from x_{start} guided by $\mathcal{AF}(\mathcal{M}, x)$
  Add the new search trajectory and $\mathcal{AF}(\mathcal{M}, x_{end})$ to $\mathcal{T}$
  Update heuristic $\mathcal{H}(\theta, x)$ via rank learner using $\mathcal{T}$
  $\mathcal{S}_{start} \leftarrow \mathcal{S}_{start} \cup x_{start}$
until convergence or maximum iterations
$\hat{x}_{next} \leftarrow$ best scoring structure as per $\mathcal{AF}(\mathcal{M}, x)$ found during the entire search process
return the selected structure for evaluation $\hat{x}_{next}$

1) Training data. The set of search trajectories $\mathcal{T}$ obtained by performing local search from different starting states and acquisition function scores for local optima correspond to the training data. Each search trajectory $T \in \mathcal{T}$ consists of the sequence of states from the starting state x_{start} to the local optima x_{end}. Suppose $V(T)$=$\mathcal{AF}(\mathcal{M}, x_{end})$ represents the acquisition function score of the local optima for local search trajectory T.

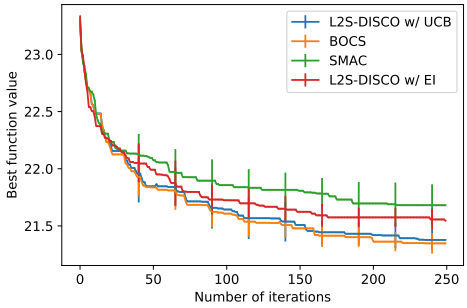
2) Rank learning formulation. The role of the heuristic $\mathcal{H}(\theta, x)$ is to rank candidate starting states according to their utility in uncovering high-scoring structures from $\mathcal{X}$ via local search. Recall that if we perform local search guided by $\mathcal{AF}(\mathcal{M}, x)$ from any state x on a search trajectory $T \in \mathcal{T}$, we will reach the same local optima with acquisition function score $V(T)$. In other words, every state on the trajectory $T \in \mathcal{T}$ has the same utility. Therefore, we formulate the problem of learning the search heuristic as an instance of bipartite ranking [1]. Specifically, for every pair of search trajectories $T_1, T_2 \in \mathcal{T}$, if $V(T_1) > V(T_2)$, then we want to rank every state on the trajectory T_1 better than every state on the trajectory T_2. We will generate one ranking example for every pair of states (x_1, x_2), where x_1 is a state on the trajectory T_1 and x_2 is a state on the trajectory T_2. The aggregate set of ranking examples are given to an off-the-shelf rank learner to induce $\mathcal{H}(\theta, x)$, where θ are the parameters of the ranking function. In our experiments, we employed RankNet [36] as the base

rank learner. We leveraged existing code[1] for our purpose.

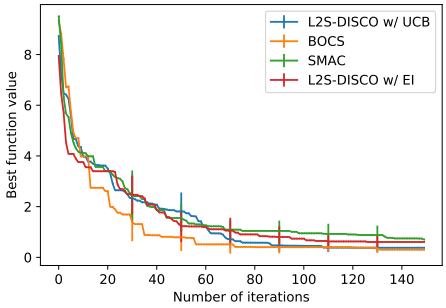
L2S-DISCO for local search based optimization. Figure 8.2 illustrates L2S-DISCO instantiation for local search based acquisition function optimization. At a high-level, each iteration of L2S-DISCO consists of two alternating local search runs. First, local search guided by heuristic $\mathcal{H}$ to select the starting state. Second, local search guided by $\mathcal{AF}$ from the selected starting state. After each local search run, we get a new local search trajectory, and the heuristic function $\mathcal{H}$ is updated to be consistent with this new search trajectory.

Algorithm 8 shows the pseduo-code for learning based local search to solve AFO problems arising in BO iterations. It reuses the learned search heuristic from the previous BO iteration and updates it in an online manner using the new training data generated during AF optimization. In each iteration, we perform the following sequence of steps. First, we perform local search from a random state guided by the search heuristic $\mathcal{H}(\theta, x)$ until reaching the local optima $x_{restart}$ to select the next starting state. Second, if $x_{restart}$ was not explored as a starting state in previous local search iterations, we select $x_{restart}$ as the starting state to perform local search guided by $\mathcal{AF}(\mathcal{M}, x)$ and add the local search trajectory to our training data. Third, we update the search heuristic $\mathcal{H}(\theta, x)$ using the newly added training example via rank learner. We repeat the above three steps until convergence or maximum iterations.

[1] `https://github.com/shiba24/learning2rank`



(a) Contamination domain with no. of stages $d = 25$ and $\lambda = 10^{-4}$ over 250 iterations.



(b) Ising domain with number of nodes $d = 24$ and $\lambda = 10^{-2}$ over 150 iterations.

Figure 8.3: Results for contamination and ising domain (**minimization**).

This instantiation of L2S-DISCO is similar in spirit to the STAGE algorithm [52]. At the end, we return the best scoring structure uncovered during the search $\hat{x}_{next}$ for function evaluation.

8.2 Experiments and Results

In this section, we first describe our experimental setup and then discuss the results of L2S-DISCO and baseline methods.

8.2.1 Experimental Setup

Benchmark Domains. We employ five diverse benchmark domains for our empirical evaluation.

1. Contamination. The problem considers a food supply with d stages, where a binary {0,1} decision must be made at each stage to prevent the food from being

contaminated with pathogenic micro-organisms [104, 16]. Each prevention effort at stage i can be made to decrease the contamination by a given random rate Γ_i and incurring a cost c_i. The contamination spreads with a random rate Λ_i if no prevention effort is taken. The overall goal is to ensure that the fraction of contaminated food at each stage i does not exceed an upper limit U_i with probability at least $1-\epsilon$ while minimizing the total cost of all prevention efforts. Following [16], the lagrangian relaxation based problem formulation is:

$$\arg\min_{x} \sum_{i=1}^{d} \left[c_i x_i + \frac{\rho}{T} \sum_{k=1}^{T} 1_{\{Z_k > U_i\}} \right] + \lambda \|x\|_1$$

where λ is a regularization coefficient, Z_i is the fraction of contaminated food at stage i, violation penalty coefficient ρ=1, and T=100.

2. Sparsification of zero-field Ising models. The distribution of a zero field Ising model $p(z)$ for $z \in \{-1, 1\}^n$ is characterized by a symmetric interaction matrix J^p whose support is represented by a graph $G^p = ([n], E^p)$ that satisfies $(i, j) \in E^p$ if and only if $J^p_{ij} \neq 0$ holds [16]. The overall goal is to find a close approximate distribution $q(z)$ while minimizing the number of edges in E^q. Therefore, the objective function in this case is a regularized KL-divergence between p and q as given below:

$$D_{KL}(p||q_x) = \sum_{(i,j)\in E^p} (J^p_{ij} - J^q_{ij}) E_p[z_i z_j] + log(Z_q/Z_p)$$

where Z_q and Z_p are partition functions corresponding to p and q respectively, and $x \in \{0,1\}^{E^q}$ is the decision variable representing whether each edge is present in E^q or not.

3. Low auto-correlation binary sequences (LABS). The problem is to find a binary {+1,-1} sequence $S = (s_1, s_2, \cdots, s_n)$ of given length n that maximizes *merit factor* defined over a binary sequence as given below:

$$\text{Merit Factor(S)} = \frac{n^2}{E(S)}$$
$$\text{where } E(S) = \sum_{k=1}^{n-1} \left(\sum_{i=1}^{n-k} s_i s_{i+k} \right)^2$$

The LABS problem has multiple applications in diverse scientific disciplines [171].

4. Network optimization in multicore chips. With Moore's law aging quickly, multicore architectures are considered very promising for parallel computing [41, 116, 59, 134, 74, 159, 117, 140, 141, 194, 108, 62, 162, 133, 224, 45, 118, 143, 10, 119, 51, 6, 11, 111, 109, 110, 112, 225, 7, 8, 9, 166, 167, 132, 148, 149, 161]. A key challenge in multicore research is to reduce the performance bottleneck due to data movement. One promising solution is to optimize the placement of communication links between cores to facilitate efficient data transfer. This optimization is typically guided by expensive simulators that mimics the real hardware. The network optimization problem is part of the rodinia benchmark [42] and uses the gem5-GPU

simulator [178]. There are 12 cores whose placements are fixed and the goal is to place 17 links between them to optimize performance: *66 binary variables*. There is one *constraint* to determine valid structures: existence of a viable path between any pair of cores. We report the performance improvement with respect to the provided baseline network.

5. Core placement optimization in multicore chips. This is another multicore architecture optimization problem from rodinia benchmark [42]. In this problem, we are given 64 cores of three types (8 CPUs, 40 GPUs, and 16 memory units) and they are connected by a mesh network (every core is connected to its four neighboring cores) to facilitate data transfer. The goal is to place the three types of cores to optimize performance: *64 categorical variables* with each taking three candidate values. We need to make sure that the *cardinality constraints* in terms of the number of cores of each type are satisfied. We report the performance improvement w.r.t the provided baseline placement.

Baseline Methods. We compare the local search instantiation of L2S-DISCO with two state-of-the-art methods: SMAC [106] and BOCS [16]. We employed open-source python implementations of both BOCS [2] and SMAC [3]. Since SMAC implementation does not support handling domain constraints to search over valid structures[4], we could not run SMAC for network optimization and core placement optimization

[2] `https://github.com/baptistar/BOCS`
[3] `https://github.com/automl/SMAC3`
[4] https://github.com/automl/SMAC3/issues/403

benchmarks. Similarly, SDP based solver for BOCS cannot handle constraints, so we employed simulated annealing based solver available in the BOCS code for those two benchmarks. We initialize the surrogate of all the methods by evaluating 20 random structures. For L2S-DISCO. we employed random forest model with 20 trees (tried two standard settings of scikit-learn library, namely, 10 and 20 trees, and got similar results) and two different acquisition functions (EI and UCB). For UCB, we use the adaptive rate recommended by [198] to set the exploration and exploitation trade-off parameter β_i value depending on the iteration number i. We ran L2S-DISCO (Algorithm 8) for a maximum of 60 iterations.

Evaluation Metric. We use the best function value achieved after a given number of iterations as a metric to evaluate all methods: SMAC, BOCS, and L2S-DISCO. The method that uncovers high-performing structures with less number of function evaluations is considered better. LABS is a maximization problem, but the remaining four benchmarks require the objective to be minimized. We use the total number of iterations similar to BOCS [16].

8.2.2 Results and Discussion

We discuss the results of L2S-DISCO and baseline methods on the five benchmarks below. All the reported results are averaged over 10 random runs (except for BOCS in cores placement optimization due to its poor scalability).

Contamination and Ising.

Figure 8.3 shows the comparison of L2S-DISCO with SMAC and BOCS baselines. We make the following observations. 1) Both L2S-DISCO variants that use EI and UCB acquisition functions perform better than SMAC. 2) L2S-DISCO with UCB performs better than the variant with EI. We observed a similar trend for the remaining three benchmarks also. Therefore, to avoid clutter, we only show the results of L2S-DISCO with UCB for the remaining benchmarks. 3) Results of L2S-DISCO are comparable to BOCS on the contamination problem. However, BOCS has a better anytime profile for ising domain. L2S-DISCO eventually matches the performance of BOCS after 90 iterations. The main reason BOCS performs slightly better in these two domains is that they exactly match the modeling assumptions of BOCS, which allows the use of SDP based solver to select structures for evaluation. Below we will show how the performance of BOCS degrades when the assumptions are not met, whereas L2S-DISCO peforms robustly across optimization problems of varying complexity.

LABS.

Figure 8.4 shows the comparison of L2S-DISCO with SMAC and BOCS baselines. We can see that L2S-DISCO clearly outperforms both BOCS and SMAC on this domain. BOCS has the advantage of SDP based solver, but its statistical model

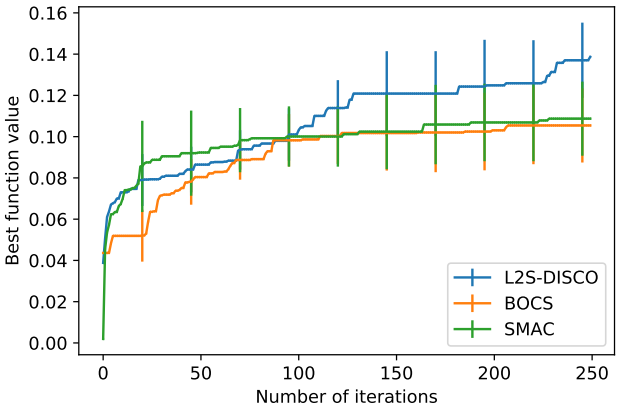


Figure 8.4: Results for LABS domain (**maximization**) with input sequence length n=30 over 250 iterations.

that accounts for only pair-wise interactions is limiting to account for the complexity in this problem. SMAC and L2S-DISCO both employ random forest model, but L2S-DISCO does better in terms of acquisition function optimization by integrating learning with search.

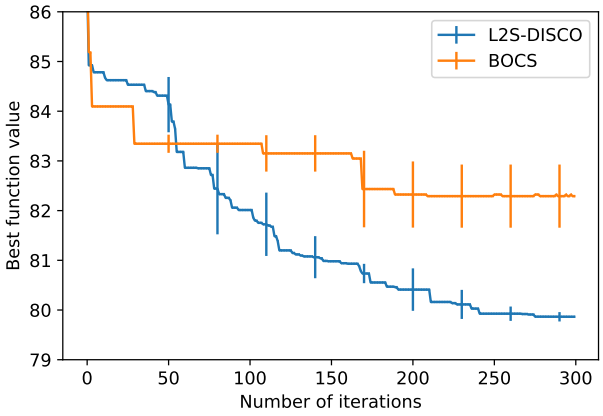


Figure 8.5: Results for network optimization in multicore chips (**minimization**) over 300 iterations.

Network optimization in multicore chips.

As mentioned earlier, we could not run SMAC for this problem as SMAC library does not allow to incorporate complex domain constraints. The SDP solver of BOCS is also not applicable due to complex constraints. Hence, we employ simulated annealing based solver for acquisition function optimization. Figure 8.5 shows the comparison of L2S-DISCO with BOCS baseline. We can see that L2S-DISCO performs significantly better than BOCS in this domain. BOCS seems to get stuck for long periods, whereas L2S-DISCO shows consistent improvement in uncovering high-performing structures. This behavior of BOCS can be partly attributed to the limitations of both surrogate model and acquisition function optimizer.

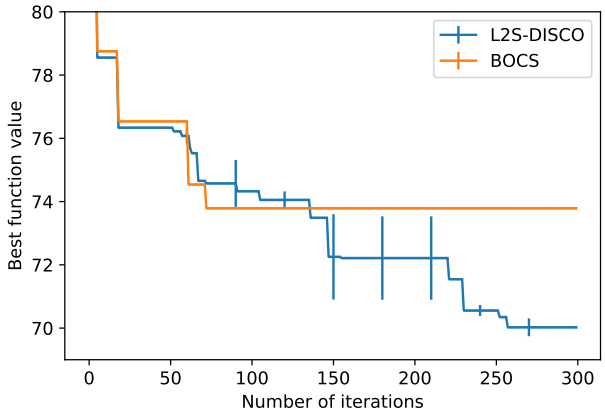


Figure 8.6: Results for core placement optimization in multicore chips (**minimization**) over 300 iterations.

Core placement optimization in multicore chips.

Figure 8.5 shows the comparison of L2S-DISCO with BOCS. L2S-DISCO significantly outperforms BOCS on this benchmark also. Additionally, BOCS scales poorly on this domain, where the discrete variables are non-binary. Recall that BOCS model was developed for binary variables and authors suggested the use of one-hot encoding to handle categorical variables. However, this transformation excessively increases the no. of dimensions. For example, we have 64 dimensions for L2S-DISCO, but it grows to 192 for BOCS due to one-hot encoding and makes its execution extremely slow. BOCS took one hour per single BO iteration on a machine with Intel Xeon(R) 2.5Ghz CPU and 96 GB memory. This is the main reason we could only perform one run of BOCS.

8.3 Summary

We introduced the L2S-DISCO framework that integrates machine learning with search-based optimization for optimizing expensive black-box functions over discrete spaces. We showed that instantiation of L2S-DISCO for local search based optimization yields significantly better performance than state-of-the-art methods on complex optimization problems.

CHAPTER NINE

CONCLUSION AND FUTURE DIRECTIONS

This chapter summarizes the main research contributions of this dissertation, lessons learned, and outlines some promising future research directions.

9.1 Summary

This thesis introduces a novel suite of methods designed to address the challenges associated with adaptive experimental design over large combinatorial spaces, a problem prevalent in numerous real-world scientific and engineering applications. Prior to this dissertation, Bayesian optimization (BO) for combinatorial spaces was an under-explored area with little to no principled prior work. This thesis significantly advanced the state-of-the-art in several key aspects, including surrogate modeling over different types of combinatorial structures (e.g., high-dimensional binary/categorical spaces, hybrid spaces consisting of a mxture of discrete and continuous variables, varying-sized graphs, and permutations); defining modern acquisition functions (e.g., Thompson sampling and input/output space entropy search) from the BO for continuous spaces literature for combinatorial spaces by developing tools to sample functions from Gaussian process surrogate models over binary/categorical spaces; tractable and effective approaches for acquisition function optimization; and associated theoretical analysis. The proposed methodologies enable principled and efficient exploration of

vast combinatorial design spaces, paving the way for new discoveries and optimizations in a broad range of engineering and science domains.

9.2 Lessons Learned

We list the main lessons learned from this dissertation below.

- There is no universal solution for Bayesian optimization over all combinatorial structures. Instead, effective methods require careful consideration of the specific types of structures (e.g., fixed-size sequences versus varying-sized graphs). A general principle emerging from this work is the incorporation of appropriate inductive biases into the data-driven surrogate models. This is exemplified in the LADDER framework, where domain-guided structured kernels complement deep representation learning for improved performance. In dictionary-based surrogate model, this appears in the form of parsimony priors from compressed-sensing-inspired dictionary-based embeddings. Hybrid diffusion kernel based surrogate model leverages regularization ideas derived from diffusion over different types of metric spaces.

- There is a trade-off between expressiveness of surrogate model and the tractability of acquisition function optimization. For instance, MerCBO demonstrates scalable acquisition function optimization for a restricted class of models parameterized by second-order Mercer features. Additionally, machine learning

techniques, as shown in L2S-DISCO, can automatically improve the effectiveness of search process based on past experience.

- This thesis also highlights the two-way relationship between AI/ML methods and Science/Engineering applications. While it is rather common to hear about AI/ML methods enabling high-impact real-world applications, this work demonstrates that the unique challenges posed by such applications can also drive the development of novel algorithms.

9.3 Future Work

Several promising avenues for future work stem from the methods developed in this thesis, some of them natural extensions while other require handling open challenges.

- First, the application of these techniques to a broader range of scientific and engineering problems holds significant potential as it has been demonstrated in multiple works recently. It will be interesting to develop and apply variants of the methods developed in this thesis to enable new unexplored scientific and engineering applications.

- Second, extending the dictionary-based embeddings approach described in Chapter 3, currently limited to fixed-size sequences, to handle varying-sized combinatorial inputs such as graphs presents an interesting avenue to explore.

- Third, the probabilistic surrogate models developed in this thesis could be leveraged in other machine learning settings, such as active learning or reinforcement learning in combinatorial domains. This could lead to new algorithms that can efficiently explore and exploit combinatorial decision spaces in these settings.

- Fourth, it maybe fruitful to consider incorporating the principles behind adaptive experimental design to improve the performance of offline model-based optimization algorithms [102, 50, 44].

- Finally, while this thesis addresses high-dimensional BO for fixed-size sequences (BODi), an important open challenge is the development of principled methods for handling high-dimensional permutation spaces. Given the prevalence of permutation spaces in various real-world applications, such as scheduling and ranking problems, this is a direction ripe for further investigation.

BIBLIOGRAPHY

[1] Shivani Agarwal and Dan Roth. Learnability of bipartite ranking functions. In *Proceedings of Annual Conference on Learning Theory (COLT)*, pages 16–31, 2005.

[2] Alaleh Ahmadianshalchi, Syrine Belakaria, and Janardhan Rao Doppa. Preference-aware constrained multi-objective bayesian optimization. In Sriraam Natarajan, Indrajit Bhattacharya, Richa Singh, Arun Kumar, Sayan Ranu, Kalika Bali, and Abinaya K, editors, *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD), Bangalore, India, January 4-7, 2024*, pages 182–191. ACM, 2024.

[3] Alaleh Ahmadianshalchi, Syrine Belakaria, and Janardhan Rao Doppa. Pareto front-diverse batch multi-objective bayesian optimization. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 10784–10794. AAAI Press, 2024.

[4] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., USA, 1988. ISBN 013617549X.

[5] Christof Angermueller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D Sculley. Population-based black-box optimization for biological sequence design. In *International Conference on Machine Learning*, pages 324–334. PMLR, 2020.

[6] Aqeeb Iqbal Arka, Srinivasan Gopal, Janardhan Rao Doppa, Deukhyoun Heo, and Partha Pratim Pande. Making a case for partially connected 3d noc: NFIC versus TSV. *ACM J. Emerg. Technol. Comput. Syst.*, 16(4):41:1–41:17, 2020.

[7] Aqeeb Iqbal Arka, Janardhan Rao Doppa, Partha Pratim Pande, Biresh Kumar Joardar, and Krishnendu Chakrabarty. Regraphx: Noc-enabled 3d heterogeneous reram architecture for training graph neural networks. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2021, Grenoble, France, February 1-5, 2021*, pages 1667–1672. IEEE, 2021.

[8] Aqeeb Iqbal Arka, Biresh Kumar Joardar, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Performance and accuracy tradeoffs for training graph neural networks on reram-based architectures. *IEEE Trans. Very Large Scale Integr. Syst.*, 29(10):1743–1756, 2021.

[9] Aqeeb Iqbal Arka, Biresh Kumar Joardar, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Dare: Droplayer-aware manycore reram architecture for training graph neural networks. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2021, Munich, Germany, November 1-4, 2021*, pages 1–9. IEEE, 2021.

[10] Aqeeb Iqbal Arka, Biresh Kumar Joardar, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. ReGraphX: NoC-enabled 3D heterogeneous ReRAM architecture for training graph neural networks. In *DATE*, 2021.

[11] Aqeeb Iqbal Arka, Biresh Kumar Joardar, Ryan Gary Kim, Dae Hyun Kim, Janardhan Rao Doppa, and Partha Pratim Pande. HeM3D: Heterogeneous manycore architecture based on monolithic 3D vertical integration. *ACM Trans. Design Autom. Electr. Syst.*, 26(2):16:1–16:21, 2021.

[12] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[13] Raul Astudillo and Peter Frazier. Bayesian optimization of composite functions. volume 97 of *Proceedings of Machine Learning Research*, pages 354–363. PMLR, 09–15 Jun 2019.

[14] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Ben-

jamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: Programmable Bayesian Optimization in PyTorch. *arxiv e-prints*, 2019. URL `http://arxiv.org/abs/1910.06403`.

[15] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo Bayesian optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[16] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *Proceedings of the 35th International Conference on Machine Learning*, pages 462–471, 2018.

[17] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *Proc. of ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 471–480. PMLR, 2018.

[18] Luis A Barrera, Anastasia Vedenko, Jesse V Kurland, Julia M Rogers, Stephen S Gisselbrecht, Elizabeth J Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, et al. Survey of variation in human transcription factors reveals prevalent dna binding changes. *Science*, 351(6280):1450–1454, 2016.

[19] Leonard Baumert, Solomon W Golomb, and Marshall Hall Jr. Discovery of an hadamard matrix of order 92. 1962.

[20] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *NeurIPS*, 2019.

[21] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Conference on Neural Information Processing Systems*, pages 7823–7833, 2019.

[22] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Information-theoretic multi-objective bayesian optimization with continuous approximations. *NeurIPS Workshop on Machine Learning for Engineering Modeling, Simulation, and Design*, 2020.

[23] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. In *AAAI*, pages 10035–10043, 2020.

[24] Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *AAAI*, 2020.

[25] Syrine Belakaria, Derek Jackson, Yue Cao, Janardhan Rao Doppa, and Xiaonan Lu. Machine learning enabled fast multi-objective optimization for electrified

aviation power system design. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020.

[26] Syrine Belakaria*, Derek Jackson*, Yue Cao, Janardhan Rao Doppa, and Xiaonan Lu. Machine learning enabled fast multi-objective optimization for electrified aviation power system design. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020.

[27] Syrine Belakaria*, Zhiyuan Zhou*, Aryan Deshwal, Janardhan Rao Doppa, Partha Pande, and Deuk Heo. Design of multi-output switched-capacitor voltage regulator via machine learning. In *Proceedings of the Twenty-Third IEEE/ACM Design Automation and Test in Europe Conference (DATE)*, 2020.

[28] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Output space entropy search framework for multi-objective bayesian optimization. *Journal of Artificial Intelligence Research*, 72:667–715, 2021.

[29] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554, 2011.

[30] Jakob Bernasconi. Low autocorrelation binary sequences:statistical mechanics and configuration space analysis. *Journal de Physique*, 48(4):559–567, 1987.

[31] Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. *Annals of statistics*, 41(3):1111, 2013.

[32] Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[33] Nikolay Bliznyuk, David Ruppert, Christine Shoemaker, Rommel Regis, Stefan Wild, and Pradeep Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2): 270–294, 2008.

[34] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph kernels: state-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*, 2020.

[35] David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

[36] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML)*, pages 89–96, 2005.

[37] Rainer E Burkard, Stefan E Karisch, and Franz Rendl. Qaplib–a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403, 1997.

[38] Rainer E Burkard, Eranda Cela, Panos M Pardalos, and Leonidas S Pitsoulis. The quadratic assignment problem. In *Handbook of combinatorial optimization*, pages 1713–1809. Springer, 1998.

[39] Leticia Cagnina, S. Esquivel, and Carlos Coello. Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32:319–326, 2008.

[40] Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. Word sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082, 2003.

[41] Luis Ceze, Mark D. Hill, and Thomas F. Wenisch. Arch2030: A vision of computer architecture research over the next 15 years. *CoRR*, abs/1612.03182, 2016. URL `http://arxiv.org/abs/1612.03182`.

[42] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W. Sheaffer, Sang-Ha Lee, and et al. Rodinia: A benchmark suite for heterogeneous computing. In *2009 IEEE international symposium on workload characterization (IISWC)*, 2009.

[43] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W Sheaffer, Sang-Ha Lee, and Kevin Skadron. Rodinia: A benchmark suite for heterogeneous computing. In *IEEE International Symposium on Workload Characterization (IISWC)*, pages 44–54. IEEE, 2009.

[44] Yassine Chemingui, Aryan Deshwal, Trong Nghia Hoang, and Janardhan Rao Doppa. Offline model-based optimization via policy-guided gradient search. In *AAAI Conference on Artificial Intelligence*, 2024.

[45] Wonje Choi, Karthi Duraisamy, Ryan Gary Kim, Janardhan Rao Doppa, Partha Pratim Pande, Diana Marculescu, and Radu Marculescu. On-chip communication network for efficient training of deep convolutional networks on heterogeneous manycore systems. *IEEE Transactions on Computers (TC)*, 67 (5):672–686, 2018.

[46] F. A. Rezaur Rahman Chowdhury, Chao Ma, Md. Rakibul Islam, Mohammad Hossein Namaki, Mohammad Omar Faruk, and Janardhan Rao Doppa. Select-and-evaluate: A learning framework for large-scale knowledge graph search. In Min-Ling Zhang and Yung-Kyun Noh, editors, *Proceedings of The 9th Asian Conference on Machine Learning, ACML 2017, Seoul, Korea, November 15-17, 2017*, volume 77 of *Proceedings of Machine Learning Research*, pages 129–144. PMLR, 2017.

[47] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning*, page 844–853, 2017.

[48] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[49] Hamid Dadkhahi, Karthikeyan Shanmugam, Jesus Rios, Payel Das, Samuel C. Hoffman, Troy David Loeffler, and Subramanian Sankaranarayanan. Combinatorial black-box optimization with expert advice. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, page 1918–1927, 2020.

[50] Manh Cuong Dao, Phi Le Nguyen, Thao Nguyen Truong, and Trong Nghia Hoang. Boosting offline optimizers with surrogate sensitivity. In *ICML*, 2024.

[51] Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Monolithic 3d-enabled high performance and energy efficient network-on-chip. In *ICCD*, pages 233–240, 2017.

[52] Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Design-space exploration and optimization of an energy-efficient and reliable 3D small-world network-on-chip. *IEEE TCAD*, 36(5), 2017.

[53] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *arXiv preprint arXiv:2006.05078*, 2020.

[54] Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *arXiv preprint arXiv:2210.10199*, 2022.

[55] Erik Daxberger. Personal communication about MiVaBO implementation and code.

[56] Erik Daxberger, Anastasia Makarova, Matteo Turchetta, and Andreas Krause. Mixed-variable bayesian optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2633–2639, 7 2020.

[57] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

[58] Aryan Deshwal and Janardhan Rao Doppa. Combining latent space and structured kernels for Bayesian optimization over combinatorial spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8185–8200, 2021.

[59] Aryan Deshwal, Nitthilan Kannappan Jayakodi, Biresh Kumar Joardar, Janardhan Rao Doppa, and Partha Pratim Pande. MOOS: A multi-objective design space exploration and optimization framework for NoC enabled many-core systems. *ACM TECS*, 2019.

[60] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Scalable combinatorial Bayesian optimization with tractable statistical models. *CoRR*, abs/2008.08177, 2020. URL `https://arxiv.org/abs/2008.08177`.

[61] Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Alan Fern. Optimizing discrete spaces via expensive evaluations: A learning to search framework. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[62] Aryan Deshwal, Syrine Belakaria, Ganapati Bhat, Janardhan Rao Doppa, and Partha Pratim Pande. Learning pareto-frontier resource management policies for heterogeneous socs: An information-theoretic approach. In *(DAC)*, 2021.

[63] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *ICML*, 2021.

[64] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Mercer features for efficient combinatorial Bayesian optimization. In *AAAI conference on Artificial Intelligence*, 2021.

[65] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 2632–2643. PMLR, 2021.

[66] Aryan Deshwal, Cory M. Simon, and Janardhan Rao Doppa. Bayesian optimization of nanoporous materials. *Mol. Syst. Des. Eng.*, 6:1066–1086, 2021. doi: 10.1039/D1ME00093D. URL `http://dx.doi.org/10.1039/D1ME00093D`.

[67] Aryan Deshwal, Syrine Belakaria, Janardhan Rao Doppa, and Dae Hyun Kim. Bayesian optimization over permutation spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022.

[68] Aryan Deshwal, Sebastian Ament, Maximilian Balandat, Eytan Bakshy, Janardhan Rao Doppa, and David Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. *CoRR*, abs/2303.01774, 2023. doi: 10.48550/arXiv.2303.01774. URL `https://doi.org/10.48550/arXiv.2303.01774`.

[69] Dragomir Z Djoković, Oleg Golubitsky, and Ilias S Kotsireas. Some new orders of hadamard and skew-hadamard matrices. *Journal of combinatorial designs*, 22(6):270–277, 2014.

[70] Janardhan Rao Doppa. Adaptive experimental design for optimizing combina-

torial structures. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4940–4945, 2021.

[71] Janardhan Rao Doppa, Alan Fern, and Prasad Tadepalli. HC-Search: A Learning Framework for Search-based Structured Prediction. *Journal of Artificial Intelligence Research (JAIR)*, 50:369–407, 2014.

[72] Janardhan Rao Doppa, Alan Fern, and Prasad Tadepalli. Structured prediction via output space search. *Journal of Machine Learning Research (JMLR)*, 15 (1):1317–1350, 2014.

[73] Janardhan Rao Doppa, Jun Yu, Chao Ma, Alan Fern, and Prasad Tadepalli. HC-Search for Multi-Label Prediction: An Empirical Study. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[74] Janardhan Rao Doppa, Justinian Rosca, and Paul Bogdan. Autonomous design space exploration of computing systems for sustainability: Opportunities and challenges. *IEEE Design and Test*, 36(5):35–43, 2019.

[75] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. *URL: http://archive.ics.uci.edu/ml*, 7(1), 2019.

[76] David K Duvenaud, Hannes Nickisch, and Carl E. Rasmussen. Additive gaussian processes. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and

K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 226–234, 2011.

[77] Stephan Eissman, Daniel Levy, Rui Shu, Stefan Bartzsch, and Stefano Ermon. Bayesian optimization and attribute adjustment. In *Proceedings of the Thirty Fourth Conference on Uncertainty in Artificial Intelligence*, 2018.

[78] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[79] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[80] José FS Bravo Ferreira, Yuehaw Khoo, and Amit Singer. Semidefinite programming approach for the quadratic assignment problem with a sparse graph. *Computational Optimization and Applications*, 69(3):677–712, 2018.

[81] Stanley F Florkowski III. Spectral graph theory of the hypercube. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2008. URL `https://apps.dtic.mil/dtic/tr/fulltext/u2/a493796.pdf`.

[82] S. Fujishige. *Submodular Functions and Optimization*. ISSN. Elsevier Science, 2005.

[83] Nickolas Gantzler, Aryan Deshwal, Janardhan Rao Doppa, and Cory M Simon. Multi-fidelity bayesian optimization of covalent organic frameworks for xenon/krypton separations. *Digital Discovery*, 2:1937–1956, 2023. doi: 10.1039/D3DD00117B. URL `https://pubs.rsc.org/en/content/articlehtml/2023/dd/d3dd00117b`.

[84] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for Bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319. PMLR, 2017.

[85] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[86] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *International Conference on Machine Learning*, volume 2014, pages 937–945, 2014.

[87] Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Process-*

ing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 7587–7597, 2018.

[88] Roman Garnett, Michael A Osborne, and Philipp Hennig. Active learning of linear embeddings for Gaussian processes. *arXiv preprint arXiv:1310.6740*, 2013.

[89] Eduardo C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes. *Neurocomputing*, 380:20–35, 2020.

[90] P. Gijsbers, E. LeDell, S. Poirier, J. Thomas, B. Bischl, and J. Vanschoren. An open source automl benchmark. *arXiv preprint arXiv:1907.00909.*, 2019. Accepted at AutoML Workshop at ICML 2019.

[91] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[92] Lena Gorelick, Yuri Boykov, Olga Veksler, Ismail Ben Ayed, and Andrew Delong. Submodularization for binary pairwise energies. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 1154–1161, 2014.

[93] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

[94] Anvita Gupta and James Zou. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1(2):105–111, 2019.

[95] Jacques Hadamard. Resolution d'une question relative aux determinants. *Bull. des sciences math.*, 2:240–246, 1893.

[96] Richard Hammack, Wilfried Imrich, and Sandi Klavžar. *Handbook of product graphs*. CRC press, 2011.

[97] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, pages 75–102, 2006.

[98] A Hedayat and Walter Dennis Wallis. Hadamard matrices and their applications. *The Annals of Statistics*, pages 1184–1238, 1978.

[99] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pages 918–926, 2014.

[100] José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, and Zoubin Ghahramani. A general framework for constrained bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(1):5549–5601, 2016.

[101] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. In *Proceedings of the 34th International Conference on Machine Learning*, page 1470–1479, 2017.

[102] Minh Hoang, Azza Fadhel, Aryan Deshwal, Jana Doppa, and Trong Nghia Hoang. Learning surrogates for offline black-box optimization via gradient matching. In *ICML*, 2024.

[103] Kathy J Horadam. *Hadamard matrices and their applications*. Princeton university press, 2012.

[104] Yingjie Hu, JianQiang Hu, Yifan Xu, Fengchun Wang, and Rong Zeng Cao. Contamination control in food supply chain. In *Proceedings of the Winter Simulation Conference*, WSC '10, pages 2678–2681, 2010. ISBN 978-1-4244-9864-2. URL `http://dl.acm.org/citation.cfm?id=2433508.2433840`.

[105] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration (extended version). Technical

Report TR-2010-10, University of British Columbia, Department of Computer Science, 2010. Available online: http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf.

[106] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on Learning and Intelligent Optimization*, pages 507–523, 2011.

[107] Shinji Ito and Ryohei Fujimaki. Large-scale price optimization via network flow. In *Advances in Neural Information Processing Systems*, pages 3855–3863, 2016.

[108] Nitthilan Kanappan Jayakodi, Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Design and optimization of energy-accuracy tradeoff networks for mobile platforms via pretrained deep models. *ACM Transactions on Embedded Computing Systems (TECS)*, 19(1):4:1–4:24, 2020.

[109] Nitthilan Kanappan Jayakodi, Janardhan Rao Doppa, and Partha Pratim Pande. Petnet: Polycount and energy trade-off deep networks for producing 3d objects from images. In *57th ACM/IEEE Design Automation Conference, DAC 2020, San Francisco, CA, USA, July 20-24, 2020*, pages 1–6. IEEE, 2020.

[110] Nitthilan Kanappan Jayakodi, Janardhan Rao Doppa, and Partha Pratim Pande. SETGAN: scale and energy trade-off gans for image applications on mobile platforms. In *IEEE/ACM International Conference On Computer Aided*

Design, ICCAD 2020, San Diego, CA, USA, November 2-5, 2020, pages 23:1–23:9. IEEE, 2020.

[111] Nitthilan Kannappan Jayakodi, Anwesha Chatterjee, Wonje Choi, Janardhan Rao Doppa, and Partha Pratim Pande. Trading-off accuracy and energy of deep inference on embedded systems: A co-design approach. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 37(11):2881–2893, 2018.

[112] Nitthilan Kannappan Jayakodi, Janardhan Rao Doppa, and Partha Pratim Pande. A general hardware and software co-design framework for energy-efficient edge AI. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2021, Munich, Germany, November 1-4, 2021*, pages 1–7. IEEE, 2021.

[113] Yunlong Jiao and Jean-Philippe Vert. The kendall and mallows kernels for permutations. In *International Conference on Machine Learning*, pages 1935–1944. PMLR, 2015.

[114] Yunlong Jiao and Jean-Philippe Vert. The weighted kendall and high-order kernels for permutations. *arXiv preprint arXiv:1802.08526*, 2018.

[115] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332. PMLR, 2018.

[116] Biresh Kumar Joardar, Ryan Gary Kim, Janardhan Rao Doppa, Partha Pratim Pande, Diana Marculescu, and Radu Marculescu. Learning-based application-agnostic 3D NoC design for heterogeneous manycore systems. *IEEE Transactions on Computers*, 68(6):852–866, 2018.

[117] Biresh Kumar Joardar, Aqeeb Iqbal Arka, Janardhan Rao Doppa, and Partha Pratim Pande. 3D++: Unlocking the next generation of high-performance and energy-efficient architectures using M3D integration. In *DATE*, 2021.

[118] Biresh Kumar Joardar, Janardhan Rao Doppa, Partha Pratim Pande, Hai Li, and Krishnendu Chakrabarty. AccuReD: High accuracy training of cnns on ReRAM/GPU heterogeneous 3D architecture. *IEEE TCAD*, 40(5):971–984, 2021.

[119] Biresh Kumar Joardar, Aryan Deshwal, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. High-throughput training of deep cnns on reram-based heterogeneous architectures via optimized normalization layers. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(5):1537–1549, 2022.

[120] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[121] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pages 3183–3191. PMLR, 2019.

[122] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pages 3183–3191. PMLR, 2019.

[123] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

[124] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pages 295–304. PMLR, 2015.

[125] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, and et al. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Conference on Neural Information Processing Systems*, 2016.

[126] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. Parallelised bayesian optimisation via thompson sampling. In *Proceedings of Machine Learning Research*, volume 84, pages 133–142, 2018.

[127] B. K. Kannan and S. N. Kramer. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 116(2):405–411, 06 1994.

[128] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2): 81–93, 1938.

[129] Elias Boutros Khalil, Pierre Le Bodic, Le Song, George Nemhauser, and Bistra Dilkina. Learning to branch in mixed integer programming. In *Proceedings of 30th AAAI Conference on Artificial Intelligence*, 2016.

[130] Jungtaek Kim and Seungjin Choi. On uncertainty estimation by tree-based surrogate models in sequential model-based optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 4359–4375. PMLR, 2022.

[131] Jungtaek Kim, Seungjin Choi, and Minsu Cho. Combinatorial Bayesian optimization with random mapping functions to convex polytopes. In *Uncertainty in Artificial Intelligence*, pages 1001–1011. PMLR, 2022.

[132] Ryan Gary Kim, Wonje Choi, Zhuo Chen, Janardhan Rao Doppa, Partha Pratim Pande, Diana Marculescu, and Radu Marculescu. Imitation learning for dynamic VFI control in large-scale manycore systems. *IEEE Trans. Very Large Scale Integr. Syst.*, 25(9):2458–2471, 2017.

[133] Ryan Gary Kim, Janardhan Rao Doppa, and Partha Pratim Pande. Machine learning for design space exploration and optimization of manycore systems. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, page 48. IEEE, 2018.

[134] Ryan Gary Kim, Janardhan Rao Doppa, Partha Pratim Pande, Diana Marculescu, and Radu Marculescu. Machine learning and manycore systems design: A serendipitous symbiosis. *IEEE Computer*, 51(7):66–77, 2018.

[135] Risi Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2002, pages 315–322, 2002.

[136] Risi Kondor and Jean-Philippe Vert. Diffusion kernels. *Kernel methods in computational biology*, pages 171–192, 2004.

[137] John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6(1):129–163, 2005.

[138] Michael Lam, Janardhan Rao Doppa, Sinisa Todorovic, and Thomas G. Dietterich. Hc-search for structured prediction in computer vision. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4923–4932, 2015.

[139] Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, 2017. doi: 10.1109/FOCS.2017.64.

[140] Dongjin Lee, Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Performance and thermal tradeoffs for energy-efficient monolithic 3D network-on-chip. *ACM TODAES*, 23(5):60:1–60:25, 2018.

[141] Dongjin Lee, Sourav Das, Janardhan Rao Doppa, Partha Pratim Pande, and Krishnendu Chakrabarty. Impact of electrostatic coupling on monolithic 3D-enabled network on chip. *ACM TODAES*, 24(6):62:1–62:22, 2019.

[142] Benjamin Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[143] Bing Li, Janardhan Rao Doppa, Partha Pratim Pande, Krishnendu Chakrabarty, Joe X. Qiu, and Hai (Helen) Li. 3d-reg: A 3d reram-based heterogeneous architecture for training deep neural networks. *ACM J. Emerg. Technol. Comput. Syst.*, 16(2):20:1–20:24, 2020.

[144] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris

Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.

[145] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.

[146] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Bi-population CMA-ES agorithms with surrogate models and line searches. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 1177–1184, 2013.

[147] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[148] Sumit K. Mandal, Ganapati Bhat, Chetan Arvind Patil, Janardhan Rao Doppa, Partha Pratim Pande, and Ümit Y. Ogras. Dynamic resource management of heterogeneous mobile platforms via imitation learning. *IEEE Trans. Very Large Scale Integr. Syst.*, 27(12):2842–2854, 2019.

[149] Sumit K. Mandal, Ganapati Bhat, Janardhan Rao Doppa, Partha Pratim Pande, and Ümit Y. Ogras. An energy-aware online learning framework for

resource management in heterogeneous platforms. *ACM Trans. Design Autom. Electr. Syst.*, 25(3):28:1–28:26, 2020.

[150] Horia Mania, Aaditya Ramdas, Martin J Wainwright, Michael I Jordan, and Benjamin Recht. On kernel methods for covariates that are rankings. *Electronic Journal of Statistics*, 12(2):2537–2577, 2018.

[151] Horia Mania, Aaditya Ramdas, Martin J Wainwright, Michael I Jordan, and Benjamin Recht. On kernel methods for covariates that are rankings. *Electronic Journal of Statistics*, 12(2):2537–2577, 2018.

[152] Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: an easy approach to molecular descriptor calculations. *Match*, 56(2):237–248, 2006.

[153] Natalie Maus, Haydn T. Jones, Juston S. Moore, Matt J. Kusner, John Bradshaw, and Jacob R. Gardner. Local latent space Bayesian optimization over structured inputs. *CoRR*, abs/2201.11872, 2022.

[154] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[155] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2 (117-129), 1978.

[156] Henry B Moss and Ryan-Rhys Griffiths. Gaussian process molecule property prediction with flowmo. *arXiv preprint arXiv:2010.01118*, 2020.

[157] Henry B Moss, Daniel Beck, Javier González, David S Leslie, and Paul Rayson. Boss: Bayesian optimization over string spaces. *arXiv preprint arXiv:2010.00979*, 2020.

[158] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[159] Shouvik Musavvir, Anwesha Chatterjee, Ryan Gary Kim, Dae Hyun Kim, Janardhan Rao Doppa, and Partha Pratim Pande. Power, performance, and thermal trade-offs in M3D-enabled manycore chips. In *DATE*, 2020.

[160] Mohammad Hossein Namaki, F. A. Rezaur Rahman Chowdhury, Md. Rakibul Islam, Janardhan Rao Doppa, and Yinghui Wu. Learning to speed up query planning in graph databases. In Laura Barbulescu, Jeremy Frank, Mausam, and Stephen F. Smith, editors, *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017*, pages 443–451. AAAI Press, 2017.

[161] Gaurav Narang, Raid Ayoub, Michael Kishinevsky, Janardhan Rao Doppa, and Partha Pratim Pande. Uncertainty-aware online learning for dynamic power management in large manycore systems. In *IEEE/ACM International Sympo-*

sium on Low Power Electronics and Design, ISLPED 2023, Vienna, Austria, August 7-8, 2023, pages 1–6. IEEE, 2023.

[162] Gaurav Narang, Aryan Deshwal, Janardhan Rao Doppa, Partha Pratim Pande, Raid Ayoub, and Mike Kishinevsky. Dynamic power management in large manycore systems: A learning-to-search framework. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2023.

[163] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pages 4752–4761. PMLR, 2019.

[164] Radford M Neal. Slice sampling. *Annals of statistics*, 31(3):705–741, 6 2003.

[165] Pascal Notin, José Miguel Hernández-Lobato, and Yarin Gal. Improving black-box optimization in vae latent space using decoder uncertainty. *arXiv preprint arXiv:2107.00096*, 2021.

[166] Chukwufumnanya Ogbogu, Aqeeb Iqbal Arka, Biresh Kumar Joardar, Janardhan Rao Doppa, Hai Helen Li, Krishnendu Chakrabarty, and Partha Pratim Pande. Accelerating large-scale graph neural network training on crossbar diet. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(11):3626–3637, 2022.

[167] Chukwufumnanya Ogbogu, Aqeeb Iqbal Arka, Lukas Pfromm, Biresh Kumar Joardar, Janardhan Rao Doppa, Krishnendu Chakrabarty, and Partha Pratim Pande. Accelerating graph neural network training on reram-based PIM architectures via graph and model pruning. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 42(8):2703–2716, 2023.

[168] Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian Optimization using the Graph Cartesian Product. In *NeurIPS*, 2019.

[169] ChangYong Oh, Jakub M. Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2910–2920, 2019.

[170] Changyong Oh, Efstratios Gavves, and Max Welling. Mixed variable bayesian optimization with frequency modulated kernels. In *Conference on Uncertainty in Artificial Intelligence*, 2021.

[171] Tom Packebusch and Stephan Mertens. Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical, 49 (2016) 165001*, 2015. doi: 10.1088/1751-8113/49/16/165001.

[172] Tom Packebusch and Stephan Mertens. Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical, 49 (2016) 165001*, 2015. doi: 10.1088/1751-8113/49/16/165001.

[173] Theodore P Papalexopoulos, Christian Tjandraatmadja, Ross Anderson, Juan Pablo Vielma, and David Belanger. Constrained discrete black-box optimization using mixed-integer programming. In *International Conference on Machine Learning*, pages 17295–17322. PMLR, 2022.

[174] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[175] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: reliable high-dimensional bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*, 36:1764–1793, 2023.

[176] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2020.

[177] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine Learning*, pages 745–750, 2007.

[178] Jason Power, Joel Hestness, Marc Orr, Mark Hill, and David Wood. gem5-gpu: A heterogeneous cpu-gpu simulator. *Computer Architecture Letters*, 13 (1), Jan 2014. ISSN 1556-6056. doi: 10.1109/LCA.2014.2299539. URL `http://gem5-gpu.cs.wisc.edu`.

[179] Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005.

[180] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[181] Gerhard Reinelt. Tsplib95. *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg*, 338:1–16, 1995.

[182] Gintaras V Reklaitis, A Ravindran, and Kenneth M Ragsdell. *Engineering optimization: methods and applications*. Wiley New York, 1983.

[183] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

[184] Binxin Ru, Ahsan S. Alvi, Vu Nguyen, Michael A. Osborne, and Stephen J. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning (ICML)*, 2020.

[185] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999.

[186] Walter Rudin. Real and complex analysis, mcgraw-hill. *Inc.,*, 1974.

[187] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *arXiv preprint: arXiv 1301.2609*, 2014.

[188] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.

[189] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[190] Bernhard Scholkopf, Sebastian Mika, Chris JC Burges, Philipp Knirsch, K-R Muller, Gunnar Ratsch, and Alexander J Smola. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5): 1000–1017, 1999.

[191] Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems*, pages 1209–1216, 2005.

[192] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[193] Irwin I Shapiro, Gordon H Pettengill, Michael E Ash, Melvin L Stone, William B Smith, Richard P Ingalls, and Richard A Brockelman. Fourth test of general relativity: preliminary results. *Physical Review Letters*, 20(22):1265, 1968.

[194] Harsh Sharma, Sumit K. Mandal, Janardhan Rao Doppa, Ümit Y. Ogras, and Partha Pratim Pande. SWAP: A server-scale communication-aware chiplet-based manycore PIM accelerator. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD)*, 41(11):4145–4156, 2022.

[195] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2960–2968, 2012.

[196] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems - Volume 2*, page 2951–2959, 2012.

[197] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[198] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

[199] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. of ICML*, pages 1015–1022. Omnipress, 2010.

[200] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

[201] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93, 2001.

[202] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[203] Ingo Steinwart, Philipp Thomann, and Nico Schmid. Learning with hierarchical gaussian kernels. *arXiv preprint arXiv:1612.00824*, 2016.

[204] Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective bayesian optimization using pareto-

frontier entropy. In *International Conference on Machine Learning*, pages 9279–9288. PMLR, 2020.

[205] Mitchell David Swanson and Ahmed H Tewfik. A binary wavelet decomposition of binary images. *IEEE Transactions on Image Processing*, 5(12):1637–1650, 1996.

[206] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In *ICML*, pages 9334–9345. PMLR, 2020.

[207] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

[208] Meng Tang, Ismail Ben Ayed, and Yuri Boykov. Pseudo-bound optimization for binary energies. In *European Conference on Computer Vision*, pages 691–707. Springer, 2014.

[209] Alastair P Thurlbeck and Yue Cao. Analysis and modeling of uav power system architectures. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–8. IEEE, 2019.

[210] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-

efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33, 2020.

[211] Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

[212] Tea Tušar, Dimo Brockhoff, and Nikolaus Hansen. Mixed-integer benchmark problems for single-and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 718–726, 2019.

[213] Richard S Varga. *Geršgorin and his circles*, volume 36. Springer Science & Business Media, 2010.

[214] Sébastien Verel, Bilel Derbel, Arnaud Liefooghe, Hernan Aguirre, and Kiyoshi Tanaka. A surrogate model based on walsh decomposition for pseudo-boolean functions. In *International Conference on Parallel Problem Solving from Nature*, pages 181–193. Springer, 2018.

[215] Xingchen Wan, Vu Nguyen, Huong Ha, Bin Xin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10663–10674. PMLR, 2021.

[216] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, page 3627–3635, 2017.

[217] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754, 2018.

[218] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[219] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Annual Conference on Neural Information Processing Systems*, 2001.

[220] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT Press, 2006.

[221] Benjamin Yackley, Eduardo Corona, and Terran Lane. Bayesian network score approximation using a metagraph kernel. In *Advances in Neural Information Processing Systems*, pages 1833–1840, 2009.

[222] Kevin K. Yang, Yuxin Chen, Alycia Lee, and Yisong Yue. Batched stochastic bayesian optimization via combinatorial constraints design. In Kamalika

Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 3410–3419. PMLR, 16–18 Apr 2019.

[223] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

[224] Xiaoxuan Yang, Syrine Belakaria, Biresh Kumar Joardar, Huanrui Yang, Janardhan Rao Doppa, Partha Pratim Pande, Krishnendu Chakrabarty, and Hai Helen Li. Multi-objective optimization of reram crossbars for robust DNN inferencing under stochastic noise. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2021, Munich, Germany, November 1-4, 2021*, pages 1–9. IEEE, 2021.

[225] Xiaoxuan Yang, Huanrui Yang, Janardhan Rao Doppa, Partha Pratim Pande, Krishnendu Chakrabarty, and Hai Li. ESSENCE: exploiting structured stochastic gradient pruning for endurance-aware reram-based in-memory training systems. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 42(7):2187–2199, 2023.

[226] Yehong Zhang, Trong Nghia Hoang, and et al. Information-based multi-fidelity Bayesian optimization. In *Conference on Neural Information Processing Systems Workshop on Bayesian Optimization*, 2017.

[227] Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.