Correlated Multi-objective Multi-fidelity Optimization for HLS Directives Design

Qi Sun\textsuperscript{1}, Tinghuan Chen\textsuperscript{1}, Siting Liu\textsuperscript{1}, Jin Miao\textsuperscript{2}, Jianli Chen\textsuperscript{3}, Hao Yu\textsuperscript{4}, Bei Yu\textsuperscript{1}

\textsuperscript{1}The Chinese University of Hong Kong
\textsuperscript{2}Synopsys
\textsuperscript{3}Fudan University
\textsuperscript{4}SUSTech
Background

High-level synthesis (HLS)

- Translate high-level programming languages (e.g., C/C++) to low-level hardware description languages (HDLs).
- Under the guidance of the HLS directives (pragmas).
- Same high-level descriptions, different HLS directives → different hardware implementations.
- For each application, a group of HLS directives is represented as a configuration vector $x$.

```pseudo
comp(int in[10], int out[10]):
    #pragma HLS INLINE={ON, OFF}
    for (i = 0; i < 10; i ++) {
        #pragma HLS UNROLL factor={2,5,10}
        in[i] = out[i];
    }
```

Pseudo-codes and HLS directives. The directives are in red. Each directive has some factors, e.g., 2, 5, and 10.
Background

Various types of directives

- **Function**
  - dataflow, inline

- **Loop**
  - unroll, pipeline

- **Resource**
  - DSP, BRAM

- **Array Partition**
  - block, cyclic, complete

Design flow

C/C++

HLS Directives

- **HLS**
  - High Level Synthesis Stage
  - Logic Synthesis Stage
  - Implementation Stage

Post-HLS Reports

- Post-Synth Reports
- Post-Impl Reports

Longer running times, more accurate reports

Multiple conflicting design objectives (three fidelities)

- delay, power consumption, and resource consumption
**Background**

Various types of directives

**HLS Directives**
- **Function**
  - dataflow, inline
- **Loop**
  - unroll, pipeline
- **Resource**
  - DSP, BRAM
- **Array Partition**
  - block, cyclic, complete

**Design flow**

- HLS Directives
- C/C++
- FPGA Design Tool

**High Level Synthesis Stage**
- Post-HLS Reports

**Logic Synthesis Stage**
- Post-Synth Reports

**Implementation Stage**
- Post-Impl Reports

**Longer running times, more accurate reports**
Background

Various types of directives

- **Function**
  - dataflow, inline

- **Loop**
  - unroll, pipeline

- **Resource**
  - DSP, BRAM

- **Array Partition**
  - block, cyclic, complete

Design flow

- **HLS Directives** -> **C/C++** -> **FPGA Design Tool**

  - **High Level Synthesis Stage**
  - **Logic Synthesis Stage**
  - **Implementation Stage**

  - Post-HLS Reports
  - Post-Synth Reports
  - Post-Impl Reports

Multiple conflicting design objectives (three fidelities)

- delay, power consumption, and resource consumption

Longer running times, more accurate reports
Background

Pareto optimality – find some Pareto-optimal points

- 3 objective functions. \( f_m : \mathcal{X} \rightarrow \mathbb{R} \), for \( m = 1, 2, 3 \).
Background

Pareto optimality – find some Pareto-optimal points

- 3 objective functions. $f_m : \mathcal{X} \rightarrow \mathbb{R}$, for $m = 1, 2, 3$.

- A value point $y = [f_1(x), f_2(x), f_3(x)]$, in the value space $\mathcal{Y}$.
Background

Pareto optimality – find some Pareto-optimal points

- 3 objective functions. $f_m : \mathcal{X} \rightarrow \mathbb{R}$, for $m = 1, 2, 3$.

- A value point $y = [f_1(x), f_2(x), f_3(x)]$, in the value space $\mathcal{Y}$.

- For $y_i, y_j \in \mathcal{Y}$, $y_i$ dominates $y_j$ when $y_{i,m} \geq y_{j,m}$, for $\forall m \in \{1, 2, 3\}$, represented as $y_i \succeq y_j$. 
Background

Pareto optimality – find some Pareto-optimal points

- 3 objective functions. $f_m : \mathcal{X} \rightarrow \mathbb{R}$, for $m = 1, 2, 3$.

- A value point $y = [f_1(x), f_2(x), f_3(x)]$, in the value space $\mathcal{Y}$.

- For $y_i, y_j \in \mathcal{Y}$, $y_i$ dominates $y_j$ when $y_{i,m} \geq y_{j,m}$, for $\forall m \in \{1, 2, 3\}$, represented as $y_i \succeq y_j$.

- The non-dominated points are called Pareto-Optimal Set, $\mathcal{P}(\mathcal{Y}) \in \mathcal{Y}$.
Background

Pareto optimality – find some Pareto-optimal points

- 3 objective functions. \( f_m : \mathcal{X} \rightarrow \mathbb{R} \), for \( m = 1, 2, 3 \).

- A value point \( y = [f_1(x), f_2(x), f_3(x)] \), in the value space \( \mathcal{Y} \).

- For \( y_i, y_j \in \mathcal{Y} \), \( y_i \) dominates \( y_j \) when \( y_{i,m} \geq y_{j,m} \), for \( \forall m \in \{1, 2, 3\} \), represented as \( y_i \succeq y_j \).

- The non-dominated points are called Pareto-Optimal Set, \( \mathcal{P}(\mathcal{Y}) \in \mathcal{Y} \).

- Blank cells are dominated

- Pareto hyper-volume \( PV_{v_{ref}}(\mathcal{P}(\mathcal{Y})) \).
Background

Target

- Find the **Pareto-optimal** points in HLS design problem
Background

Target

- Find the **Pareto-optimal** points in HLS design problem

Challenges

- Hard to predict the performance values according to the directives
- Hard to characterize the complicated relationships between the multiple objectives
- Hard to balance the consumption of running time and accuracy of results
Background

Target

- Find the **Pareto-optimal** points in HLS design problem

Challenges

- Hard to predict the performance values according to the directives
- Hard to characterize the complicated relationships between the multiple objectives
- Hard to balance the consumption of running time and accuracy of results

Requirements

- Develop a **flexible and general** method
- Strike a **balance** between optimization workloads and accuracy of results
- Able to characterize the complicated relationships between the **HLS directives and multiple objectives**
## Our Solution

### Optimization strategy
- Bayesian optimization
- Acquisition function: expected improvement

### Multi-fidelity model
- Non-linear Gaussian process model

### Multi-objective model
- Pareto learning
- Correlated Gaussian process model
Multi-Fidelity Model

**Traditional linear correlation model**

\[ f_{m}^{h}(x) = \rho^{h} \times f_{m}^{l}(x) + f_{m}^{e}(x). \]

- \( \rho^{h} \): a scaling factor. \( f_{m}^{e}(x) \): error term.

**Our non-linear correlation model**

The reports of the low fidelity are concatenated as part of the inputs to the next high fidelity.

\[ f_{m}^{h}(x) = z_{m}^{h}(f_{m}^{l}(x), x) + f_{m}^{e}(x). \]

- \( z_{m}^{h}(\cdot) \): correlation term, modelled by a GP model.
Multi-Objective Model – Pareto Learning

Acquisition function: expected improvement of Pareto hyper-volume

- At step $t + 1$ of Bayesian optimization, we already have data set $D = \{x_s, y_s\}_{s=1}^t$, with $\mathcal{P}(\mathcal{Y}) = \{y_s\}_{s=1}^t$. Sample a new point $x_{t+1}$, the predicted value is $y(x_{t+1})$.

$$EIPV(x_{t+1}|D) = \mathbb{E}_{p(y(x_{t+1})|D)} \left[ PV_{v_{ref}} \left( \mathcal{P}(\mathcal{Y} \cup y(x_{t+1})) \right) - PV_{v_{ref}} \left( \mathcal{P}(\mathcal{Y}) \right) \right].$$
At step $t + 1$ of Bayesian optimization, we already have data set $D = \{x_s, y_s\}_{s=1}^t$, with $\mathcal{P}(\mathcal{Y}) = \{y_s\}_{s=1}^t$. Sample a new point $x_{t+1}$, the predicted value is $y(x_{t+1})$.

$$EIPV(x_{t+1}|D) = \mathbb{E}_{p(y(x_{t+1})|D)} \left[ PV_{\text{ref}} (\mathcal{P}(\mathcal{Y} \cup y(x_{t+1}))) - PV_{\text{ref}} (\mathcal{P}(\mathcal{Y})) \right].$$
Combined Model

▶ Two dimensions: one for the multi-objective functions, one for the multi-fidelities.
▶ Augment acquisition function:

\[
\text{PEIPV}_i(x_{t+1} | D) = \text{EIPV}_i(x_{t+1} | D) \cdot \frac{T_{\text{impl}}}{T_i}, \ i \in \{\text{hls, syn, impl}\},
\]

\[
\max_i \text{PEIPV}_i, \ i \in \{\text{hls, syn, impl}\}
\]

▶ Select the largest one, and run the compilation flow to that fidelity.
Experiments and Results

Experimental settings

- 5 traditional benchmarks, 1 DNN benchmark
- All HLS code are compiled via Vivado HLS to get the reports (for validation of results of various algorithms).

Quality metric – average distance to reference set (ADRS)

- $\Gamma$ reference set (real Pareto set).
- $\Omega$ learned Pareto set.

$$ADRS(\Gamma, \Omega) = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \min_{\omega \in \Omega} f(\gamma, \omega)$$
Results

All algorithms use the same input features.

- Bayesian methods: 8 initial samples, at most 40 optimization steps.
- Other methods, each training set has 48 points.

### Table: Normalized Experimental Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Normalized ADRS</th>
<th>Normalized Standard Deviation of ADRS</th>
<th>Normalized Overall Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ours</td>
<td>FPL18</td>
<td>ANN</td>
</tr>
<tr>
<td>GEMM</td>
<td>0.27</td>
<td>0.50</td>
<td>1.00</td>
</tr>
<tr>
<td>iSmart2</td>
<td>0.65</td>
<td>0.68</td>
<td>1.00</td>
</tr>
<tr>
<td>SORT_RADIX</td>
<td>0.64</td>
<td>0.72</td>
<td>1.00</td>
</tr>
<tr>
<td>SPMV_ELLPACK</td>
<td>0.19</td>
<td>0.47</td>
<td>1.00</td>
</tr>
<tr>
<td>SPMV_CRS</td>
<td>0.22</td>
<td>0.29</td>
<td>1.00</td>
</tr>
<tr>
<td>STENCIL3D</td>
<td>0.39</td>
<td>0.41</td>
<td>1.00</td>
</tr>
<tr>
<td>Average</td>
<td>0.39</td>
<td>0.51</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Example – GEMM

Directives

- INLINE, PIPELINE, UNROLL, Mul_LUT, DSP48, ARRAY_PARTITION, BRAM.

---

**REFERENCES**

Thank you!