Task Scheduling
A Lecture in CE Freshman Seminar Series:
Ten Puzzling Problems in Computer Engineering
About This Presentation

This presentation belongs to the lecture series entitled “Ten Puzzling Problems in Computer Engineering,” devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

<table>
<thead>
<tr>
<th>Edition</th>
<th>Released</th>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>May 2007</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mini-Sudoku Puzzle

Complete entries in this chart so that numbers 1-6 appear without repetition in each row, each column and each $2 \times 3$ block.

Standard Sudoku consists of a $9 \times 9$ chart, but this mini version is good for a quick fix.

USA Today carries a daily mini-Sudoku at its site: http://puzzles.usatoday.com

Sudoku isn’t a math puzzle: Can substitute letters A-F or any other six symbols for numbers 1-6.

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>
**Mini-Sudoku Puzzle: Solution Method**

Complete entries in this chart so that letters A-F appear without repetition in each row, each column and each 2 × 3 block.

To continue from here, write down all possible choices in the remaining blank boxes and see whether the resulting info leads to more progress.

SuDoKu: abbr. in Japanese for “numbers must be single.” Euler may have invented it; Howard Garns (US) & Wayne Gould (HK) popularized it in modern times.

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th></th>
<th>D</th>
<th></th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td>D</td>
<td></td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>D</td>
<td>C</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>E</td>
<td>C</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>D</td>
<td>C</td>
<td></td>
<td>F</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

May 2007

Task Scheduling

Slide 4
Sudoku Puzzle: Easy Example

Complete entries in this chart so that numbers 1-9 appear without repetition in each row, each column and each $3 \times 3$ block.

Many newspapers carry these puzzles; there are also many collections in book form.

Sudoku puzzles of varying difficulties (easy, medium, hard, evil) are available at http://www.websudoku.com and several other Web sites, such as USA Today’s site http://puzzles.usatoday.com.
Strategy 1: Identify a missing number from a row, column, or block; if you can exclude all but one cell for that number, then write it down.

Strategy 2: When you can’t make progress by Strategy 1, write down all candidate numbers in the cells and try to eliminate a number of options via reasoning. For example if xy, xy, xyz are candidates in three cells of a block, then the cell marked xyz must hold z.
Sudoku Puzzle: Hard Example

Complete entries in this chart so that numbers 1-9 appear without repetition in each row, each column and each 3 × 3 block.

Hard puzzles typically have fewer entries supplied, with each row, column, or block containing only a few entries.

Hard puzzles may have handles or starting points (5 in the top left block or 9 in center and lower right blocks).

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>4</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Variations on Sudoku

Other sizes (e.g., 6 \times 6, with 2 \times 3 blocks; or 16 \times 16, with 4 \times 4 blocks)

Combining this 2000s phenomenon with Rubik’s cube of the 1980s . . .

or with the age-old sliding 15 puzzle
Task Scheduling Problem

We have a set of tasks

There are some "processors" that can execute tasks

Assign tasks to processors so as to meet certain constraints

A task may fit only some processors
Tasks may have prerequisites tasks
Preemption may (not) be allowed
Tasks may have deadlines
Shortest schedule may be required

Numbers in Sudoku puzzle

Cells in Sudoku puzzle can hold numbers

Place numbers in cells while honoring some constraints

Use only numbers 1-9
Some numbers already placed
Different numbers in each row
Different numbers in each column
Different numbers in each block

Virtually all instances of the task scheduling problem are difficult (NP-complete), just like Sudoku
Resource Allocation Problem

We have a set of resources

There are “locations” where resources may be placed

Assign resources to locations to meet certain constraints

A resource may fit only some locations
Resources must be “easily” accessible
Resource mobility may (not) be allowed
Resource cost may differ by location
Lowest-cost assignment may be required

Numbers in Sudoku puzzle

Cells in Sudoku puzzle can hold numbers

Place numbers in cells while honoring some constraints

Use only numbers 1-9
Some numbers already placed
Different numbers in each row
Different numbers in each column
Different numbers in each block

Virtually all instances of the resource allocation problem are difficult (NP-complete), just like Sudoku

UCSB
Scheduling Required CE Courses

Constraints
Prerequisite:
Solid downward arrow
Corequisite:
Dashed sideways arrow
Units per quarter: \( \leq 18 \)
Scheduling Required CE Courses

Constraints
Prerequisite:
Solid downward arrow
Corequisite:
Dashed sideways arrow
Units per quarter: ≤ 18

Units
1
2
3
4
5

12 units
16
20 units

Almost done!

May 2007
Task Scheduling
Slide 12
### Job-Shop Scheduling

#### Job Details

<table>
<thead>
<tr>
<th>Job</th>
<th>Task</th>
<th>Machine</th>
<th>Time</th>
<th>Staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ja</td>
<td>Ta1</td>
<td>M1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Ja</td>
<td>Ta2</td>
<td>M3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Jb</td>
<td>Tb1</td>
<td>M2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Jb</td>
<td>Tb2</td>
<td>M1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Jb</td>
<td>Tb3</td>
<td>M2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Jc</td>
<td>Tc1</td>
<td>M3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Jd</td>
<td>Td1</td>
<td>M1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Jd</td>
<td>Td2</td>
<td>M2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Diagram

![Diagram showing job-shop scheduling with resource allocation and task timing]
Truck Scheduling

- Seattle
- Portland
- Boise
- LA
- SF
- Orlando
- NYC
- Chicago
- Dallas

- Truck load:
  - 00%
  - 20%
  - 35%
  - 40%
  - 50%
  - 75%

- Required trip:
  - Directional arrows between cities.
Multiprocessor Scheduling

**Task graph with unit-time tasks**
Here’s a heuristic known as list scheduling:
1. Find the depth $T_\infty$ of the task graph
2. Take $T_\infty$ as a goal for the running time $T_p$
3. Determine the latest possible start times
4. Assign priorities in order of latest times

$T_\infty = 8$ (execution time goal)
Latest start times: see the layered diagram
Priorities: shown on the diagram in red

When two tasks have the same “latest start time,” a secondary tie-breaking rule is used
Assignment to Processors

Tasks listed in priority order

Even in this simple case of unit-time tasks, multiprocessor scheduling remains difficult with as few as 3 processors.