Edge Separability Based Circuit Clustering with Application to Circuit Partitioning

Jason Cong and Sung Kyu Lim
UCLA VLSI CAD Lab

Support : MARCO/DARPA GSRC
and Fujitsu Laboratories of America under CA MICRO
Outline

• Preliminaries
  – Problem formulation
  – Previous works
  – Motivation and contribution

• ESC algorithm
  – Overview
  – Two-phase framework

• Experimental results

• Conclusions and ongoing works
Problem Formulation

• Clustering vs partitioning
  – Divide cells into groups under area constraint $A$
  – Clustering if $A$ is small; partitioning otherwise
  – Clustering = pre-process of partitioning

• Clustering Metrics
  – Absorption, Density, Rent Parameter, Ratio Cut, Closeness, Connectivity, etc....

• Partitioning Metrics
  – Cutsizwe and delay
Previous Works

• Clustering algorithms
  – Cutsize oriented vs delay oriented

• Bottom-up clustering
  – Connectivity based: Absorption [SS93], Density [CS93, HK95], Rent Parameter [NOP87], Closeness [SK93], Connectivity [SU72], etc…
  – Signal flow based: MFFC [CD93], MFFS [CL+97], etc…

• Top-down clustering
  – Ratio Cut [WC92]
Motivation

- Existing algorithms rely on local connectivity $w(e)$

$$DEN(C_1) = \frac{\sum_{e \in C_1} w(e)}{\sum_{v \in C_1} s(v)} = \frac{w(e_3) + w(e_4) + w(e_5)}{s(v_1) + s(v_2) + s(v_3)}$$
Motivation

• Existing algorithms rely on local connectivity $w(e)$

$$CLO(C_1, C_2) = \frac{w(e)}{\min\{c(C_1), c(C_2)\}} = \frac{w(e_6)}{w(e_6) + w(e_{10})}$$
Contribution

• Proposed global connectivity information $\lambda(e)$
• Efficient way to estimate $\lambda(e)$
• Demonstrate effectiveness of $\lambda(e)$
  – compared to existing algorithms that use $w(e)$
  – two-level ESC vs multi-level hMetis [KA+97, KK99]
Edge Separability

- Edge Separability \( \lambda(e) \)
  - Minimum cutsize that separates \( x \) and \( y \), where \( e=(x, y) \)
  - Beyond local connectivity \( w(e) \)
  - Search on entire circuit: max-flow in \( O(n^2 \log n) \)

\[
\begin{align*}
x & \quad w(e) \quad y \\
\end{align*}
\]

\( w(e) \leq \lambda(e) \)
Edge Separability

- CAPFOREST [Nagamochi and Ibaraki, 1992]
  - $O(n\log n)$ edge labeling algorithm
  - Identify *contractible* edges
  - Label each edge with $q(e)$
  - Repeat $O(n)$ to find global mincut

we show that

$w(e) \leq q(e) \leq \lambda(e)$
CAPFOREST Algorithm

• Illustration

visit nodes in Maximum Adjacency ordering
CAPFOREST Algorithm

• Illustration

edge labels denote $w(e) / q(e) / \lambda(e)$

following holds $w(e) \leq q(e) \leq \lambda(e)$
ESC Clustering Algorithm

• Classification
  – Graph connectivity based $O(n \log n)$ algorithm
  – Two-level cluster hierarchy

• Implementation
  – Transform netlist into $G$
  – Perform CAPFOREST to label all edges with $q(e)$
  – Sort contractible edges based on rank $r(e)$
    \[ r(e) = \frac{q(e)}{\min\{c(x), c(y)\}} \]
  – Grow clusters from contraction of edges in the sorted order
ESC Clustering Algorithm

- Illustration (size limit = 3)

\[
\begin{align*}
\text{edge labels denote} & \qquad \frac{w(e)}{q(e)} / \frac{q(e)}{r(e)} \\
\text{where} & \qquad r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}
\end{align*}
\]
ESC Clustering Algorithm

- Illustration (size limit = 3)

edge labels denote $w(e) / q(e) / r(e)$

$$r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}$$
ESC Clustering Algorithm

- Illustration (size limit = 3)

```
edge labels denote
w(e) / q(e) / r(e)

r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}
```
ESC Clustering Algorithm

• Illustration (size limit = 3)

edge labels denote $w(e) / q(e) / r(e)$

\[
r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}
\]
ESC Clustering Algorithm

- Illustration (size limit = 3)

Edge labels denote $w(e) / q(e) / r(e)$

$$r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}$$
ESC Clustering Algorithm

- Illustration (size limit = 3)

\[
\begin{align*}
    w(e) &\mid q(e) \mid r(e) \\
    r(e) &= \frac{q(e)}{\min\{c(x), c(y)\}}
\end{align*}
\]
ESC Clustering Algorithm

- Illustration (size limit = 3)

edge labels denote \( w(e) / q(e) / r(e) \)

\[
r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}
\]
ESC Clustering Algorithm

- Illustration (size limit = 3)

edge labels denote $w(e) / q(e) / r(e)$

$$r(e) = \frac{q(e)}{\min\{c(x), c(y)\}}$$
Two-Phase Framework

• Edge contraction based bottom-up clustering
  – Sort edges based on contraction cost
  – Grow clusters from edge contraction

• Top-down partitioning
  – Partition top-level
  – Project cutline
  – Partition bottom-level
Existing Algorithms

- Absorption [SS93] (max) : weight of edges absorbed into $C$
  \[ ABS(C) = \sum_{e \in C} w(e) \]

- Density [CS93, HK95] (max) : density of $C$ in terms of $w(e)$
  \[ DEN(C) = \frac{ABS(C)}{s(C)}, \quad s(C) = \text{size of } C \]

- Rent Parameter [NOP87] (min) : entail better placement result
  \[ REP(C) = \frac{\ln c(C) - \ln d(C)}{\ln s(C)}, \quad d(C) = \frac{1}{|C|} \sum_{x \in C} c(x) \]
Existing Algorithms (cont)

• Ratio Cut [WC92] (min) : identify natural clusters
  
  \[ RTC(C) = \frac{c(C)}{s(C)}, \quad c(C) = \sum_{e \text{ incident to } C} w(e) \]

• Closeness [SK93] (max) : connectivity to neighboring vertices
  
  \[ CLO(C) = \frac{w(e)}{\min\{c(x), c(y)\}} - \gamma \frac{s(x) + s(y)}{\text{ave}_\text{size}} \]

• Connectivity [SU72] (max) : connectivity to neighboring vertices
  
  \[ CON(C) = \frac{w(e)}{s(x)[c(x) - w(e)]} + \frac{w(e)}{s(y)[c(y) - w(e)]} \]
Experimental Result

- LR [CL+97] bipartitioning on ISPD98 [Alp98] circuits
Experimental Result

- LR/ESC bipartitioning algorithm
  - Cluster netlist with ESC algorithm
  - Partition top-level with LR [CL+97] algorithm
  - Project cutline
  - Refine bottom-level with LR algorithm

- LR/ESC-PM multiway partitioning algorithm
  - Obtain matching of blocks for Pair-wise Movement [CL98]
  - Apply LR/ESC bipartitioning on block-pairs
Experimental Result

- LR-ESC/PM: two-level
- hMetis[KA+97] and hMetis-Kway[KK99]: multi-level
Conclusions & Ongoing Works

- **Edge separability** $\lambda(e)$
  - Beyond local connectivity $w(e)$
  - Tighter lower bound of $\lambda(e)$ in $O(n \log n)$ for all edges

- **ESC clustering algorithm**
  - Connectivity based two-level bottom-up clustering
  - Outperforms other existing algorithms
  - Comparable to hMetis [KA+97] and hMetis-Kway [KK99]

- **Ongoing work**
  - Multi-level ESC
  - Impact on placement
  - www.cadlab.ucla.edu