Stochastic Search for Signal Processing Algorithm Optimization

Bryan Singer and Manuela Veloso
Computer Science Department
Carnegie Mellon University, Pittsburgh

Liat Ben-Haim, November 16th 2011
Overview

- Genetic algorithm
- Signal processing
  - Walsh-Hadamard Transform
- **STEER: Split Tree Evolution for Efficient Runtimes**
  - For Walsh-Hadamard Transform
  - Results Walsh-Hadamard Transform
  - For Arbitrary Transform
  - Results Arbitrary Transform
- **Conclusion: Strengths and Weaknesses**
Genetic Algorithm

Complete Search space (very large)
Genetic Algorithm

- Base Population
- Repeat
- Survival of the fittest
- Crossover (only parts of the population)
- Mutation (only parts of the population)
- Population consists of algorithms
  - Algorithms modeled as trees
- Fitness is measured in runtime on given device
- Genetic algorithm

- **Signal processing**
  - Walsh-Hadamard Transform

- **STEER: Split Tree Evolution for Efficient Runtimes**
  - For Walsh-Hadamard Transform
  - Results Walsh-Hadamard Transform
  - For Arbitrary Transform
  - Results Arbitrary Transform

- Conclusion: Strengths and Weaknesses
Signal Processing: \( y = Ax \)
Walsh-Hadamard Transform (WHT)

\[ \begin{align*}
\text{\( y = WHT(2^n) \cdot x \)} & \quad \text{\( WHT(2^n) = \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \otimes \cdots \otimes \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \text{\( n \) factors} \)} \\
\text{\( n = 1: \)} & \quad \text{\( y = \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \cdot x \)} \\
\text{\( n = 2: \)} & \quad \text{\( y = \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \otimes \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \cdot x = \left[ \begin{array}{cc|cc}
1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1
\end{array} \right] \cdot x \)} \\
\text{\( n = 3: \)} & \quad \text{\( y = \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \otimes \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \otimes \left[ \begin{array}{cc}
1 & 1 \\
1 & -1
\end{array} \right] \cdot x = \left[ \begin{array}{ccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1
\end{array} \right] \cdot x \)}
\end{align*} \]
Fast Walsh-Hadamard Transform

\[ n = 2: \quad \text{Opcount: 12} \]

\[ y = WHT(2^2) \cdot x = [I_{2^1} \otimes WHT(2^1)] \cdot [WHT(2^1) \otimes I_{2^1}] \cdot x = \]

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & -1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & -1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & -1 & 0 \\
0 & 1 & 0 & -1 \\
\end{bmatrix}
\cdot x
\]

\[ \text{Opcount: 4} \]

\[ \text{Opcount: 4+4 = 8} \]
Fast Walsh-Hadamard Transform

\[ n = 3: \]

\[
\begin{align*}
\text{Opcount:} & \quad 24 & \quad 24 & \quad 24 \\
\text{Opcount WHT}(2^3): & \quad 56
\end{align*}
\]
General Break Down Rule

\[ WHT(2^n) = \prod_{i=1}^{t} (I_{2^{n_1+\cdots+n_{i-1}}} \otimes WHT(2^{n_i}) \otimes I_{2^{n_{i+1}+\cdots+n_t}}) \]

\[ n = n_1 + \cdots + n_t \quad (n_j: \text{positive integers}) \]
WHT Example

$$WHT(2^n) = \prod_{i=1}^{t} (I_{2^{n_1+\ldots+n_{i-1}}} \otimes WHT(2^{n_i}) \otimes I_{2^{n_{i+1}+\ldots+n_t}})$$

$$WHT(2^5)$$

$$= [WHT(2^3) \otimes I_{2^2}][I_{2^3} \otimes WHT(2^2)]$$

$$= [\{(WHT(2^1) \otimes I_{2^2})(I_{2^1} \otimes WHT(2^2))\} \otimes I_{2^2}]$$

$$[I_{2^3} \otimes \{(WHT(2^1) \otimes I_{2^1})(I_{2^1} \otimes WHT(2^1))\}]$$

Stochastic Search for Signal Processing Algorithm Optimization

$$I_{2^0} \otimes A = A$$

$$A \otimes I_{2^0} = A$$
- Genetic algorithm
- Signal processing
  - Walsh-Hadamard Transform
- **STEER: Split Tree Evolution for Efficient Runtimes**
  - For Walsh-Hadamard Transform
  - Results Walsh-Hadamard Transform
  - For Arbitrary Transform
  - Results Arbitrary Transform
- Conclusion: Strengths and Weaknesses
Goal

Given input signal $x$ of size $2^n$ and specific device, find fastest program for this signal size and device

Search Techniques

- **Exhaustive Search**
  - Does not scale

- **Dynamic Programming**
  - Assumption: «combination of optimal solutions for subproblems leads to optimal solution»
  - K-Best DP
  - Search space restriction: Binary trees
  - Bad choices lead to inferior solution

- **Split Tree Evolution for Efficient Runtimes: STEER**
STEER

- Split Tree Evolution for Efficient Runtimes
  - Genetic Algorithm
  - Part of the SPIRAL research group
    - «Can we teach computers to write fast libraries?»
  - Adapted to the system used by the research group
STEER: Genetic Algorithm

Base Population

Repeat

Survival of the fittest (runtime)

Mutation

Crossover
WHT: Crossover

Stochastic Search for Signal Processing Algorithm Optimization
WHT: Mutation

Original

Flip

Grow

Truncate

Up

Down

Join

Split

Stochastic Search for Signal Processing Algorithm Optimization
Results: WHT

- Pentium III 450 MHz (32-Bit Architecture)
- Linux 2.2.5-15
- WHT Package from Johnson and Püschel
  - «In search of the optimal Walsh-Hadamard transform», 2000
  - Leaves of sizes $2^1$ to $2^8$
  - Unrolled straight-line code
Stochastic Search for Signal Processing Algorithm Optimization
Arbitrary Signal Transform

- **Random tree generation:**
  - Randomly choose an applicable break down rule
  - Apply to node, to generate a random set of children
  - Recursively apply to each child

- **Crossover**
  - Equivalent nodes: same size and transform

- **New mutations**
Arbitrary Transform: Mutation

Stochastic Search for Signal Processing Algorithm Optimization
Results: Arbitrary Transform

Stochastic Search for Signal Processing Algorithm Optimization
Stochastic Search for Signal Processing Algorithm Optimization
Strengths & Weaknesses

- STEER can be used for arbitrary transforms using SPIRAL
- Finds good if not necessarily optimal solutions
- Found solutions are generally better than DP
- Times significantly less formulas than exhaustive search
- Missing parameters of the evolutionary algorithm
- No guarantee for a «good» solution
- Times more formulas than DP
- No mention of how long STEER usually runs
- How much better than Ax?
Questions?