Combinatorial Bayesian Optimization using the Graph Cartesian Product

QUVA Deep Vision Lab

Qualcomm AI research

Changyong Oh[+], Jakub M. Tomczak[*], Efstratios Gavves[+], Max Welling[+*]

[+] QUVA Lab, University of Amsterdam
[*] Qualcomm Technologies Netherlands B.V.

C.Oh@uva.nl

1. Introduction

- Black-box function optimization:

$$x_{opt} = argmin_{x\in X} f(x)$$

  - Non-differentiable f
  - Expensive to query f
  - Noisy f
- Typically, the search space X is continuous and the function f is assumed to be smooth.
- A more challenging problem is when the search space is **combinatorial** (e.g., variables are categorical or ordinal).
- There are two main challenges for combinatorial problems:
  1. How to define a smooth function on combinatorial objects?
     → **kernel (prior on smoothness)**
  2. How to efficiently select next points in a combinatorial space?
     → **acquisition function optimization**

2. Smoothness

- A space of combinatorial variables -

$$C_1, C_2, \cdots, C_{d-1}, C_d$$

$$\Downarrow$$

- A graph corresponding to the space -

$$G_i = G(C_i) \text{ for } i = 1, \cdots, d$$

$$G = G_1 \boxdot G_2 \boxdot \cdots \boxdot G_{d-1} \boxdot G_d$$

$$\Downarrow$$

- The concept of smoothness on functions on a graph -

Graph Fourier Analysis using graph Laplacian $L(G)$

Eigendecomposition of $\{(\lambda, u)\}$

Smaller λ ⇒ Smoother u

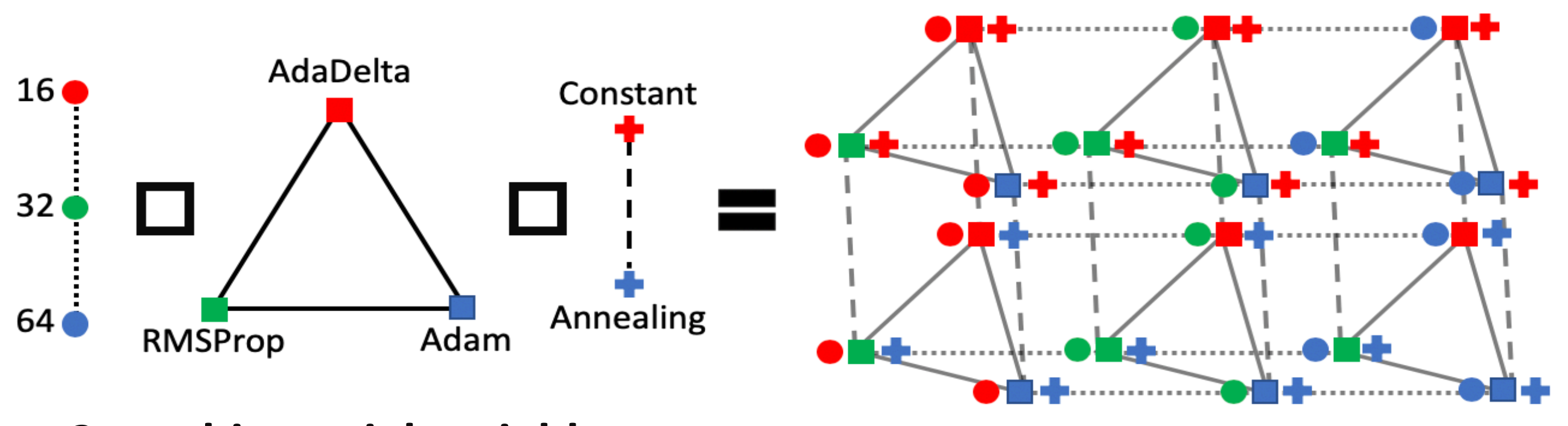
$$\Downarrow$$

- Smoothness of functions on combinatorial variables -

A kernel on a space of combinatorial variables

$$K((c_1, \cdots, c_d), (c_1', \cdots, c_d'))$$

3. Combinatorial graph (Search space)



- **3 combinatorial variables**
  - *Batch size* $C_1 = \{16, 32, 64\}$
  - *Optimizer* $C_2 = \{AdaDelta, RMSProp, Adam\}$
  - *Learning rate annealing* $C_3 = \{Constant,\ Annealing\}$
- **3 subgraphs**

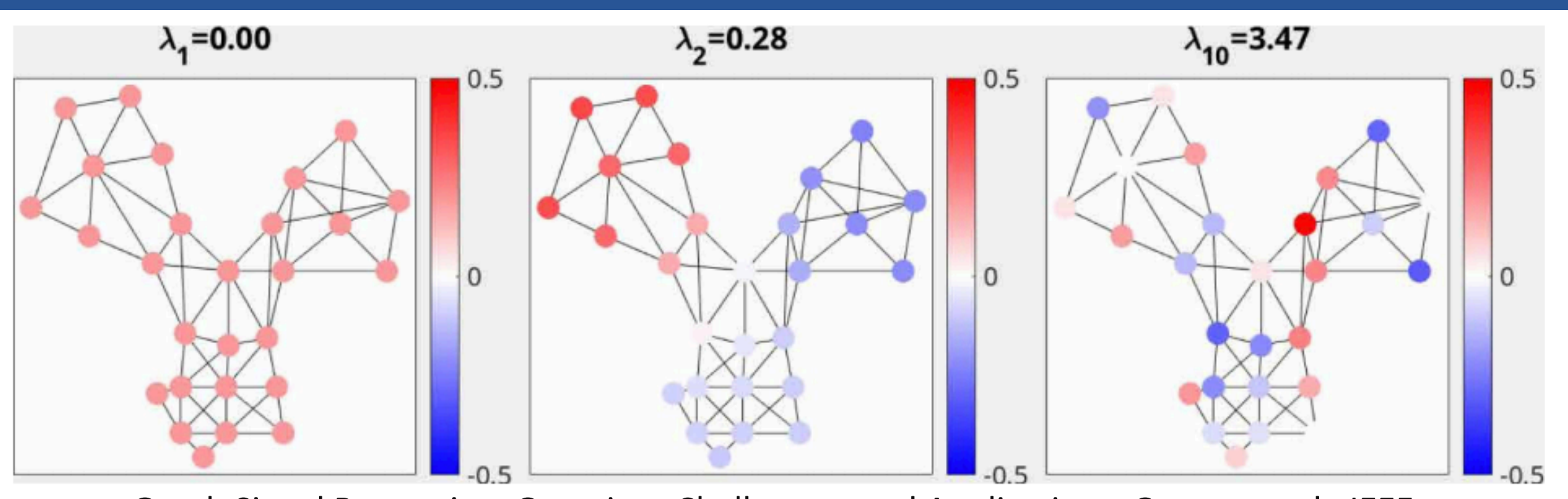
$$G_1 = G(C_1), G_2 = G(C_2), G_3 = G(C_3)$$

  - Complete graphs for categorical variables
  - Path graphs for ordinal variables
- **Combinatorial graph :** Graph Cartesian product of subgraphs

$$G = G_1 \boxdot G_2 \boxdot G_3$$

  - **Vertices** : sets of specific choices of categorical/ordinal variables.
  - **Edges** : similarities between 2 sets of choices

4. Graph Signal Processing



Graph Signal Processing: Overview, Challenges, and Applications, Ortega et. al., IEEE.

- **Graph Fourier Transform**
  - Graph Laplacian $L(G) = D - A$, D: deg. mat., A: adj. mat.
  - Eigendecomposition of $L(G) : \{(\lambda_i, u_i)\}_{i=1,\cdots,|V|}$
  - λ_i represents smoothness(energy) of u_i
- Approximate a function with eigenfunctions with small eig.vals.
  → f ≈ $\sum c_i u_i$ while trying to make c_i small if λ_i is large
- Diffusion kernel → GP **nonparametric** approach

$$K_G(v, \tilde{v}|\beta) = \left[e^{-\beta L(G)}\right]_{v,\tilde{v}} = \left[U e^{-\beta\Lambda} U^T\right]_{v,\tilde{v}}$$

5. Graph Cartesian Product

- Graph Cartesian product and Kronecker sum

$$L(G_1 \boxdot G_2) = L(G_1\ ) \oplus L(G_2)$$

$$= L(G_1) \otimes I_2 + I_1 \otimes L(G_2)$$

- Kronecker sum and matrix exponential

$$K_{G_1\boxdot G_2} = e^{-\beta L(G_1 \boxdot G_2)} = \mathrm{e}^{-\beta(L(G_1\ )\oplus L(G_2))}$$

$$= e^{-\beta(L(G_1)\otimes I_2 + I_1 \otimes L(G_2)\ )}$$

$$= e^{-\beta L(G_1)} \otimes e^{-\beta L(G_2)}$$

$$= K_{G_1} \otimes K_{G_2}$$

  - **Efficient computation of diffusion kernels**

$$O(\prod_{i=1}^{d} |V_i|^3) \rightarrow O(\sum_{i=1}^{d} |V_i|^3)$$

  - Able to handle large graphs

$$|V| \in \{2^{24}, 5^{21}, 2^{28}, 2^{43}, 2^{60}, 2^{32}\}$$

6. ARD Diffusion Kernel

- Diffusion kernel has a single kernel parameter β

$$K_G(v, \tilde{v}|\beta) = \bigotimes_i K_{G_i}(v_i, \tilde{v}_i|\boldsymbol{\beta})$$

- A multiplicand in the Kronecker product corresponds to a sub-graph
- Each sub-graph corresponds to a variable
- Variable-wise kernel parameter → **ARD diffusion kernel**

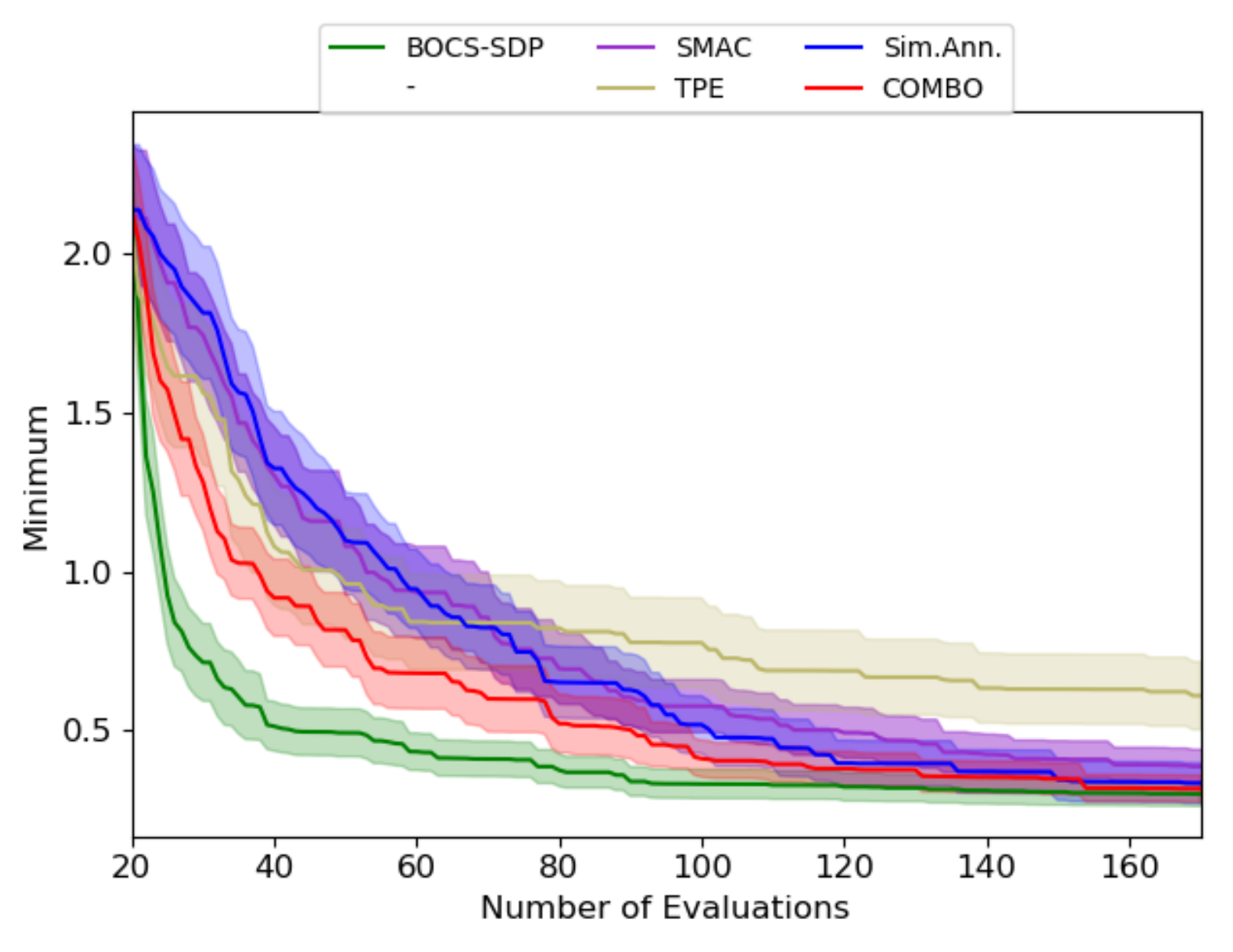
$$K_G(v, \tilde{v}|\beta) = \bigotimes_i K_{G_i}(v_i, \tilde{v}_i|\boldsymbol{\beta_i})$$

  - Relevant variables can be selected automatically
- Horseshoe priors on $\{\beta_i\}_{i=1,\cdots,d}$ promotes more effective feature selection
- We use slice sampling to sample $\{\beta_i\}_{i=1,\cdots,d}$.

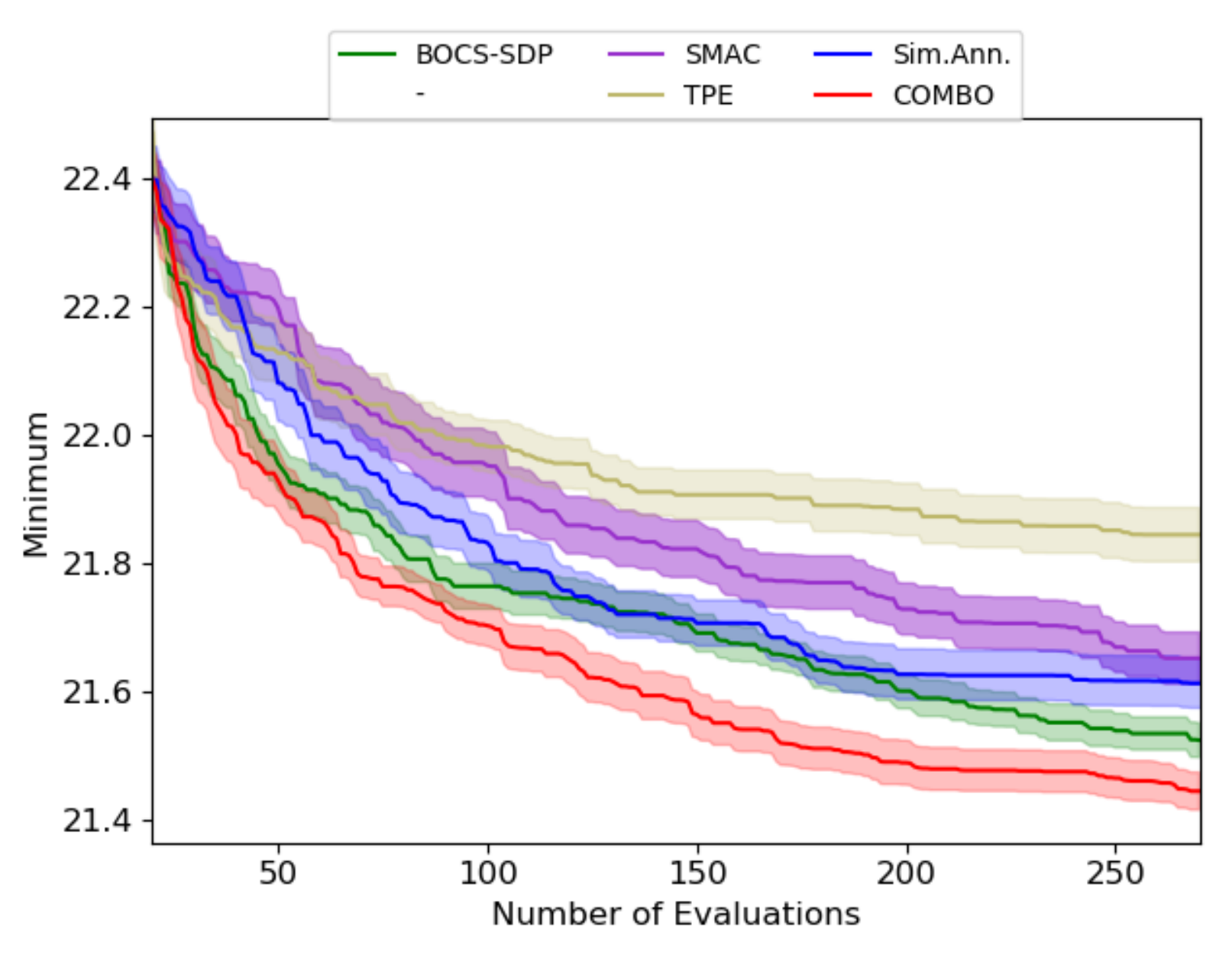
7. Experiments

Binary

Ising (2×24)

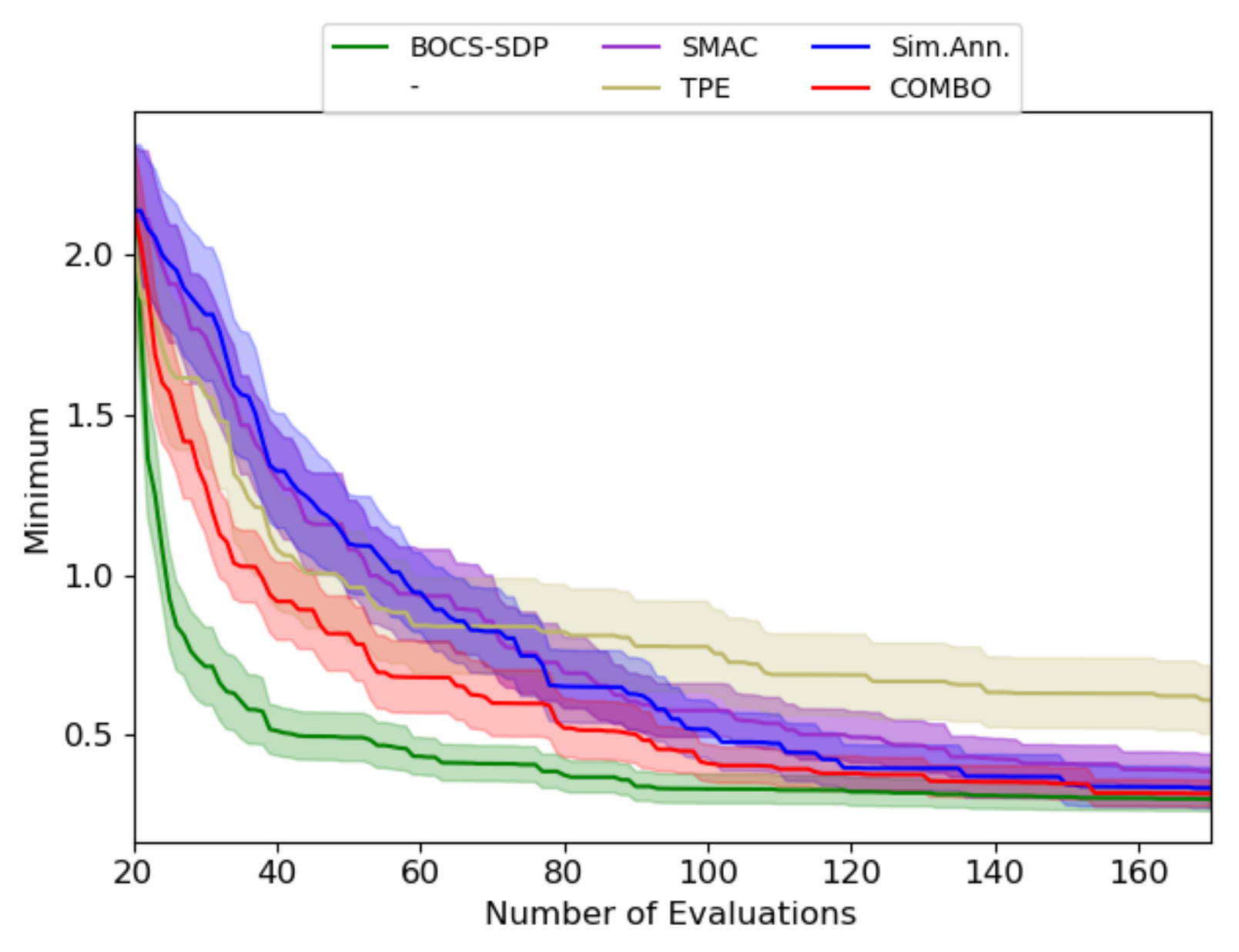


Contamination (2×21)

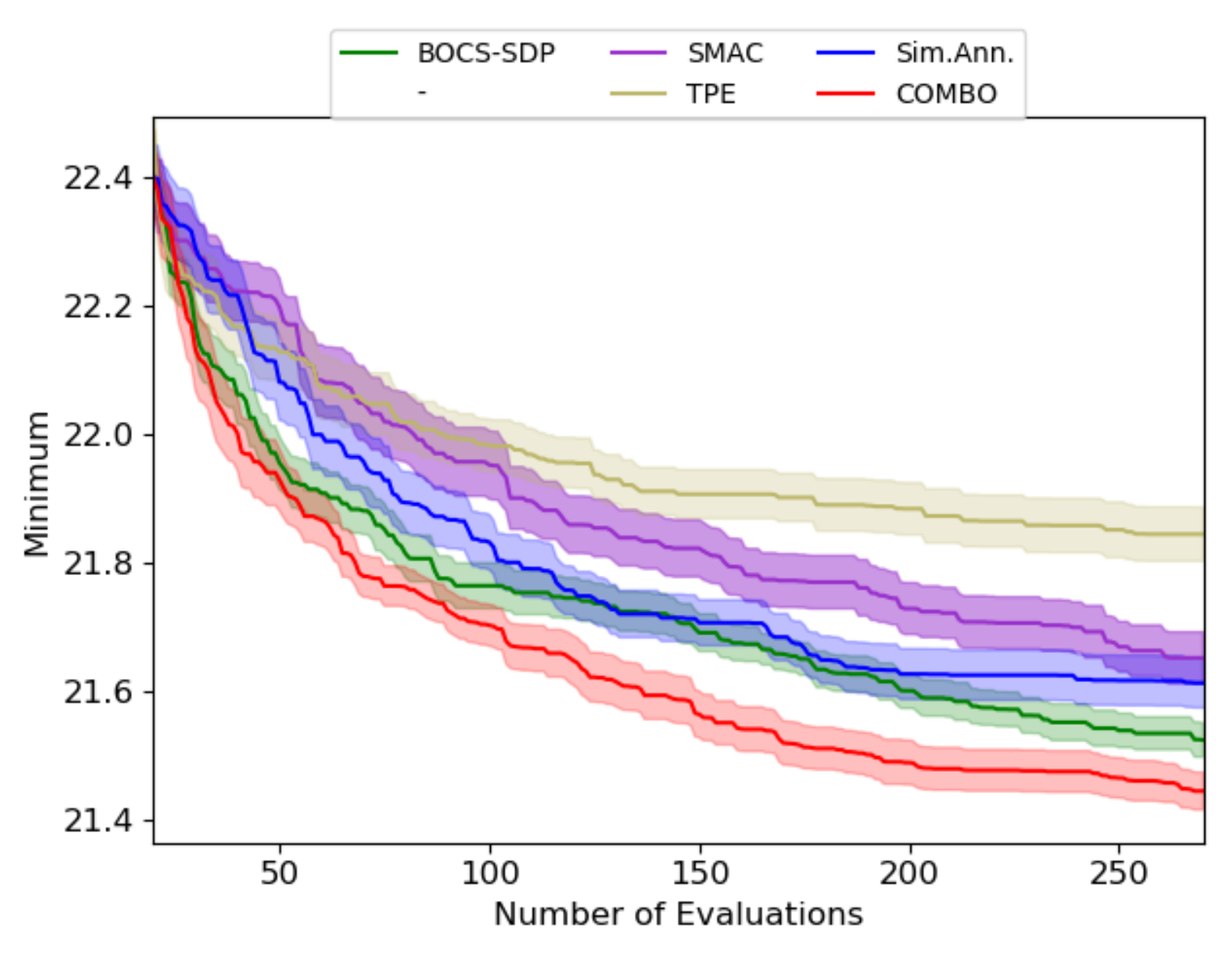


Ordinal & Multi-cate.

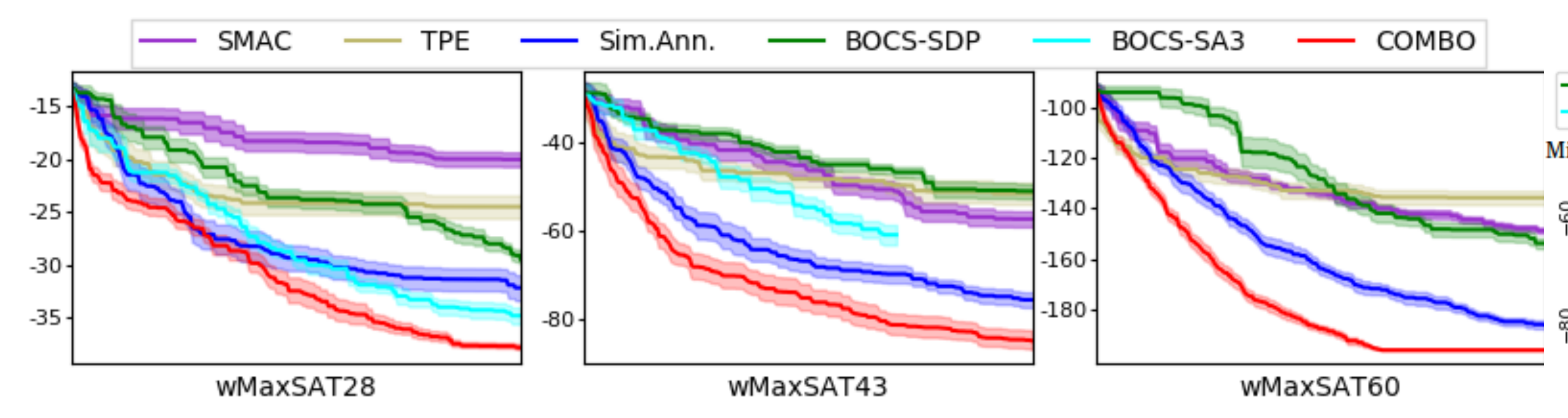
Branin (51×2)



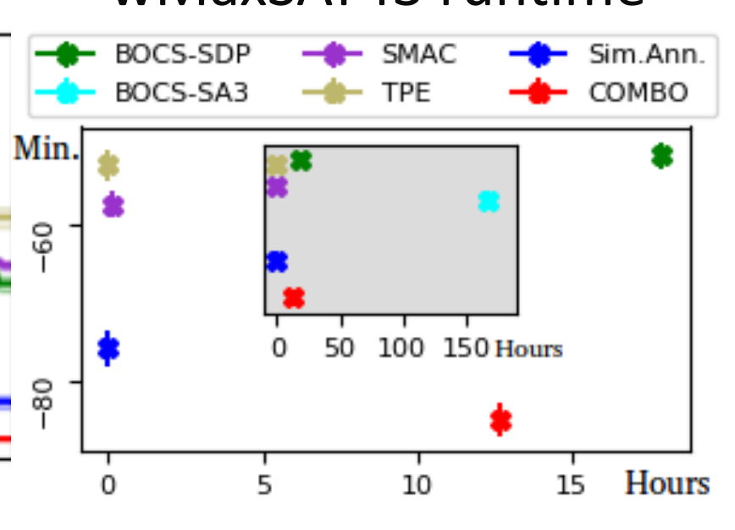
Pest (5×21)



Weighted MaxSAT

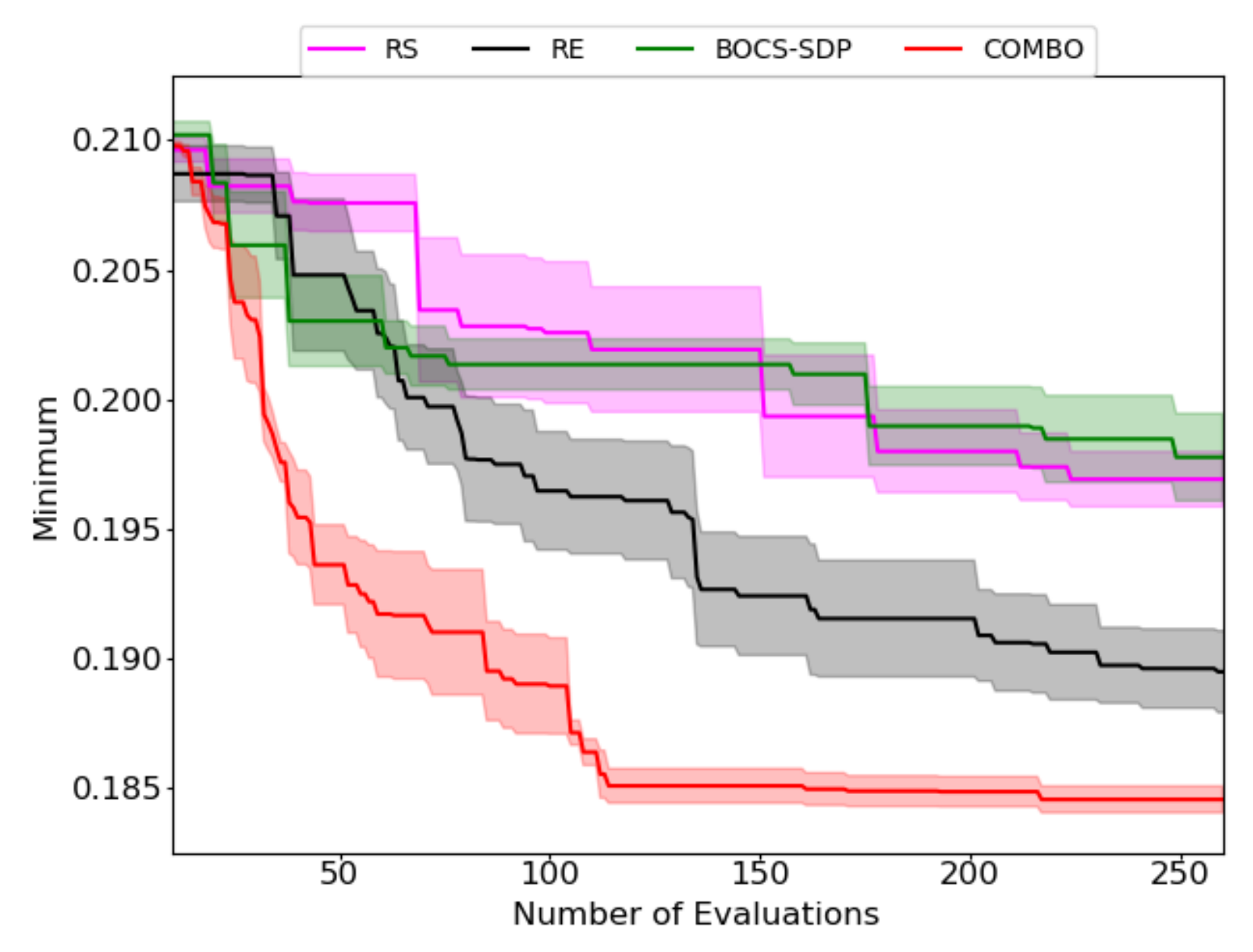


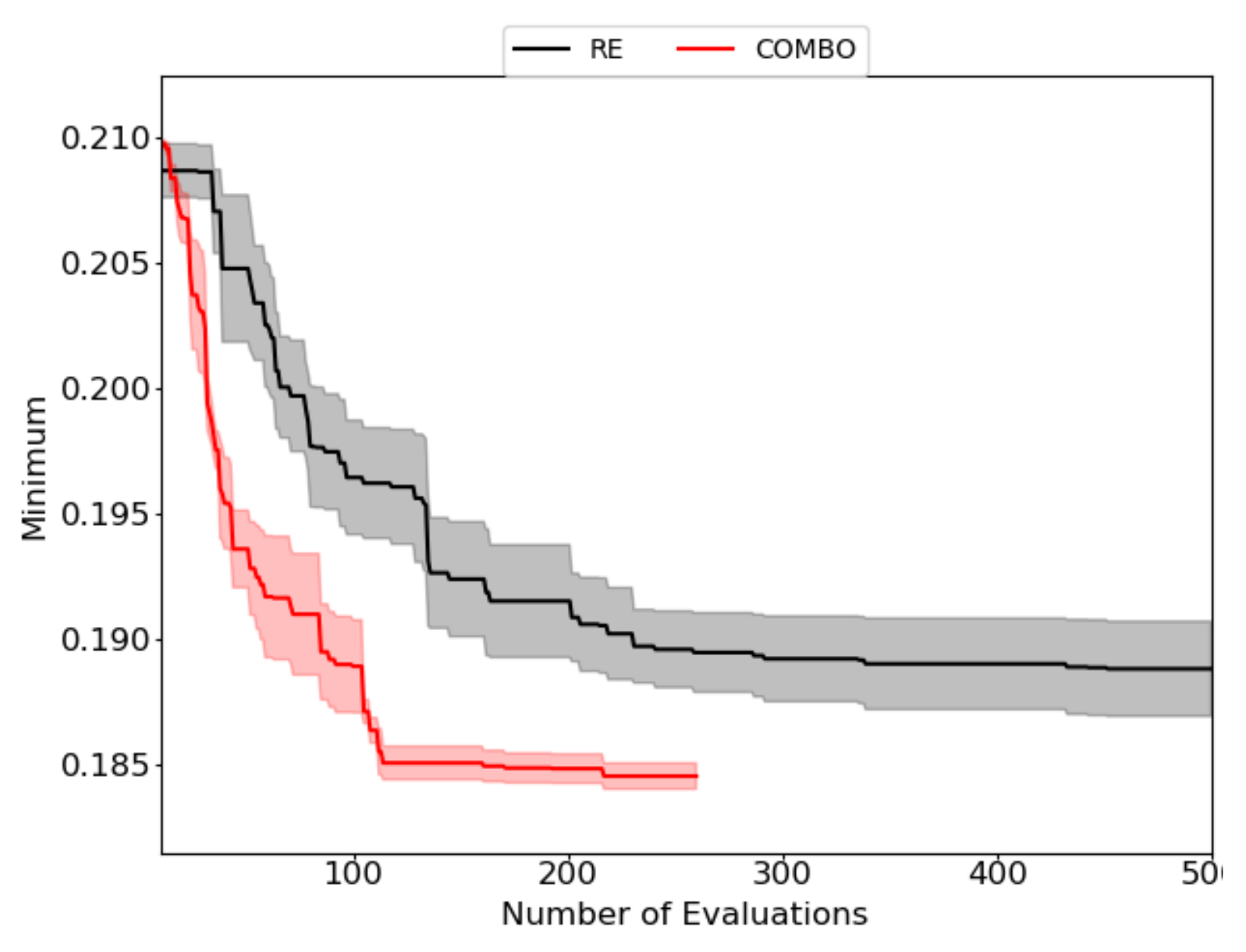
wMaxSAT43 runtime



Neural architecture Search

- A lighter version of neural architecture search
- On CIFAR10, valid. err. + FLOPS penalty
- 31 binary choice to choose an optimal block
- 3 Stack of chosen block is used for prediction





RE : E. Real et al. Regularized Evolution for Image Classifier Architecture Search, AAAI2019

8. Conclusions

- We propose COMBO, a Bayesian Optimization for combinatorial search spaces using GPs.
- The graph Cartesian product allows to reduce exponential complexity to linear complexity.
- The ARD diffusion kernel allows to better model complex functions on combinatorial objects.
- We show supremacy of COMBO on various combinatorial optimization problems.