Today

- Problem
- Brute-Force/Exhaustive
- Greedy
- Estimators
- Analytical Provisioning
- ILP Schedule and Provision

Previously

- General formulation for scheduled operator sharing
  - VLIW
- Fast algorithms for scheduling onto fixed resource set
  - List Scheduling
- More extensive algorithms for time-constrained
  - Force Directed, Branch-and-Bound

Today

- How do we determine the set of resources?

Today: Provisioning

- Given
  - An area budget
  - A graph to schedule
  - A library of operators
- Determine:
  - Delay minimizing set of operators
  - Or delay-achieving set of operators
  - i.e. select the operator set
Exhaustive

1. Identify all area-feasible operator sets
   - E.g. preclass exercise
2. Schedule for each
3. Select best
   • → optimal
   • Drawbacks?

Exhaustive

- How large is space of feasible operator sets?
  - As function of
    • operator types – O
    • Types: add, multiply, divide, ....
    • Maximum number of operators of type m
      \[ m^O \]

Implication

- Feasible operator space can be too large to explore exhaustively

Greedy Incremental

- Start with one of each operator
- While (there is area to hold an operator)
  - Which single operator
    • Can be added without exceeding area limit?
    • Schedule (maybe list-schedule?)
    • Calculate benefit (maybe \( \Delta T / \Delta A \)?)
    • Pick largest benefit
  - Add one operator of that type
- How long does this run?
  - \( T_{\text{schedule}}(E)^* O(\text{operator-types} \times A) \)

Greedy Incremental

- Work Preclass with greedy incremental
  - For each step
    • half class evaluate each candidate resource

Greedy Incremental

- Start with one of each operator
- While (there is area to hold an operator)
  - Which single operator
    • Can be added without exceeding area limit?
    • Schedule (maybe list-schedule?)
    • Calculate benefit (maybe \( \Delta T / \Delta A \)?)
    • Pick largest benefit
  - Add one operator of that type
- Weakness?
Example

Find best 5 operator solution.

One of each.

<table>
<thead>
<tr>
<th>Sq</th>
<th>Dia</th>
<th>Circ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>I</td>
</tr>
<tr>
<td>D</td>
<td>G</td>
<td>J</td>
</tr>
<tr>
<td>H</td>
<td>J</td>
<td></td>
</tr>
</tbody>
</table>

Example

Find best 5 operator solution.

Two Squares

<table>
<thead>
<tr>
<th>Sq</th>
<th>Dia</th>
<th>Circ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>E</td>
<td>C,D</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>D</td>
<td>I</td>
<td>H</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>J</td>
</tr>
</tbody>
</table>

Example

Find best 5 operator solution.

Two Diamonds

<table>
<thead>
<tr>
<th>Sq</th>
<th>Dia</th>
<th>Circ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>E</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>D</td>
<td>I</td>
<td>H</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>J</td>
</tr>
</tbody>
</table>

Example

Find best 5 operator solution.

Two Circles

<table>
<thead>
<tr>
<th>Sq</th>
<th>Dia</th>
<th>Circ</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>E</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>D</td>
<td>I</td>
<td>H</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>J</td>
</tr>
</tbody>
</table>
Example

Which should greedy add?

Find best 5 operator solution.

Incremental addition does not accelerate.

Example

Two sqs + Two diamonds

Find best 5 operator solution.

Max effect: Incremental may not suggest next single addition.

Analytic Formulation

Challenge

- Scheduling expensive
  - $O(|E|)$ or $O(|E|^* \log(|V|))$ using list-schedule
- Results not analytic
  - Cannot write an equation around them
- Bounds are sometimes useful
  - No precedence $\rightarrow$ is resource bound
  - Often one bound dominates
    - Latency bound unaffected by operator count

Estimations

- Step 1: estimate with resource bound
  - $O(|E|)$ vs. $O(|V|)$ evaluation
- Step 2: use estimate in equations
  - $T = \max(N_1/M_1, N_2/M_2, \ldots)$
- Most useful when $\text{RB} >> \text{CP}$

Constraints

- Let $A_i$ be area of operator type $i$
- Let $M_i$ be number of operators of type $i$

$$\sum A_i \times M_i \leq \text{Area}$$

(start summary of variables on board)
Achieve Time Target

- Want to achieve a schedule in $T$ cycles
- What constraint equation does that imply? (what property must hold?)
- Each resource bound must be less than $T$ cycles:
  - $N_i/M_i \leq T$

Algebraic Solve

- Set of equations
  - $N_i/M_i \leq T$
  - $\sum A_i M_i \leq \text{Area}$
- Assume equality for time bound
  - $N_i/M_i = T \Rightarrow M_i = N_i/T$

$$\sum A_i \times N_i \leq \text{Area}$$

Rearranging

$$\frac{\sum A_i \times N_i}{T} \leq \text{Area}$$

$$\frac{\sum A_i \times N_i}{\text{Area}} \leq T$$

Bounding $T$

- Gives Lower Bound on $T$

$$\frac{\sum A_i \times N_i}{\text{Area}} \leq T$$

Intuition: $N$ of each is right balance given unbounded area; Scale to area available.

Preclass

- What is $T_{\text{lower}}$ for preclass?

$$\frac{\sum A_i \times N_i}{\text{Area}} \leq T$$

$$T \geq \frac{1 \times 8 + 2 \times 4}{7} = \frac{16}{7} = 2.3 \quad T \geq 3$$

Back Substitute from $T$ to $x$

- $M_i = N_i/T$

$$\frac{\sum A_i \times N_i}{\text{Area}} \leq T$$

- $M_i$ won’t necessarily be integer
  - Round down definitely feasible solution
  - May have room to move a few up by 1
- Reduces range may need to search
  - (just over the residual area once rounded down)
Preclass

- \( M = N / T \)
- \( T \geq 3 \)
- \( M_{\text{add}}, M_{\text{mpy}} ? \)
- \( M_{\text{add}} = 8/3 \Rightarrow 2 \text{ or } 3 \)
- \( M_{\text{mpy}} = 4/3 \Rightarrow 1 \text{ or } 2 \)

Counter Example

- 1 Unit each
- Area = 4 Units
- What would analytic predict?
- What is best?
- How does CP compare to RB?
- Analytic Resource Estimate
  - Most useful when RB >> CP

Analytic Counter Example

- How would greedy incremental work on this one?

ILP

Integer Linear Programming

- Formulate set of linear equation constraints (inequalities)
  - \( Ax_0 + Bx_1 + Cx_2 \leq D \)
  - \( x_0 + x_1 = 1 \)
  - \( A, B, C, D \) – constants
  - \( x_i \) – variables to satisfy
  - No products on variables, just linear weighted sums
  - Can constrain variables to integers
  - No polynomial time guarantee
    – But often practical
    – Solvers exist (significant piece on April 1 (seriously))

ILP Provision and Schedule

Now to make it look like an ILP nail...
- Formulate operator selection and scheduling as ILP problem
Formulation

• Integer variables $M_i$ - number of operators of type $i$
• 0-1 (binary) variables $x_{i,j}$ - 1 if node $i$ is scheduled into timestep $j$ - 0 otherwise
• Variable assignment completely specifies operator selection and schedule
• This formulation for achieving a target time $T$ (time constrained) - $j$ ranges 0 to $T-1$

Target $T \rightarrow \text{Min } T$

• Formulation targets $T$
• What if we don’t know $T$?
  - Want to minimize $T$?
• Do binary search for minimum $T$
  - How does that impact solution time?

Constraints

What properties must hold true for a solution to be valid?

1. Total area constraints
2. Not assign too many things to a timestep
3. Assign every node to some timestep
4. Maintain precedence

(1) Total Area

• Same as before

\[ \sum A_i \times M_i \leq \text{Area} \]

(2) Not overload timestep

• For each timestep $j$
  - For each operator type $k$

\[ \sum_{o_i \in F} U_k \leq M_k \]

(3) Node is scheduled

• For each node in graph

\[ \sum_j x_{i,j} = 1 \]

Can narrow to sum over slack window.
(4) Precedence Holds

- For each edge from node src to node snk
  \[
  \sum_j j \times x_{src,j} - \sum_j j \times x_{snk,j} \leq -1
  \]

Can narrow to sum over slack windows.

Example (Time Permitting)

- What are the ILP equations for the preclass example?
  1. Total area constraints
  2. Not assign too many things to a timestep
  3. Assign every node to some timestep
  4. Maintain precedence

Constraints

1. Total area constraints
2. Not assign too many things to a timestep
3. Assign every node to some timestep
4. Maintain precedence

ILP Solver

- ILP Solver can take these constraints and find a solution (satisfying assignment)
  - On Wednesday, will see how to start to make this practical

Round up Algorithms and Runtimes

- Exhaustive Schedule
- Greedy Schedule
- Analytic Estimates
- ILP formulation

Big Ideas:

- Estimators
- Dominating Effects
- Reformulate as a problem we already have a solution for
  - ILP
- Technique: Greedy
- Technique: ILP
Admin

- Assignment 5 Thursday
- No class on Monday
  - Will have class on Wednesday
- No assignment 6 supplement
  - Focus on project and writeup
- Reading for Wednesday online