HW/SW Codesign

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

22 November 2017

Rehan Ahmed
rehan.ahmed@tik.ee.ethz.ch
Plan

• Recap
• Question explanation
• Your time to solve (~75 mins)
• Solution explanation
• Projects @ TEC
Multi-criteria optimization / Pareto-points

A dominates B

A cheaper, faster

B more expensive, slower
Multi-criteria optimization / Pareto-points

A and C not comparable

cheaper

cost

defaster

latency
Multi-criteria optimization / Pareto-points

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

Points dominated by C

Pareto-front
Evolutionary Algorithm Cycle

- Representation operator
- Mating selection
- Crossover
- Initial/parent set
- Children set
- Environmental selection
- Mutation operator
Representation

solutions encoded by vectors, matrices, trees, lists, ...

Issues:
- completeness (each solution has an encoding)
- uniformity (all solutions are represented equally often)
- feasibility (each encoding maps to a feasible solution)
- redundancy (a solution has multiple encodings)
Vector Mutation: Examples

**Bit vectors:**

```
1 0 1 1 1 0
```

each bit is flipped with probability 1/6

```
1 0 0 1 1 0
```

**Permutations:**

```
1 2 3 4 5 6
```

swap

```
1 3 4 2 5 6
```

```
1 2 3 4 5 6
```

rearrange

```
1 3 4 2 5 6
```
Vector Crossover: Examples

**Bit vectors:**

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \rightarrow \]

| 1 | 1 | 0 | 0 | 0 | 1 |

**Permutations:**

\[ \text{parents} \]

| 1 | 2 | 3 | 4 | 5 | 6 |

\[ \rightarrow \]

\[ \text{child} \]

| 1 | 2 | 3 | 6 | 4 | 5 |
**Constraint Handling**

**Constraint:** \( g(x) \geq 0 \)

solution in decision space

**Approaches:**

- representation is chosen such that decoding always yields a feasible solution
- construct initialization and neighborhood operators such that infeasible solutions are not generated
- add to children population only feasible solutions
- in environmental selection, preferably select feasible solutions
- calculate constraint violation \( g(x) \) and incorporate it into objective function using a penalty function: \( \text{penalty}(x) > 0 \) if \( g(x) < 0 \), \( \text{penalty}(x) = 0 \) if \( g(x) \geq 0 \). For example, add penalty function to every objective.
- include the constraints as new objectives
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.

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

<table>
<thead>
<tr>
<th>component</th>
<th>number</th>
<th>cost</th>
<th>execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>MIPS</td>
<td>1</td>
<td>200,-</td>
<td>5 ms</td>
</tr>
<tr>
<td>DSP</td>
<td>1</td>
<td>120,-</td>
<td>---</td>
</tr>
<tr>
<td>FPGA</td>
<td>1</td>
<td>240,-</td>
<td>---</td>
</tr>
<tr>
<td>ASIC</td>
<td>1</td>
<td>400,-</td>
<td>---</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 schedule.

- 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 = \text{MIPS}+\text{DSP} = 200+120 = 320

Execution time: 5ms + 20ms + 18ms + 5ms = 48ms
Exercise 8.1 (b)

Draw the design points in a cost-time diagram. Which design points are Pareto points?
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?
An individual is defined by:

- **Allocation** (which resources are used?)
- **Binding** (which task runs on which resource?)
- **Schedule** (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:

- 3 resources
  with possible bindings between tasks and resources:
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
• Your time to solve (~75 mins)
• Solution explanation
• Projects @ TIK
Plan

• Recap
• Question explanation
• Your time to solve
• Solution explanation
• Projects @ TEC
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>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>FPGA</td>
<td>FPGA</td>
<td>FPGA</td>
<td>ASIC</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>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution time</td>
<td>48 ms</td>
<td>45 ms</td>
<td>30 ms</td>
<td>27 ms</td>
<td>30 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>
</tr>
</tbody>
</table>

Considering 4 additional design points where $T_3$ executes before $T_2$ is valid solution.

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Number</th>
<th>Cost</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
<th>$T_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIPS</td>
<td>1</td>
<td>200.-</td>
<td>5 ms</td>
<td>—</td>
<td>—</td>
<td>2 ms</td>
</tr>
<tr>
<td>DSP</td>
<td>1</td>
<td>120.-</td>
<td>—</td>
<td>20 ms</td>
<td>18 ms</td>
<td>5 ms</td>
</tr>
<tr>
<td>FPGA</td>
<td>1</td>
<td>240.-</td>
<td>—</td>
<td>12 ms</td>
<td>10 ms</td>
<td>—</td>
</tr>
<tr>
<td>ASIC</td>
<td>1</td>
<td>400.-</td>
<td>—</td>
<td>—</td>
<td>800 $\mu$s</td>
<td>—</td>
</tr>
</tbody>
</table>
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>FPGA2</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:

<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>DSP</td>
<td>DSP</td>
<td>DSP</td>
<td>DSP</td>
</tr>
<tr>
<td>$T_3$</td>
<td>DSP</td>
<td>FPGA</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
<td>ASIC</td>
</tr>
<tr>
<td>$T_4$</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>Execution time</td>
<td>48 ms</td>
<td>48 ms</td>
<td>30 ms</td>
<td>30 ms</td>
<td>27 ms</td>
<td>27 ms</td>
<td>27 ms</td>
<td>27 ms</td>
<td>32 ms</td>
<td>32 ms</td>
<td>22 ms</td>
<td>19 ms</td>
</tr>
</tbody>
</table>

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

Minimum number of design points: $3^n$
DSE With EA: Exercise 8.2 (2)

- **6 tasks**
  with sequence graph:

```
1 -> 4 -> 5 -> 6
```

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

```
1 1
2 2 2 3 4
3 5
4 7
5 8
6 9 10
```

```
Exercise 8.2: Possible Solution 1

- **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>Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

- **Non-Uniform, redundant, complete, leads to infeasible solutions**

- **Why infeasible?** Can bind a task to an unallocated resource.
  - e.g. Binding for edge 8 is 1, while allocation for resource 3 is 0
Decoding the schedule

1. Each resource (R1, R2, R3) constructs a set of tasks $S$ which are eligible for execution.

2. **If $S$ is empty**: the resource waits for a task to become eligible

3. **If $S$ has one task**: that task is executed

4. **If $S$ has more than one tasks**: the task with lowest order is executed

5. After completion of a task, steps 1-4 are repeated
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>
</table>

- **Non-uniform, redundant, complete, feasible**
Why Redundant??

For redundancy: list all possible schedule encodings and count the ones where T2 Order < T3 Order

Only relative order of T2 and T3 important for this binding
Why Non-uniform

Different individuals (allocation, binding, schedule) will have different number of redundant schedule encodings.

Feel free to prove/disprove
Exercise 8.2: Possible Solution 2

- Mutation & Recombination

### Scheduling 1

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

### Mutated 1

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

### Mutation/Swapping

### Recombination/Crossover

### Scheduling 2

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

### Final

<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>Order</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
Exercise 8.2: Possible Solution 2

• What if it goes wrong?

...Discard

...Change your randomization rule

...or keep based on fitness evaluation

$2^{nd}$ in general difficult.