HW/SW Codesign

Exercise 8: Design Space Exploration
(Pareto optimality and evolutionary algorithm)

19 November 2014

Pengcheng Huang
pengcheng.huang@tik.ee.ethz.ch
HW/SW Codesign

Exercise 8:
Design Space Exploration
(Pareto optimality and evolutionary algorithm)

19 November 2014

Recap

Warm-up yourself

Pengcheng Huang
pengcheng.huang@tik.ee.ethz.ch
Plan

• Recap

• Question explanation

• Your time to solve (45 mins)

• Solution explanation
Plan

- Recap
- Question explanation
  - Your time to solve
  - Solution explanation
Multi-criteria optimization / Pareto-points

A dominates B
Multi-criteria optimization / Pareto-points

A and C not comparable

A

C

cost

cheaper

faster

latency
Multi-criteria optimization / Pareto-points

Points dominated by C

Pareto-front

Pareto-Points. = Points, which are not dominated by others

faster

cheaper

cost

latency

Swiss Federal Institute of Technology

Computer Engineering
and Networks Laboratory
Pareto-dominance, Pareto Points

• A (design) point $J_k$ is **pareto-dominated** by $J_i$, if $J_i$ is
  – better or equal than $J_k$ in all criteria and
  – better in at least one criterion.

• A point is Pareto-optimal or a **Pareto-point**, if it is not dominated.

• The domination relation imposes a partial order on all design points
  – We are faced with a set of optimal solutions.
Let us suppose, we would like to select a typewriting device. Criteria are
- mobility (related to weight)
- comfort (related to keyboard size and performance)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Device</th>
<th>weight (kg)</th>
<th>comfort rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>☂</td>
<td>PC of 2009</td>
<td>20.00</td>
<td>10</td>
</tr>
<tr>
<td>☂</td>
<td>PC of 1984</td>
<td>7.50</td>
<td>7</td>
</tr>
<tr>
<td>☂</td>
<td>Laptop</td>
<td>3.00</td>
<td>9</td>
</tr>
<tr>
<td>☂</td>
<td>Typewriter</td>
<td>9.00</td>
<td>5</td>
</tr>
<tr>
<td>☂</td>
<td>Touchscreen Smartphone</td>
<td>0.11</td>
<td>2</td>
</tr>
<tr>
<td>☂</td>
<td>PDA with large keyboard</td>
<td>0.09</td>
<td>3</td>
</tr>
<tr>
<td>☂</td>
<td>PDA with small keyboard</td>
<td>0.11</td>
<td>4</td>
</tr>
<tr>
<td>☂</td>
<td>Organizer with tiny keyboard</td>
<td>0.08</td>
<td>1</td>
</tr>
</tbody>
</table>
Multiobjective Optimization
Exercise Topics

• Given a task graph, a number of resource components, and constraints (binding to component, cost of a component, execution times of tasks on a component)
  – Determine all possible resource allocations, bindings of tasks, (and schedules). Calculate cost and total execution time for each design point

• Determine Pareto points in the design space
• Evolutionary algorithm operations
Exercise 8.1

- Four tasks T1 .. T4 can be executed on components as shown in Table 1.
- Each component executes tasks serially.
- Tasks are non-preemptive.

![Task graph](image)

**Table 1: Available components with cost and execution times for tasks T1 .. T4.**

<table>
<thead>
<tr>
<th>component</th>
<th>number</th>
<th>cost</th>
<th>execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIPS</strong></td>
<td>1</td>
<td>200,-</td>
<td>5 ms — — 2 ms</td>
</tr>
<tr>
<td><strong>DSP</strong></td>
<td>1</td>
<td>120,-</td>
<td>— 20 ms 18 ms 5 ms</td>
</tr>
<tr>
<td><strong>FPGA</strong></td>
<td>1</td>
<td>240,-</td>
<td>— 12 ms 10 ms —</td>
</tr>
<tr>
<td><strong>ASIC</strong></td>
<td>1</td>
<td>400,-</td>
<td>—  — 800 μs —</td>
</tr>
</tbody>
</table>
Exercise 8.1 (a)

- Construct the design-space by listing all possible design points. Design point consists of allocation, binding, and scheduling.
  - Find the total cost and execution time for each design point.

- For example, one of the design points is: (T1 \rightarrow\text{MIPS}, T2 \rightarrow\text{DSP}, T3 \rightarrow\text{DSP}, T4 \rightarrow\text{DSP}). The total cost and execution time for this design point are 320 and 48 ms respectively.
Exercise 8.1 (b)

• Draw the design points in a cost-time diagram. Which design points are Pareto points?
Exercise 8.1 (c)

• Consider an allocation without resource constraints. That means we are given an arbitrarily high number of components of each type (MIPS, DSP, FPGA and ASIC).
  – Are there new design points?
  – Does the set of Pareto points change?
Exercise 8.1 (d)

• How many design points exist at least if a task graph comprises \( n \) tasks, each task can be executed on at least three component types, and there are no resource constraints?
DSE With EA: Exercise 8.2

An individual is defined by:

- **Allocation** (which resources are used?)
- **Binding** (which task runs on which resource?)
- **Scheduling** (how are tasks scheduled on a single resource?)
  - Here, scheduling is also part of the chromosome
DSE With EA: Exercise 8.2 (2)

- **6 tasks**
  with sequence graph:

  ![Sequence Graph](image)

- **3 resources**
  with possible bindings between tasks and resources:

  ![Resource Bindings](image)
DSE With EA: Exercise 8.2 (2)

• Tasks:
  – Find suitable **encoding** for (i) allocation, (ii) binding, and (iii) scheduling
  – Prove **completeness**
  – Evaluate **uniformity, redundancy, and feasibility**
  – Specify **mutation** and **recombination** operators
    • How to handle possible unfeasible solutions?
Plan

• Recap

• Question explanation

• Solution explanation

Your time to solve (45 mins)
Plan

• Recap

• Question explanation

• Your time to solve

Solution explanation
Solution 8.1 (a)

Design points:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
</tr>
<tr>
<td>$T_2$</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>FPGA</td>
<td>FPGA</td>
<td>FPGA</td>
<td>FPGA</td>
</tr>
<tr>
<td>$T_3$</td>
<td>DSP</td>
<td>DSP</td>
<td>FPGA</td>
<td>FPGA</td>
<td>ASIC</td>
<td>ASIC</td>
<td>DSP</td>
<td>DSP</td>
<td>FPGA</td>
<td>FPGA</td>
<td>ASIC</td>
<td>DSP</td>
</tr>
<tr>
<td>$T_4$</td>
<td>DSP</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
</tr>
<tr>
<td>Time</td>
<td>48 ms</td>
<td>45 ms</td>
<td>30 ms</td>
<td>27 ms</td>
<td>27 ms</td>
<td>28 ms</td>
<td>25 ms</td>
<td>32 ms</td>
<td>29 ms</td>
<td>22 ms</td>
<td>19 ms</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Available components with cost and execution times for tasks $T_1 \ldots T_4$. 

$1 \times 2 \times 3 \times 2$
Solution 8.1 (b)

Pareto-points: 2, 8, 10, 12
Solution 8.1 (c)

New design points:

<table>
<thead>
<tr>
<th></th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
<td>MIPS</td>
</tr>
<tr>
<td>$T_2$</td>
<td>DSP</td>
<td>DSP1</td>
<td>FPGA1</td>
<td>FPGA1</td>
</tr>
<tr>
<td>$T_3$</td>
<td>DSP2</td>
<td>DSP2</td>
<td>FPGA2</td>
<td>FPGA2</td>
</tr>
<tr>
<td>$T_4$</td>
<td>MIPS</td>
<td>DSP1/2</td>
<td>MIPS</td>
<td>DSP</td>
</tr>
<tr>
<td>Cost</td>
<td>440.-</td>
<td>440.-</td>
<td>680.-</td>
<td>800.-</td>
</tr>
<tr>
<td>Execution time</td>
<td>27 ms</td>
<td>30 ms</td>
<td>19 ms</td>
<td>22 ms</td>
</tr>
</tbody>
</table>

Original design points:

Pareto-points: 2, 8, 13, 15
Solution 8.1 (d)

Minimum number of design points: $3^n$
Exercise 8.2: Possible Solution I

- Allocation:

<table>
<thead>
<tr>
<th>Res.</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alloc.</td>
<td>1</td>
<td>1</td>
<td>0/1</td>
</tr>
</tbody>
</table>

- Binding:

<table>
<thead>
<tr>
<th>Arrow</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bound</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
<td>0/1</td>
</tr>
</tbody>
</table>

- Scheduling:

<table>
<thead>
<tr>
<th>Task</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

- Uniform, redundant, complete, leads to unfeasible solutions

- Possible strategy to have feasible scheduling solutions with mutation/recombination: use a constraint matrix
Exercise 8.2: Possible Solution 2

- **Allocation and Binding:**

<table>
<thead>
<tr>
<th>Task</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.</td>
<td>1</td>
<td>2</td>
<td>1/2</td>
<td>1/2</td>
<td>2/3</td>
<td>2/3</td>
</tr>
</tbody>
</table>

- **Scheduling:**

<table>
<thead>
<tr>
<th>Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>1/2/3</td>
<td>1/2/3/4</td>
<td>1/2/3/4/5</td>
<td>2/3/4</td>
<td>2/3/5</td>
<td>6</td>
</tr>
</tbody>
</table>

- Uniform, not redundant, complete, but leads to unfeasible solutions