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>
<th>Revised</th>
<th>Revised</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>May 2012</td>
<td>May 2015</td>
<td>May 2016</td>
<td>May 2020</td>
<td></td>
</tr>
</tbody>
</table>
Complete entries in this chart so that numbers 1-6 appear without repetition in each row, each column and each $2 \times 3$ block.

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

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

The following site carries mini-Sudoku puzzles: [https://sudoku.cool/mini-sudoku.php](https://sudoku.cool/mini-sudoku.php)

Sudoku isn’t a math puzzle: We can use the letters A-F, or any other six symbols, instead of the numbers 1-6.
**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 $2 \times 3$ 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.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>B</td>
<td></td>
<td>D</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>D</td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>D</td>
<td>C</td>
<td>A</td>
<td></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></td>
</tr>
<tr>
<td>AE</td>
<td></td>
<td></td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

---

May 2020  

UCSB

Task Scheduling  

BParham
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
Sudoku
Puzzle
Solution
Method

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.

```
5 6 7 4
4 7 9 5
1 3 25 4 8 6
6 58 4 7 2 9
1 58 3
7 9 28 3 5 1
2 7 9 8 6 3
8 4 1 5
```

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.

```
7, 8, 9 missing from this column
1, 2, 3, 4 missing from this row
```
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 x 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).
Constructing a (Mini-)Sudoku Puzzle

Begin with a completed puzzle and one by one remove selected entries that can be deduced from the remaining ones. This will ensure a unique solution, which is a desirable attribute.

Q1: Remove additional entries from this puzzle while maintaining the uniqueness of solution.

Q2: Build a 4 x 4 puzzle with unique solution and the fewest initial entries.
Interesting Facts about Sudoku

Theoretically, $n^2 \times n^2$ Sudoku is NP-complete, but for standard $9 \times 9$ puzzles, the number of possibilities is small enough to be tractable.

The number of valid solution grids for the standard $9\times9$ Sudoku is $6,670,903,752,021,072,936,960$. This number is equal to $9! \times 72^2 \times 2^7 \times 27,704,267,971$, the last factor of which is prime.

In a $9 \times 9$ Sudoku puzzle, you may need at least 17 initial entries (clues) for the solution to be unique; no one knows whether a 16-clue puzzle with unique solution exists.

Q3: Solve this irregular Sudoku puzzle.
Interesting Variations on Sudoku

From *New York Times*, Sunday 2018/12/16
Other Puzzles Based 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

Q4: Construct Latin squares of sizes $3 \times 3$ and $4 \times 4$. 
Task Scheduling Problem

We have a set of tasks
Numbers in Sudoku puzzle

There are some “processors” that can execute tasks
Cells in Sudoku puzzle can hold numbers

Assign tasks to processors so as to meet certain constraints
Place numbers in cells while honoring some constraints

A task may fit only some processors
Use only numbers 1-9
Tasks may have prerequisite tasks
Some numbers already placed
Preemption may be (dis)allowed
Different numbers in each row
Tasks may have deadlines
Different numbers in each column
Shortest schedule may be sought
Different numbers in each block

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

<table>
<thead>
<tr>
<th>We have a set of resources</th>
<th>Numbers in Sudoku puzzle</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are “locations” where resources may be placed</td>
<td>Cells in Sudoku puzzle can hold numbers</td>
</tr>
<tr>
<td>Assign resources to locations to meet certain constraints</td>
<td>Place numbers in cells while honoring some constraints</td>
</tr>
</tbody>
</table>

A resource may fit only some locations
Resources must be “easily” accessible
Resource mobility may be (dis)allowed
Resource cost may differ by location
Lowest-cost assignment may be sought

<table>
<thead>
<tr>
<th>Use only numbers 1-9</th>
<th>Virtually all instances of the resource allocation problem are difficult (NP-hard), just like Sudoku</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some numbers already placed</td>
<td>Some numbers already placed</td>
</tr>
<tr>
<td>Different numbers in each row</td>
<td>Different numbers in each column</td>
</tr>
<tr>
<td>Different numbers in each column</td>
<td>Different numbers in each block</td>
</tr>
</tbody>
</table>

---

May 2020

Task Scheduling
Scheduling Required CE Courses

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

Units
1
2
3
4
5

Engr 101

CS 10
Math 3A
ECE 1
Chem 1A
Chem 1AL

CS 20
Math 3B
Phys 1
Chem 1B
Chem 1BL

CS 40
CS 60
Phys 3L
Phys 3
Phys 4L
Phys 4

CS 130A

CS 170

Math 3C
Phys 2
Math 5A

ECE 15A
ECE 15B
ECE 152A

ECE 152B

ECE 154

ECE 139

Phys 1

Math 3A
Math 3B
Phys 1

Math 5A

Phys 4L
Phys 4

ECE 15A

ECE 2A

ECE 2B

ECE 2C

ECE 139

Or CS 30
Or PSTAT 120A

12 units
20 units

12 units
20 units

Units
1
2
3
4
5

Upper - division standing

May 2020
Task Scheduling
Slide 14
Scheduling Required CE Courses

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

Almost done!

May 2020
Task Scheduling
BParhami
Slide 15
Job-Shop Scheduling

<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>

Task Scheduling
### Schedule Refinement

<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>M3</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>

May 2020
Truck Scheduling

- Truck location
- Required trip
- 00% Truck load

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

Percentage of load:
- 35%
- 40%
- 50%
- 75%
- 20%
- 20%

May 2020
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
1* 2 3 4 6 5 7 8 9 10 11 12 13

Even in this simple case of unit-time tasks, multiprocessor scheduling remains difficult with as few as 3 processors
Two Related and Similar Problems

<table>
<thead>
<tr>
<th>The knapsack problem</th>
<th>Off-line game of Tetris</th>
</tr>
</thead>
<tbody>
<tr>
<td>We have storage capacity $W$ and $n$ files of sizes $w_i$ and values $v_i$</td>
<td>We have a rectangular bin and a sequence of tetrominos</td>
</tr>
<tr>
<td>Pick a max-value subset of files that fit in the storage space $W$</td>
<td>Find optimal play to maximize the number of pieces used</td>
</tr>
</tbody>
</table>

Files cannot be broken into pieces
Naïve solution: Examine all $2^n$ subsets
Dynamic programming solution
Various heuristic aids
Approx. solutions (say, 90% of optimal)

Pieces can only be rotated
Exponentially many choices

There are many other related and similarly hard problems, some of which don’t even admit efficient approximations

May 2020

Task Scheduling

UCSB

BParhani