Optimizing Discrete Spaces via Expensive Evaluations: A Learning to Search Framework

Aryan Deshwal  
*Washington State University*  
aryan.deshwal@wsu.edu

Syrine Belakaria  
*Washington State University*  
syrine.belakaria@wsu.edu

Janardhan Rao Doppa  
*Washington State University*  
jana.doppa@wsu.edu

Alan Fern  
*Oregon State University*  
alan.fern@oregonstate.edu

Abstract

We consider the problem of optimizing expensive black-box functions over discrete spaces (e.g., sets, sequences, graphs). 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. We provide a concrete instantiation of L2S-DISCO for local search procedure and show the efficacy of L2S-DISCO over state-of-the-art algorithms by empirically evaluating it on diverse real-world benchmarks.

1. Introduction

Many scientific and engineering applications including materials and hardware design involve optimizing discrete spaces (e.g., sets, sequences, graphs) guided by expensive black-box function evaluations. For example, in the application of finding alloys with high creep-resistance, we need to search over subsets of a given set of candidate metals guided by physical lab experiments. The key challenge in this problem is to select a sequence of combinatorial structures to evaluate, in order to uncover high-performing structures as quickly as possible. This involves solving a combinatorial optimization problem in each Bayesian optimization (BO) iteration, whose difficulty depends critically on the complexity of statistical surrogate model learned from past function evaluations.

There is very limited work on BO over discrete spaces. SMAC Hutter et al. (2010, 2011) is one canonical baseline which employs random forest as surrogate model and a hand-designed local search procedure for optimizing the acquisition function. BOCS Baptista and Poloczek (2018) is a state-of-the-art method that employs a second order linear Bayesian model defined over binary variables as the surrogate model. However, this simple model with second-order interactions may not suffice for optimization problems with complex interactions. There is also work on solving BO over discrete spaces by learning continuous representation from data and perform BO in this continuous latent space Gómez-Bombarelli et al. (2018). The main drawback of this method is that it generates a large fraction of invalid structures while also requiring a large database of “relevant” structures, for learning the latent space representation.
In this paper, we introduce a new learning-to-search framework referred as L2S-DISCO to select the sequence of combinatorial structures for evaluation. L2S-DISCO employs a search procedure (e.g., local search with multiple restarts) guided by appropriate search control knowledge (e.g., heuristic function to select good starting states), and continuously improves the control knowledge using advanced machine learning techniques. We provide a concrete instantiation of L2S-DISCO for local search based optimization by specifying the form of training data, and a rank learning formulation to update the search heuristic for selecting promising starting states. Experimental results on diverse benchmarks show the efficacy of L2S-DISCO on complex real-world problems.

2. Problem Setup and Challenges

**Problem definition.** Let $X$ be a combinatorial space of objects to be optimized over, where each element $x = \{v_1, \ldots, v_k\} \in X$ is a discrete structure (e.g., set, sequence, graph) and each variable $v_i$ in $x$ can take candidate values from a finite set $C(v_i)$. We assume an unknown real-valued objective function $F: X \mapsto \mathbb{R}$, which provides noisy evaluations for each candidate structure $x \in X$. The main goal is to find a structure $x \in X$ that approximately optimizes $F$ by conducting a limited number of function evaluations.

**Bayesian optimization formulation and challenges.** Bayesian optimization (BO) methods Shahriari et al. (2016); Belakaria et al. (2019 2020a b c) build a surrogate statistical model $M$, e.g., Gaussian Process (GP), from the training data of past function evaluations and employ it to sequentially select a sequence of inputs for evaluation to solve the problem. The selection of inputs is performed by optimizing an acquisition function $AF$ that is parameterized by the current model $M$ and input $x \in X$ to score the utility of candidate inputs for evaluation. Some example acquisition functions include expected improvement (EI) Jones et al. (1998) and upper-confidence bound (UCB) Srinivas et al. (2010). There are two key challenges in using BO for discrete spaces. 1. *Surrogate statistical modeling:* GPs are the popular choice for building statistical models in BO over continuous spaces which require defining an appropriate kernel on the combinatorial space. Random forest (RF) models can be used as an alternate generic choice to handle discrete spaces. In this work, we employ RF models as part of our experiments. 2. *Acquisition function optimization.* In each iteration of BO, we need to solve the following optimization problem to select the next candidate structure for evaluation.

$$x_{next} = \arg\max_{x \in X} AF(M, x)$$ (1)

The key challenge for discrete spaces is that, Equation 1 corresponds to solving a general combinatorial optimization problem. The effectiveness of BO critically depends on the accuracy of solving this optimization problem. In this paper, our main focus is on addressing this challenge using a novel learning to search framework.

3. L2S-DISCO: A Learning to Search Framework

L2S-DISCO integrates machine learning techniques and AI 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”; 3) Allows to incorporate prior knowledge.

Overview of L2S-DISCO. L2S-DISCO is parameterized by a search space $S$ over structures, a learned function $A\mathcal{F}(\mathcal{M}, x \in \mathcal{X})$ 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 the search control knowledge $\mathcal{H}$ using the online training data generated from the recent search experience. Fig 1 illustrates the instantiation of L2S-DISCO for local search. Each structure $x$ uncovered during the entire search is scored according to $A\mathcal{F}(\mathcal{M}, x)$ and we select the highest scoring structure $x_{\text{next}}$ for function evaluation. We perform experiment using the selected structure $x_{\text{next}}$ and observe the outcome $\mathcal{F}(x_{\text{next}})$. The statistical model $\mathcal{M}$ is updated using the new training example $(x_{\text{next}}, \mathcal{F}(x_{\text{next}}))$. We repeat the next iteration of BO via L2S-DISCO using 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. Below we provide a concrete instantiation of L2S-DISCO for local search based acquisition function optimization that will be employed for our empirical evaluation.

3.1 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 $A\mathcal{F}(\mathcal{M}, x)$ 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.
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 $H(\theta, x)$ to select good starting states that will allow local search to uncover high-scoring structures from $\mathcal{X}$ according to $\mathcal{AF}(M, x)$. The two key elements of L2S-DISCO for local search are defined below:

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 $x_{\text{start}}$ to the local optima $x_{\text{end}}$. Suppose $V(T) = \mathcal{AF}(M, x_{\text{end}})$ represents the acquisition function score of the local optima for $T$.

2) Rank learning formulation. The role of the heuristic $H(\theta, x)$ 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 $\mathcal{AF}(M, x)$ 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. 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 $H(\theta, x)$, where $\theta$ are the parameters of the ranking function.

L2S-DISCO for local search based optimization. Figure 1 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 $H$ to select the starting state. Second, local search guided by $\mathcal{AF}$ from

---

**Algorithm 1** L2S-DISCO for local search

**Input:** $\mathcal{X} =$ space of combinatorial structures, $\mathcal{AF}(M, x) =$ acquisition function, $H(\theta, x) =$ search heuristic from previous BO iteration, RANKLEARN =$ rank learner

**Output:** $\hat{x}_{\text{next}}$, the selected structure for function evaluation

1: Initialization: $\mathcal{T} \leftarrow \emptyset$ (training data of local search trajectories) and $S_{\text{start}} \leftarrow \emptyset$ (set of starting states)
2: repeat
3: Perform local search from a random state $x \in \mathcal{X}$ guided by heuristic $H(\theta, x)$ to reach a local optima $x_{\text{restart}}$
4: if $x_{\text{restart}} \in S_{\text{start}}$ then
5: $x_{\text{start}} \leftarrow$ random structure from $\mathcal{X}$
6: else
7: $x_{\text{start}} \leftarrow x_{\text{restart}}$
8: end if
9: Perform local search from $x_{\text{start}}$ guided by $\mathcal{AF}(M, x)$
10: Add the new search trajectory and $\mathcal{AF}(M, x_{\text{end}})$ to $\mathcal{T}$
11: Update heuristic $H(\theta, x)$ via rank learner using $\mathcal{T}$
12: $S_{\text{start}} \leftarrow S_{\text{start}} \cup x_{\text{start}}$
13: until convergence or maximum iterations
14: $\hat{x}_{\text{next}} \leftarrow$ best scoring structure as per $\mathcal{AF}(M, x)$ found during the entire search process
15: return the selected structure for evaluation $\hat{x}_{\text{next}}$
the selected starting state. After each local search run, we get a new local search trajectory, and the heuristic function \( H \) is updated to be consistent with this new search trajectory. Algorithm 1 shows the pseudo-code for learning based local search to solve AFO problems arising in BO iterations. This instantiation of L2S-DISCO is similar in spirit to the STAGE algorithm Boyan and Moore (2000).

4. Experiments and Results

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

4.1 Experimental Setup

**Benchmark Domains.** We employ four 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 Hu et al. (2010). Following Baptista and Poloczek (2018), the la-grangian relaxation based problem formulation is

\[
\min_x \sum_{i=1}^{d} \left( c_i x_i + \frac{\rho}{T} \sum_{k=1}^{T} 1\{z_k > U_i\} \right) + \lambda \|x\|_1
\]

where \( c_i \) is the cost of prevention effort, \( \Lambda_i \) is the rate of spread of contamination, \( \lambda \) is a regularization coefficient, \( U_i \) is the upper limit on the contamination level, \( Z_i \) is the fraction of contaminated food at stage \( i \), violation penalty coefficient \( \rho=1 \), 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 Baptista and Poloczek (2018). 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 \( q \):

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

where \( Z_q \) and \( Z_p \) are partition functions corresponding to \( p \) and \( q \) respectively, and \( x \in \{0,1\}^{E^p} \) 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:

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

4. **Network optimization in multicore chips.** There are 12 cores whose placements are fixed and the goal is to place 17 links between them to optimize performance: 66 binary variables in a multicore system. There is one constraint to determine valid structures: existence of a viable path between any pair of cores.

**Baseline Methods.** We compare the local search instantiation of L2S-DISCO with two state-of-the-art methods: SMAC Hutter et al. (2011) and BOCS Baptista and Poloczek (2018) on the best function value achieved after a given number of iterations as a metric. The method that uncovers high-performing structures with less number of function evaluations is considered better. For more details, please see the full paper Deshwal et al. (2020).
Figure 2: Results on four different benchmarks comparing L2S-DISCO with BOCS and SMAC.

4.2 Results and Discussion

Figure 2 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 on contamination and Ising domain. 2) Results of L2S-DISCO are comparable to BOCS on the contamination problem. 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. 3) L2S-DISCO clearly outperforms both BOCS and SMAC on LABS domain. BOCS has the advantage of SDP based solver, but its statistical model 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. 4) We can see that L2S-DISCO performs significantly better than BOCS in the network optimization 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.
 References


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), 2020b.


