CBTune: Contextual Bandit Tuning for Logic Synthesis

Fangzhou Liu¹, Zehua Pei², Ziyang Yu¹, Haisheng Zheng², Zhuolum Hel₁,², Tinghuan Chen¹, Bei Yu¹
¹The Chinese University of Hong Kong, Hong Kong, China, ²Shanghai Artificial Intelligence Laboratory, Shanghai, China

Abstract

We propose CBTune, adapting the contextual bandit algorithm to facilitate efficient transformation selection through iterative model tuning.

We implement the Syn-LinUCB algorithm as the agent and establish a context-aware model for informed decision-making in the bandit model.

We present a novel "return-back" mechanism that revisits decisions to avoid local optima, distinguishing it from typical RL scenarios.

Our method surpasses SOTA approaches for metrics and runtime within the same action space.

ML-Enhanced Synthesis Optimization

Machine learning facilitates technology-independent optimization: 1) it models circuit structures to accurately predict performance metrics [4, 2]; 2) it employs reinforcement learning for rapid synthesis flow generation in an exponentially large solution space [1].

Bandit-based Search Model

The Multi-Arm Bandit (MAB) model, known for its efficiency in generating synthesis flows, strikes a balance between exploration and exploitation to optimize rewards. CBTune leverages domain-specific knowledge by integrating contextual data into the MAB model, enabling progressive decision-making depicted in Figure 1.

Context Generator

The vector \( \mathbf{a} \), fused circuit characteristics \( \mathbf{a}^f \) and the arm's long-term payoff \( \mathbf{a}^l \) informs the agent's decisions by providing essential environmental and state insights.

Methodology

Agent: Syn-LinUCB

1. It utilizes short-term payoffs to direct the agent to select arms toward the optimal target value per step, enhancing local performance.
2. It accounts for long-term payoffs to avert local optima and explore potential optimization trends, fostering improved decision quality.

Algorithm 1 Syn-LinUCB

Input: \( \mathbf{a} \in \mathcal{A} \), Context weights \( \mathbf{w} \in \mathbb{R}^p \), Number of iterations \( \tau \), Constant \( \beta \)

Output: Best arm \( \mathbf{a}^* \) in this step.

1. \( \mathbf{a}^* = \) Reward of all arms;
2. Extract the AIG characteristics \( \mathbf{x}_i \in \mathbb{R}^d \);
3. Arm selection times \( \mathbf{x}_i \in \mathbb{R}^d \);
4. For \( i = 1, 2, \ldots, \tau \) do:
5. Update the long-term payoff \( \mathbf{a}^l \in \mathbb{R}^d \);
6. Observe features of \( \mathbf{x} = x_1, x_2, \ldots, x_d \in \mathbb{R}^d \);
7. For \( i \neq a \), do:
8. Initialize historical context and reward by \( \mathbf{A}_t = 1, \mathbf{b}_0 = 0, \mathbf{w}_0 \) is new;
9. Update hyperparameter \( \mathbf{a}^l \) by \( \alpha = 1.8 \mathbf{a}^l \in \mathbb{R}^d \);
10. Update the decision parameter by \( \mathbf{b}_t = \mathbf{A}_t \mathbf{b}_t \);
11. Calculate the weighted context \( \mathbf{a}^w = \mathbf{a}^w \mathbf{b}_t \);
12. Update score by \( \mathbf{b}_t = \mathbf{b}_t + \alpha \mathbf{b}_t \);
13. Choose arm by \( \mathbf{a}_t = \arg \max \mathbf{b}_t \);
14. Increase the search context of arm by \( \mathbf{a}_t = \mathbf{a}_t + 1 \);
15. Update the arm selection \( \mathbf{A}_t \) and \( \mathbf{b}_0 \) of the chosen arm \( \mathbf{a}_t \);
16. end for

Conclusion

CBTune outperforms FlowTune in both AIG nodes/6-LUT optimization in both metric and runtime. Our method also outshines three RL-based methods by reducing 6-LUT counts up to 4.4%, all achieved in a swift 8.7 minutes per design.

CBTune efficiently generates synthesis flows with excellent, stable results and fast runtime, without training data or complex procedures.

References