

1 Pruning Surrogate Model Experts

The number of experts in optimization with expert advice can potentially grow exceedingly large due to the inclusion of either higher order interactions or a large number of variables or increase in dimensionality. In this section, we consider a strategy for pruning experts in an adaptive fashion.

We consider a setting where we start the algorithm with the set of all experts. We assume that the true function is sparse in the space of experts. Therefore, we consider a pruning strategy where such experts are deemed 'irrelevant' based on a condition, removed and are no longer updated. Once such experts are pruned, they no longer contribute to the surrogate model. This pruning strategy leads to a low amortized computational complexity. Our adaptive pruning algorithm is given in Algorithm 1. Let $\{\psi_i(\cdot)\}$ be the set of experts and let $f(x)$ be the true black-box function. The method keeps track of an empirical estimate of the metric: $\mathbb{E}[\psi_i(x)\text{sigmoid}(f(x))]$ for every i and deletes a sub-optimal one at time t , if it is by at least $\sqrt{2t \log(2t^2 d/\delta)}$ smaller than the optimal expert, i.e. the best expert according to the latter metric. This is a single parameter pruning strategy and depends on δ which is data-independent but could only depend on sparsity and dimensionality of the problem. We justify our pruning strategy for the abridged one-hot encoded Fourier representation by showing that our metric for pruning is large for the significant experts on average under the assumption of uniform distribution for sampling points x .

A direct approach to this problem would be to track the weights of an expert ($\alpha_{r,\mathcal{I}}$ or $\alpha_{\mathcal{I},\mathcal{J}}$ depending on the factorization we have chosen) and use a threshold to prune the experts. This method requires us to tune the thresholds and is highly data-dependent. To circumvent this problem, we take inspiration from an old technique called *timid pruning* [1]. Consider a binary classification problem where every expert $\psi_i(x)$ predicts on a random point $x \in \mathcal{X}$. Under a given distribution D over (x, y) , suppose p_i indicates the probability that $\psi_i(x)$ predicts the label y correctly. The *observed optimal expert* at any time t is defined as the one which has made the fewest mistakes until time t . In particular, we consider a pruning strategy that removes an expert if it has made at least $\sqrt{2t \log(2t^2 d/\delta)}$ more mistakes than the observed optimal until time t . Then with high probability $(1 - \delta)$, the optimal expert will not be removed according to Hoeffding bounds.

In our problem the function is real-valued and hence the above cannot be directly applied. The key idea is to consider a metric that is a thresholded version of $f(x)$ making it binary-valued. We use the following property about Boolean polynomials:

Theorem 1.1. [3] Suppose $p(x) : \{-1, 1\}^n \rightarrow \mathbb{R}$ is a Boolean polynomial where $p(x) = \sum_{S \in \mathcal{F}} \alpha_S \prod_{i \in S} x_i$ where \mathcal{F} is the family of subsets of $[n]$. Let $\text{PTF}(p(x)) = \text{sign}(p(x))$ where $\text{sign}(\cdot)$ is the sign function. Let γ_S be the Fourier coefficient of the monomial $\prod_{i \in S} x_i$ in $\text{PTF}(p(x))$. Then, we have $\sum_{S \in \mathcal{F}} |\gamma_S| \geq 1$.

The above theorem says that, the polynomial threshold function "amplifies" the coefficients of the weights that are nonzero in the original polynomial such that their ℓ_1 norm is at least 1 irrespective of what $\sum_S |\alpha_S|$ is. Let $\psi_S(x)$ denote the monomials of $f(x)$ that are present in Abridged one-hot encoded Fourier representation. Then, under the uniform distribution on $\{-1, 1\}^n$, we have: $|\gamma_S| = |\mathbb{E}[\text{PTF}(f(x))\psi_S(x)]|$. From the above and Theorem 1.1, we have that our metric for pruning over the family of monomials \mathcal{F} that are non-zero in $f(x)$ is at least one. Therefore, $\sum_{S \in \mathcal{F}} |\gamma_S| = \sum_{S \in \mathcal{F}} |\mathbb{E}[\text{PTF}(f(x))\psi_S(x)]| \geq 1$.

Moreover, in order to promote exploration in our pruning model, instead of $\text{PTF}(f(x))$, we consider a coin flip (with choices in $\{-1, 1\}$) with a probability of tossing $+1$ being equal to $\text{sigmoid}(f(x))$.

We point out the advantages of the proposed pruning strategy. This strategy has much less data dependence (because of the amplification due to thresholding) and is dependent on δ which only depends on problem dimensions (sparsity and dimension). Finally, this approach is independent of the surrogate model.

Algorithm 1 Pruning Surrogate Model Experts

```
1: Input: probability of error  $\delta$ 
2:  $t = 0$ 
3:  $\forall i \in [d] : a_i^t = 0$ 
4:  $\mathcal{D}_t = [d]$ 
5: repeat
6:    $\theta = \sqrt{2t \log(2t^2 d / \delta)}$ 
7:    $j = \arg \max_{i \in \mathcal{D}_t} |a_i^t|$ 
8:   for  $i \in \mathcal{D}_t$  do
9:      $c \sim \text{Bernoulli}(\text{sigmoid}(f(x_t)))$ 
10:     $a_i^{t+1} = a_i^t + c \cdot \psi_i(x_t)$ 
11:    if  $|a_i^{t+1}| < |a_j^{t+1}| - \theta$  then
12:       $\mathcal{D}_t \leftarrow \mathcal{D}_t \setminus \{i\}$ 
13:    end if
14:  end for
15:   $t \leftarrow t + 1$ 
16: until Stopping Criteria
17: return  $\mathcal{D}_t$ 
```

Description of Algorithm 1: Let \mathcal{D}_t denote the set of indices for incumbent experts at any given time step t . At the start of the algorithm, all the experts are active, i.e. $\mathcal{D}_t = [d]$. We also maintain a vector a^t of size d in order to keep track of the current cumulative measures for the experts at any time step t , where a_i^t denotes the cumulative measure for expert ψ_i at time t . For any incumbent expert ψ_i , we first draw a random sample $c \sim \text{Bernoulli}(\text{sigmoid}(f(x_t)))$, and then update the cumulative measure a_i^t with the value of the instantaneous measure $c \cdot \psi_i(x_t)$. Intuitively, this cumulative measure corresponds to t times the empirical version of $\mathbb{E}[\text{sigmoid}(f(x_t)) \cdot \psi_i(x_t)]$, and is an indicator of the number of correct predictions made by expert ψ_i until time t .

At any time step t , we first compute the threshold $\theta = \sqrt{2t \log(2t^2 d / \delta)}$ and find the optimal expert ψ_j as the expert with the maximum cumulative measure (in absolute value), i.e. $\forall i \in \mathcal{D}_t : |a_j^t| \geq |a_i^t|$. We then compare the absolute value of the cumulative measure a_i^t with that of the current optimal expert a_j^t ; if it is smaller than the latter by at least θ , we remove the expert from the pool of the incumbent experts. For ECO-G, we use a heuristic extension of this measure, where we replace $c \cdot \psi_i(x_t)$ (which is used in pruning ECO-F) with $c \cdot \text{sign}(\psi_i(x))$.

2 Experiments

Contamination Control Problem: To showcase the capability of pruning to capture higher order interactions, we consider a Boolean problem with known third degree interactions: the contamination problem from [5, 4, 2]. Since $k = 2$, ECO reduces to COMEX [2], where the performance of its vanilla third degree model has been already shown to outperform baselines and match COMBO (at a far lower computational cost albeit with higher number of time steps) for the latter problem. Here, we show that we can further save up on the computational cost of COMEX via pruning experts, and yet obtain a performance very close to that of the vanilla version with the entire experts included.

We start the algorithm with a pool of the entire experts up to degree three monomials, and compare the performance of the algorithm equipped with pruning with those of vanilla second (COMEX-2) and third (COMEX-3) degree models. From Figure 1, we observe that overall the pruning-based algorithm follows the third degree model very closely. As shown in the zoomed-in version of the curves from $t = 1000$ to $t = 4000$, the pruned version is even able to beat the vanilla degree three model until time step 2500, but is eventually slightly outperformed by the latter model. In particular, the pruning method reduces the number of experts from 1562 to 509 (on average), thereby leading to a $\approx 25\%$ saving in amortized computation time until time step 4000. The average computation times for each model are given in Table 1. Notably, continuing this run further up to 6000 steps would lead to a sparse representation of merely 50 monomial experts via the pruning strategy while maintaining a small margin with the vanilla third degree model.

RNA Optimization Problem: We further consider the performance of pruning in the RNA optimization problem over sequences of both small ($n = 30$) and moderate ($n = 60$) lengths, where the latter demonstrates the advantage of using pruning in problems with larger numbers of variables. At $n = 30$ and over 4000 time steps, ECO-F on average reduces the number of experts from 4006 to 913 (averaged over 20 runs), while managing a saving of $\approx 10\%$ in computation

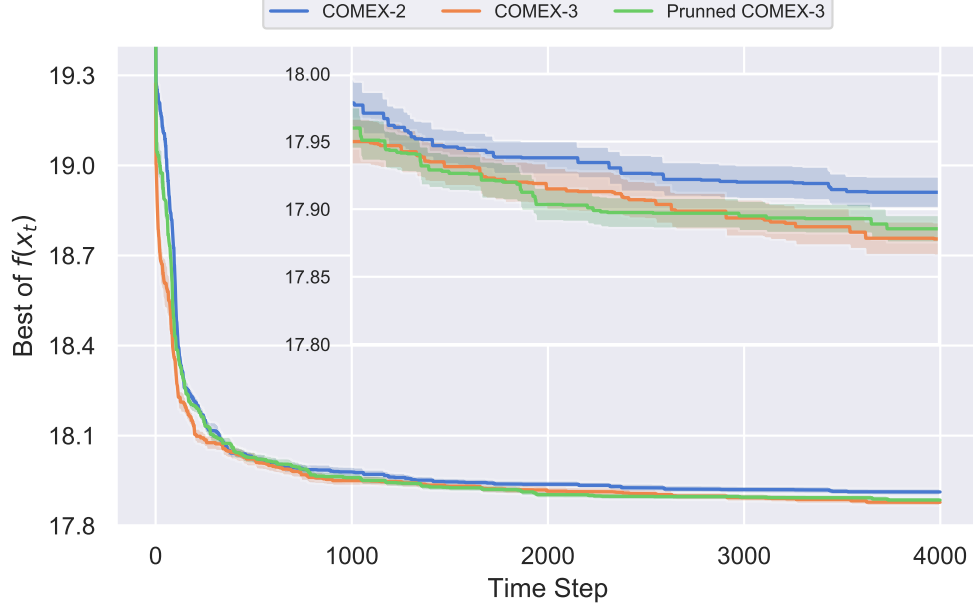


Figure 1: Effect of pruning on the contamination control problem.

Table 1: Average computation time per step (in Seconds) in contamination problem.

n	COMEX-2	COMEX-3	PRUNED COMEX-3
21	0.02	0.17	0.13

time¹. Similarly, ECO-G reduces the number of experts from 8011 to 2736 with a computation time saving of $\approx 14\%$. At $n = 60$ and over 4000 time steps, we observe a similar trend where ECO-F and ECO-G reduce the number of experts from 16111 and 32221 to 5193 and 8869, while offering a competitive performance with the vanilla counterparts. Average computation times for the RNA problem with both $n = 30$ and $n = 60$ are summarized in Table 2

Both ECO-F and ECO-G maintained a large gap with RS and SA at $n = 60$ ($n = 30$), which only managed to reach -29.9 (-18.7) and -58.9 (-23.8) over 4000 steps, respectively. The latter baselines were dropped from the plots in order to avoid clutter. We point out that COMBO was only able to produce ≈ 200 (≈ 500) samples in the RNA problem with $n = 60$ ($n = 30$) within a 24 hour time budget, leading to a poor final average performance of -43.57 (-28.4). As mentioned earlier, the high computational complexity of COMBO (as is the case for any BO algorithm in general) prohibits its use for problems with larger numbers of variables.

Discussion on Pruning Parameter: The value of the parameter δ determines how aggressively the surrogate model experts are pruned via Algorithm 1. If the value of this parameter is too small, it would take a large number of steps in order for the algorithm to prune a sufficient number of experts. On the other hand, if δ is too large, the algorithm would remove too many experts early on before ensuring that such experts are insignificant. This leads to a trade-off between the speed-up rendered by pruning and the performance of the pruning strategy. Nevertheless, due to the logarithmic (as well as square root) dependence of the pruning threshold on δ , the behavior of the algorithm is fairly stable with respect to small to moderate variations in this parameter. We also point out that experimentally, the higher the number of variables n (and thereby experts), a smaller value for the parameter δ is required. Table 3 presents a summary of the number of remaining experts upon pruning after 4000 time steps as well as the values used for the parameter δ in each case.

¹We believe that optimizing the implementation aspects of the code would further boost the improvement in computation times.

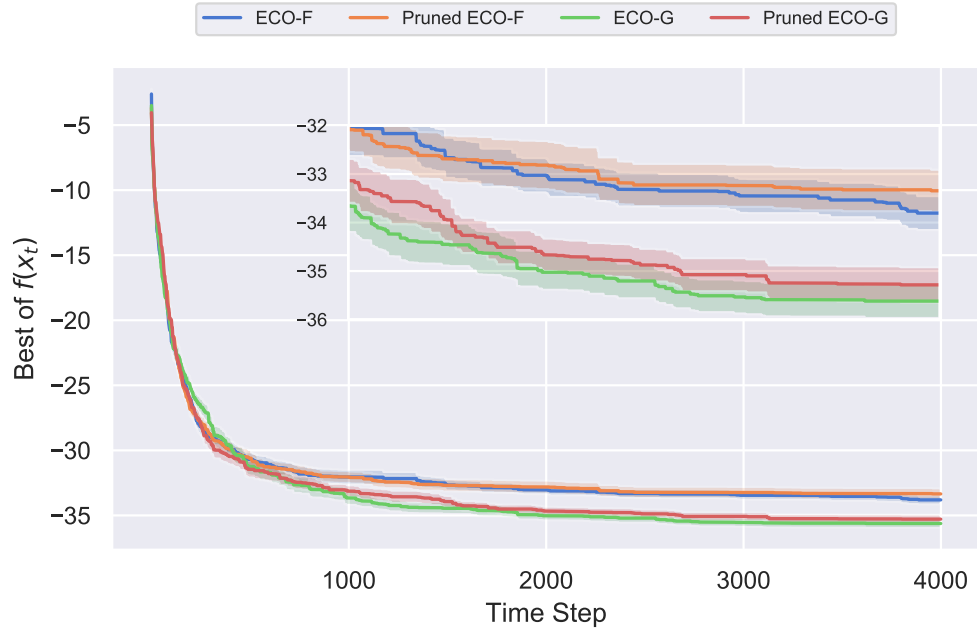


Figure 2: Effect of pruning on the RNA sequence optimization problem with $n = 30$.

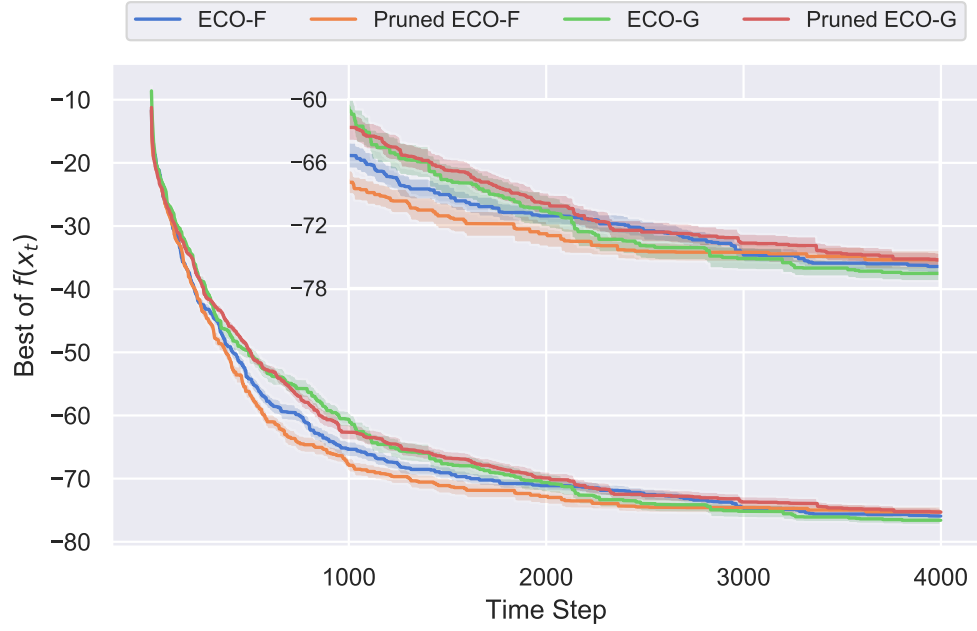


Figure 3: Effect of pruning on the RNA sequence optimization problem with $n = 60$.

Table 2: Average computation time per step (in Seconds) for the RNA problem with pruning.

n	ECO-F	PRUNED ECO-F	ECO-G	PRUNED ECO-G
30	1.97	1.81	5.85	5.04
60	7.8	7.0	24.7	21.5

Table 3: The number of remaining experts. The value of the parameter δ used in pruning experts is reported within parentheses.

n	ECO-F	PRUNED ECO-F (δ)	ECO-G	PRUNED ECO-G (δ)
30	4006	913 (10^{-3})	8011	2736 (10^{-1})
60	16111	5193 (10^{-8})	32221	8869 (10^{-3})

References

- [1] Avrim Blum. Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. In *Machine Learning*, pages 64–72. Morgan Kaufmann, 1997.
- [2] Hamid Dadkhahi, Karthikeyan Shanmugam, Jesus Rios, Payel Das, Samuel Hoffman, Troy David Loeffler, and Subramanian Sankaranarayanan. Combinatorial black-box optimization with expert advice. In *KDD*, 2020.
- [3] Ryan O’Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.
- [4] Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems 32*, pages 2910–2920. Curran Associates, Inc., 2019.
- [5] Matthias Poloczek Ricardo Baptista. Bayesian optimization of combinatorial structures. In *ICML*, 2018.