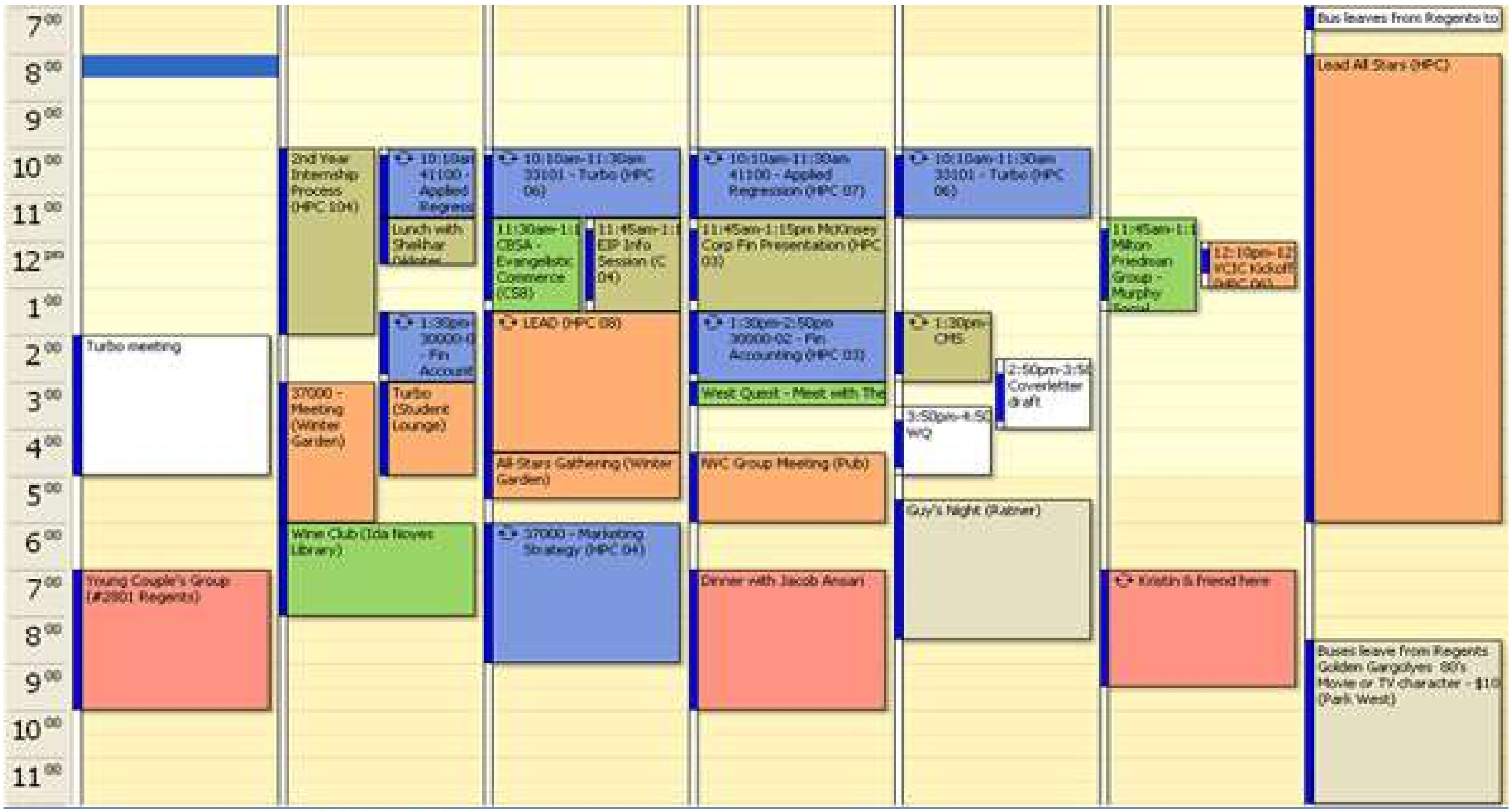
Task Scheduling

A Lecture in CE Freshman Seminar Series:
Ten Puzzling Problems in Computer Engineering



About This Presentation

Edition	Released	Revised	Revised	Revised	Revised
First	May 2007	May 2008	May 2009	May 2010	May 2011
		May 2012	May 2015	May 2016	May 2020

Mini-Sudoku Puzzle

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

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

Standard Sudoku consists of a 9 × 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

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 × 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

AE	B		D		C
C		D			A
F	D	C	A		
B	A	E	C	D	F
D	C		F	A	
AE				C	D

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 × 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

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

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

					8			
5			6	7	4			2
4		7	9		5	6		1
1	3	25	4	8	6	25	7	9
6	58	4	7	2	9	1	58	3
7	9	28	3	5	1	28	6	4
2	7	9	8	6	3	4	1	5
8	4	1	5	9	7	23	23	6
					2			

7, 8, 9 missing from this column

1, 2, 3, 4 missing from this row

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

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

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.

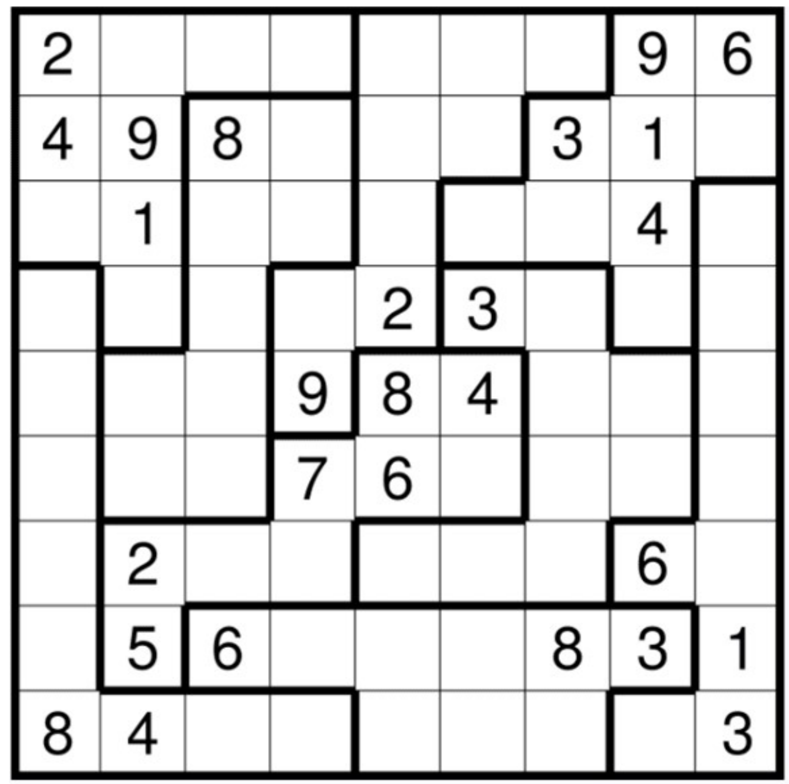
	2		4		3
3		4			1
6	4	3	1		
2	1	5	3	4	6
4	3	2	6	1	5
				3	4

Interesting Facts about Sudoku

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

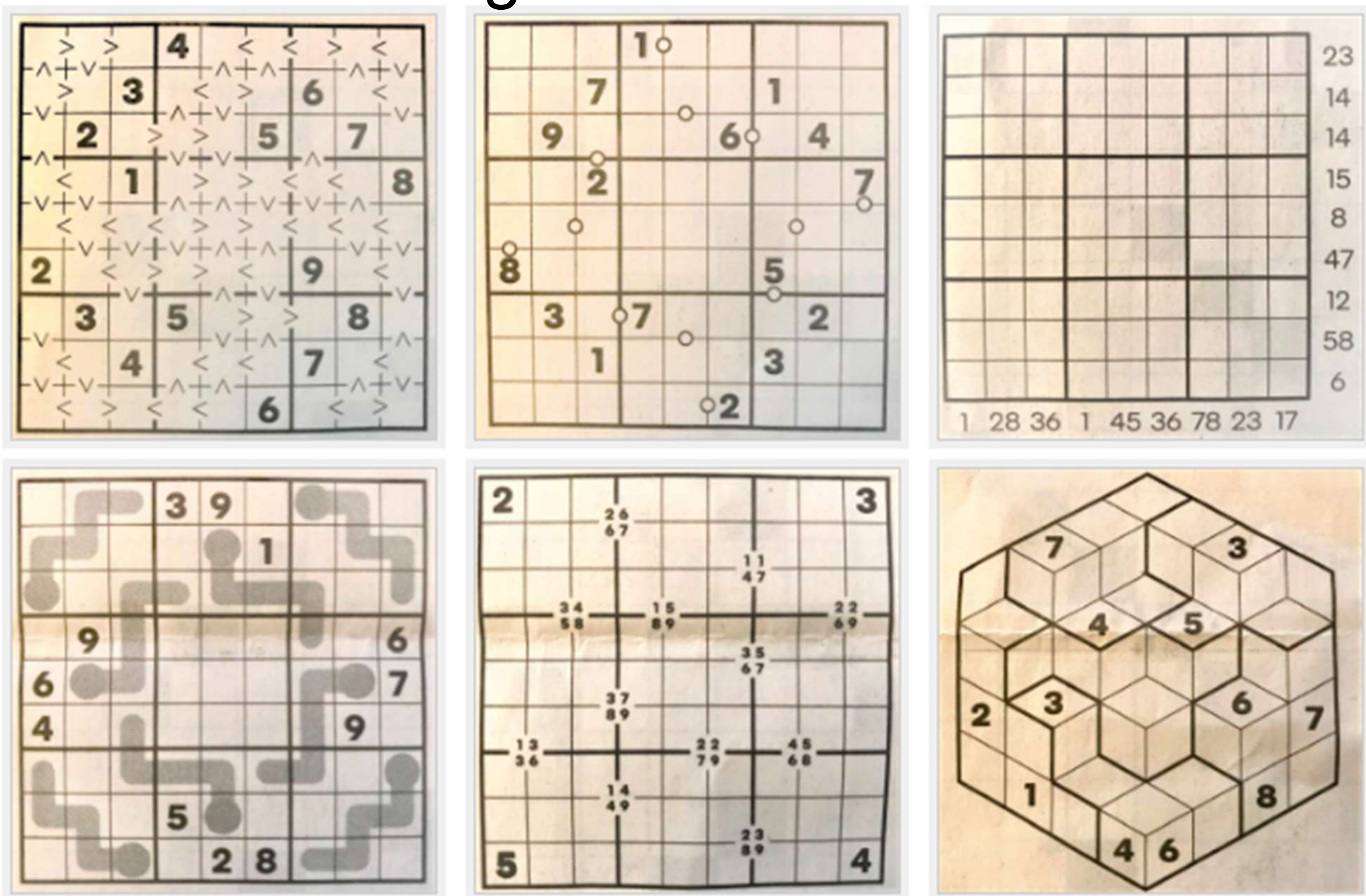
The number of valid solution grids for the standard 9×9 Sudoku is 6,670,903,752,021,072,936,960. This number is equal to 9! × 72^2 × 2^7 × 27,704,267,971, the last factor of which is prime

In a 9 × 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., 6×6, with 2×3 blocks; or 16×16, with 4×4 blocks)

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

or with the age-old sliding 15 puzzle

Latin square

Q4: Construct Latin squares of sizes 3×3 and 4×4.

Task Scheduling Problem

We have a set of 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 Tasks may have prerequisite tasks Preemption may be (dis)allowed Tasks may have deadlines Shortest schedule may be sought	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-hard), just like Sudoku

Resource Allocation Problem

We have a set of of resources

Numbers in Sudoku puzzle

There are "locations" where resources may be placed

Cells in Sudoku puzzle can hold numbers

Assign resources to locations to meet certain constraints

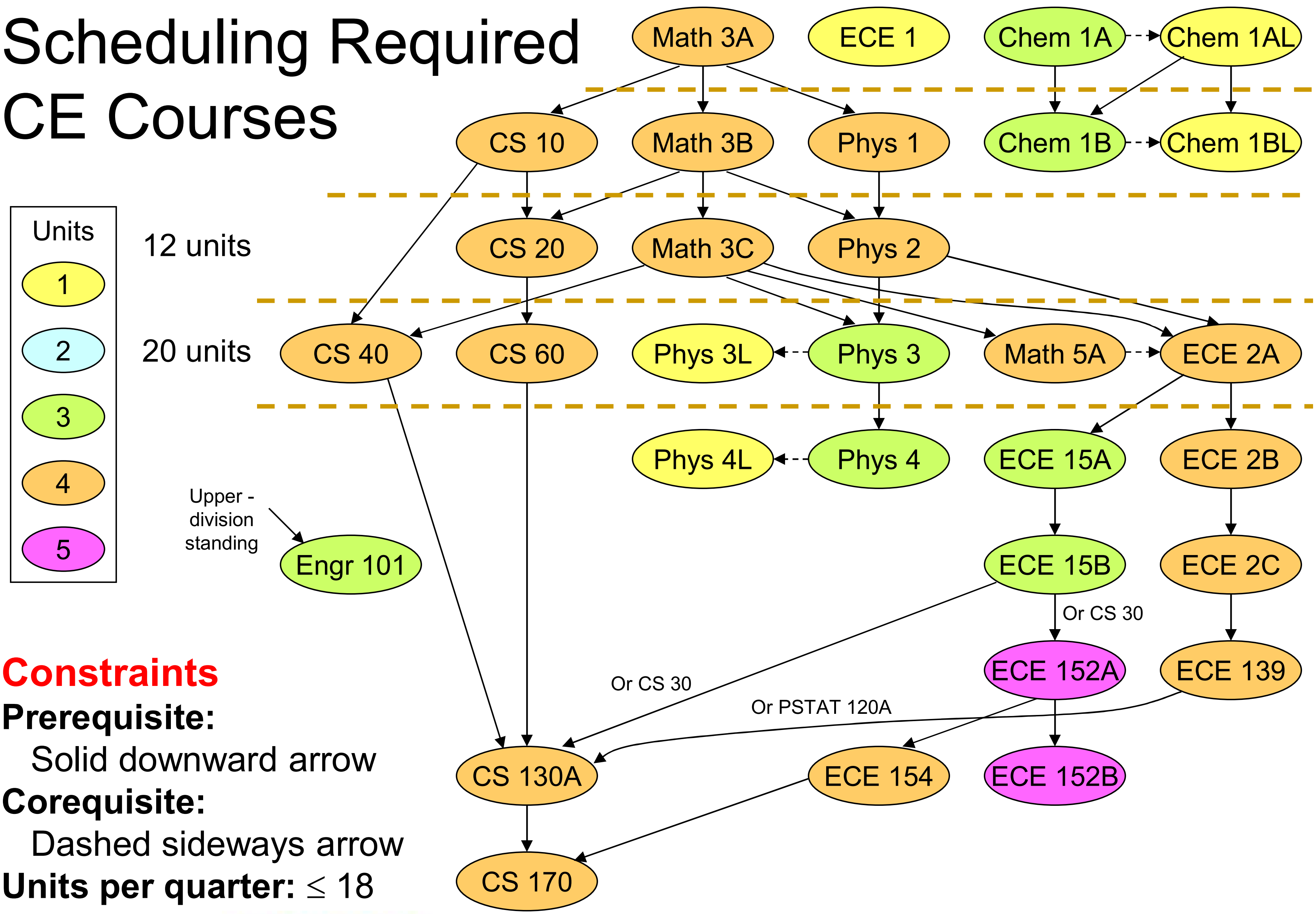
Place numbers in cells while honoring some constraints

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

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-hard), just like Sudoku

Scheduling Required CE Courses



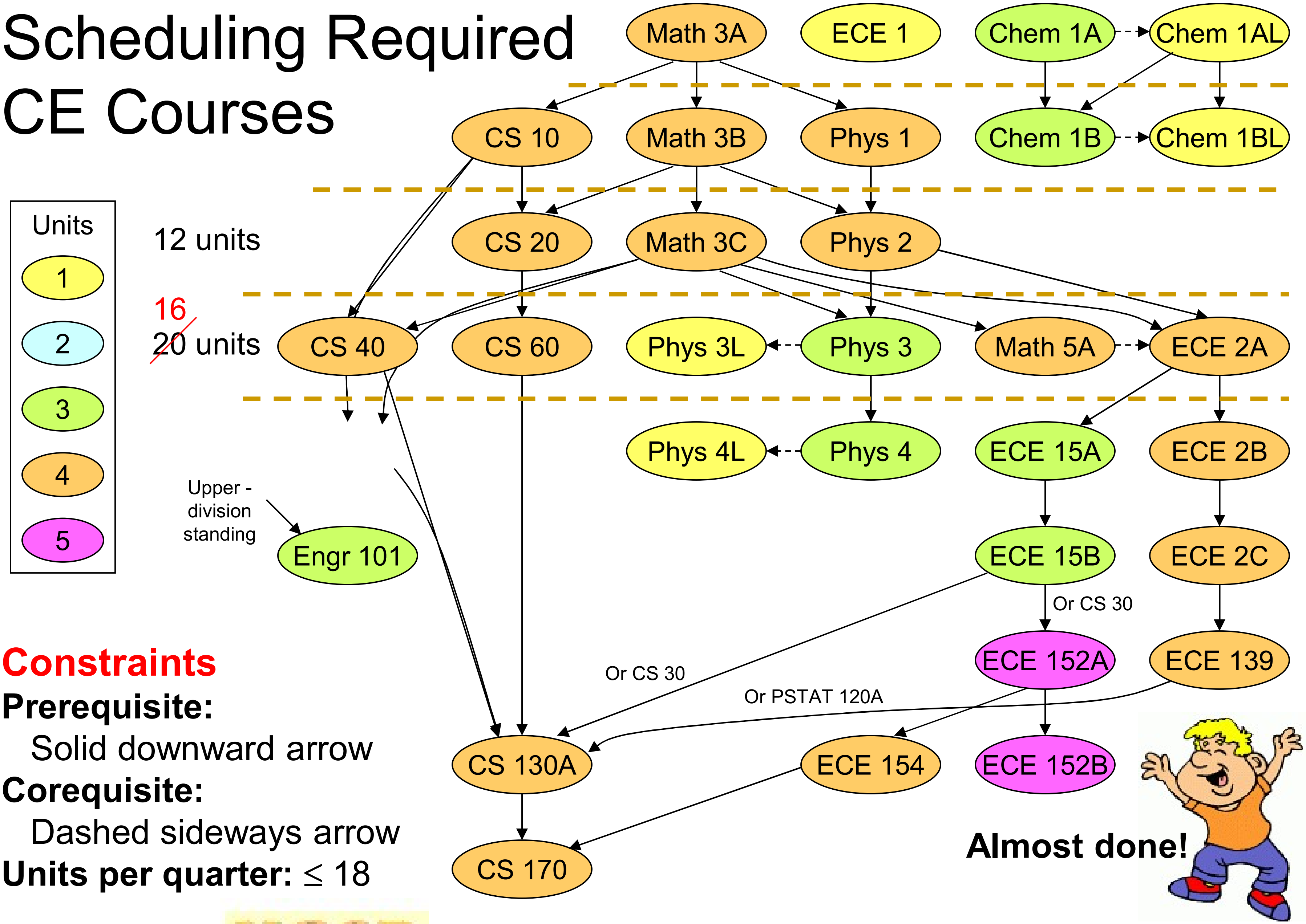
Constraints

Prerequisite:
Solid downward arrow

Corequisite:
Dashed sideways arrow

Units per quarter: ≤ 18

Scheduling Required CE Courses



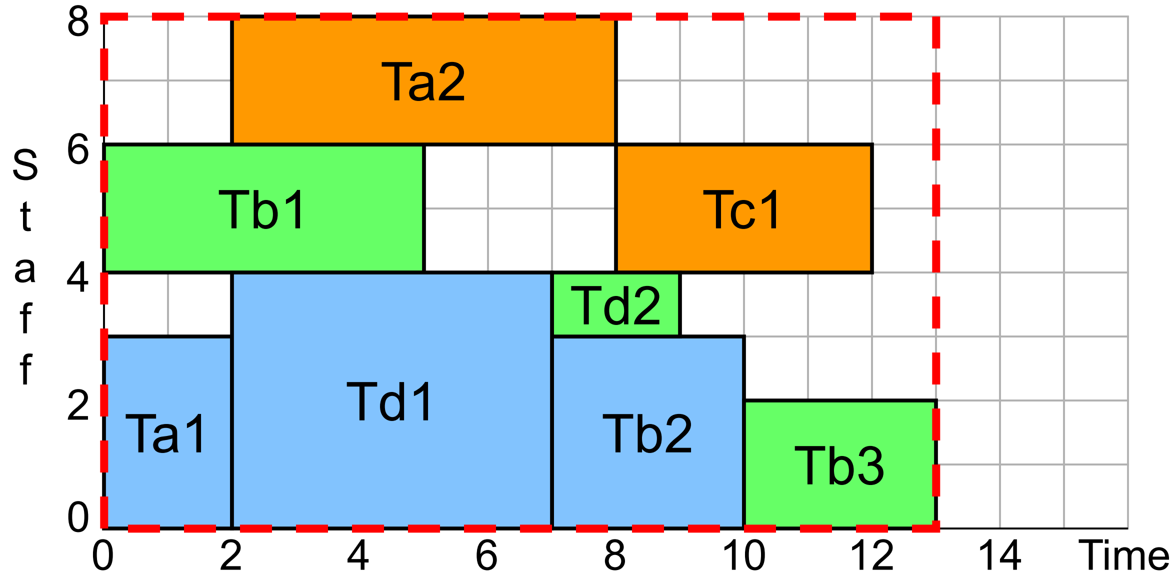
Constraints

Prerequisite:
Solid downward arrow

Corequisite:
Dashed sideways arrow

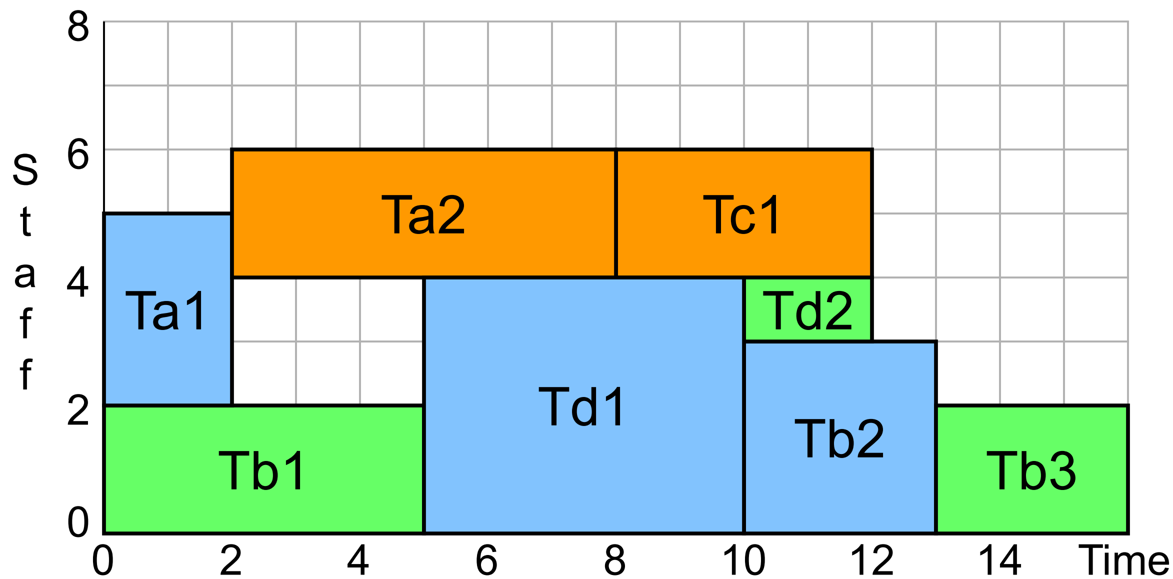
Units per quarter: $\leq$ 18

Job-Shop Scheduling

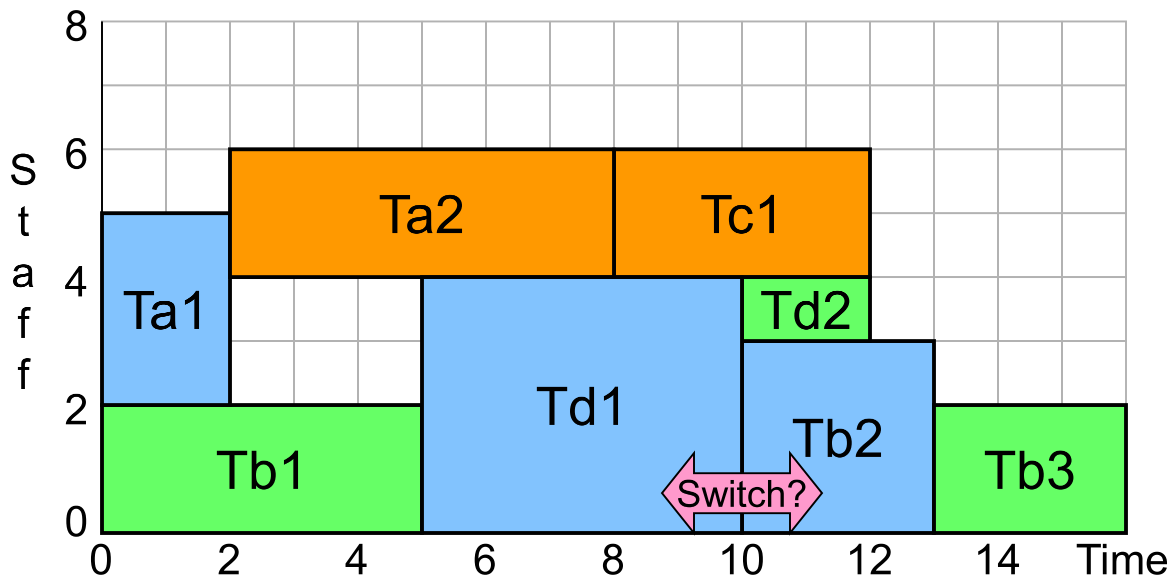


Job	Task	Machine	Time	Staff
Ja	Ta1	M1	2	3
Ja	Ta2	M3	6	2
Jb	Tb1	M2	5	2
Jb	Tb2	M1	3	3
Jb	Tb3	M2	3	2
Jc	Tc1	M3	4	2
Jd	Td1	M1	5	4
Jd	Td2	M2	2	1

M1 M2 M3

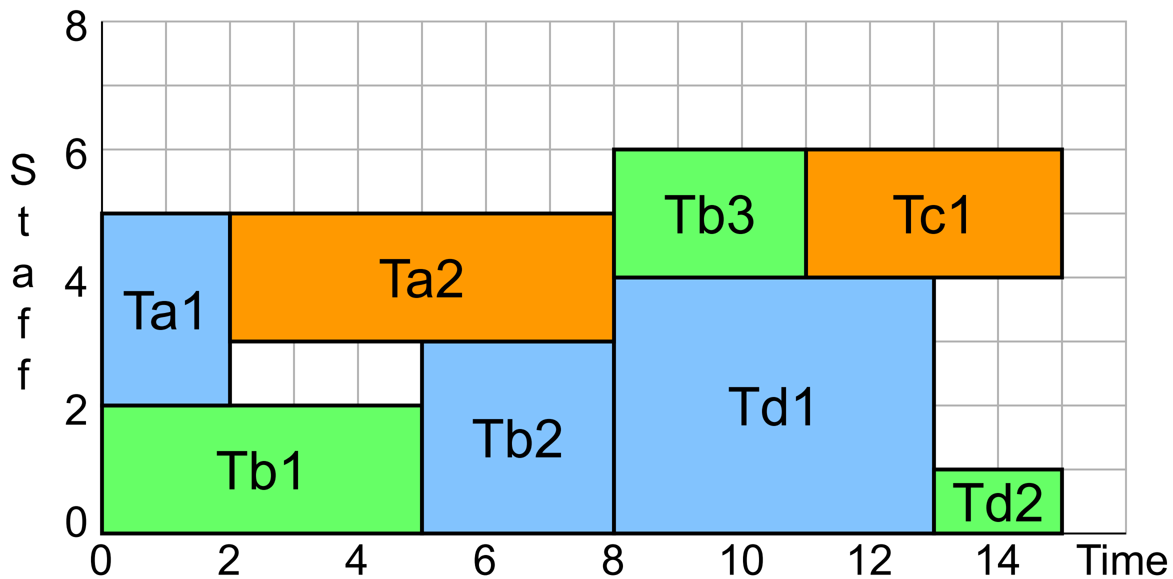


Schedule Refinement

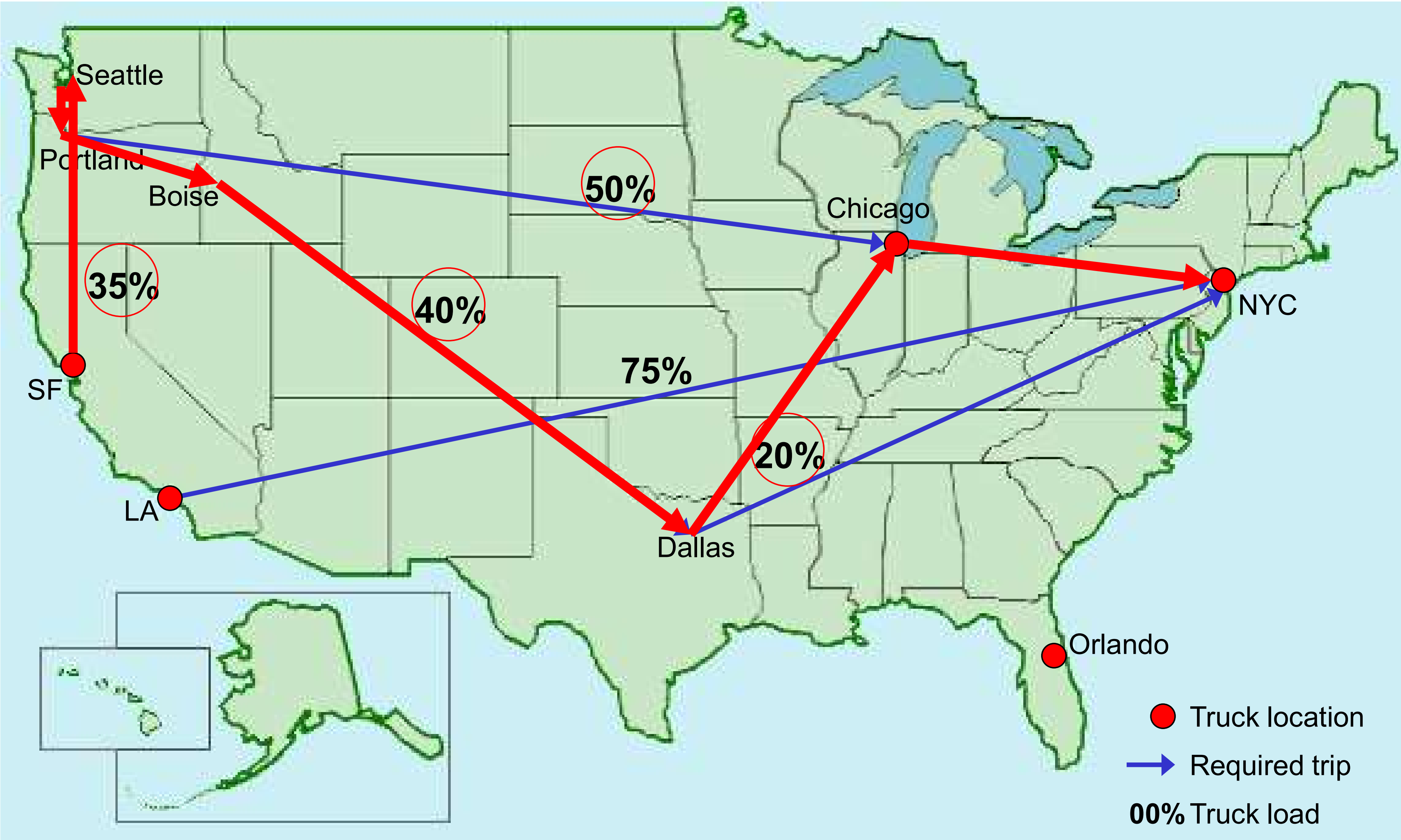


Job	Task	Machine	Time	Staff
Ja	Ta1	M1	2	3
Ja	Ta2	M3	6	2
Jb	Tb1	M2	5	2
Jb	Tb2	M1	3	3
Jb	Tb3	M2	3	2
Jc	Tc1	M3	4	2
Jd	Td1	M1	5	4
Jd	Td2	M2	2	1

M1 M2 M3



Truck Scheduling



Multiprocessor Scheduling

Task graph with unit-time tasks

Here's a heuristic known as list scheduling:

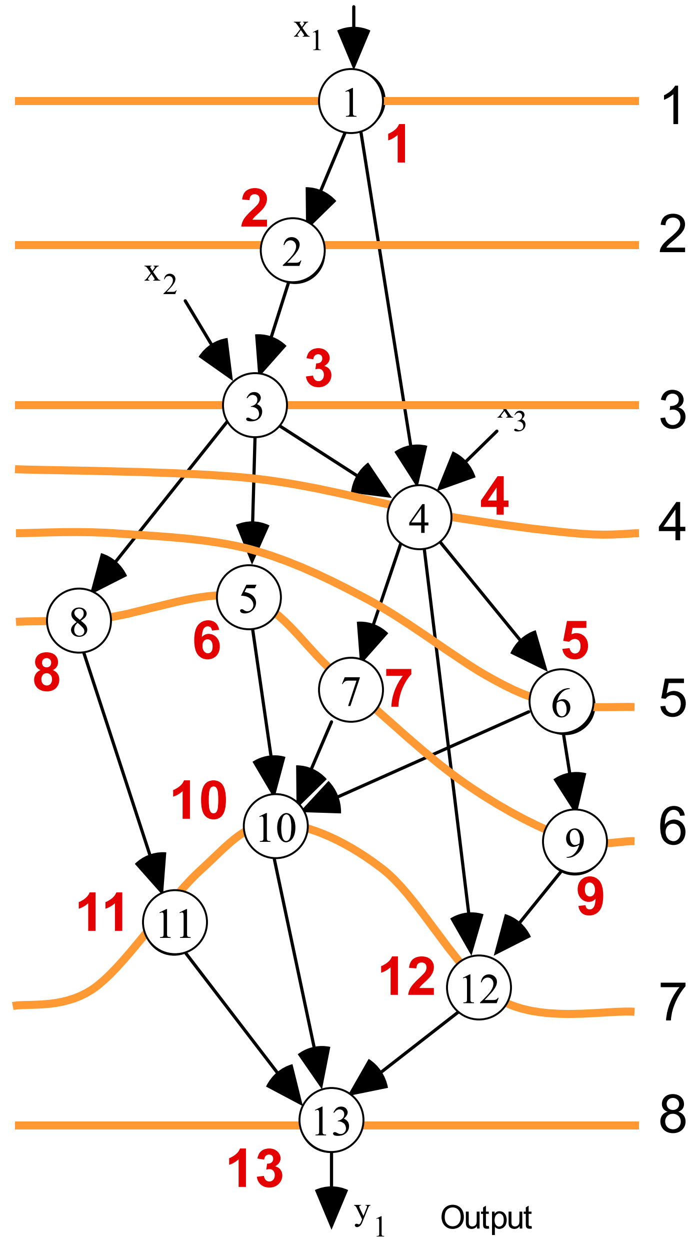
1. Find the depth T_∞ of the task graph
2. Take T_∞ as a goal for the running time T_p
3. Determine the latest possible start times
4. Assign priorities in order of latest times

T_∞ = 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

Time Step	1	2	3	4	5	6	7	8	9	10	11	12	13
P_1	1	2	3	4	6	5	7	8	9	10	11	12	13

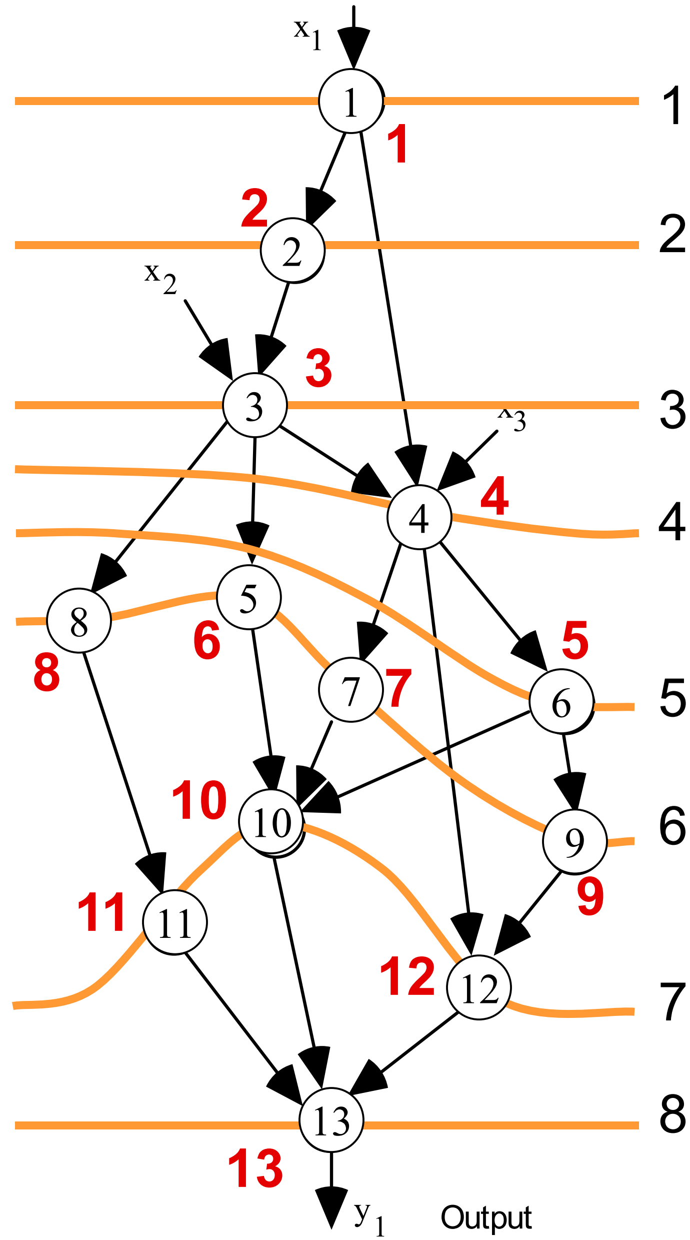
Time Step	1	2	3	4	5	6	7	8	9
P_1	1	2	3	4	6	8	10	12	13
P_2				5	7	9	11		

Time Step	1	2	3	4	5	6	7	8
P_1	1	2	3	4	6	9	12	13
P_2				5	7	10		
P_3				8	11			

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

Time Step

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



Two Related and Similar Problems

The knapsack problem

We have storage capacity *W* and *n* files of sizes w_i and values v_i

Pick a max-value subset of files that fit in the storage space *W*

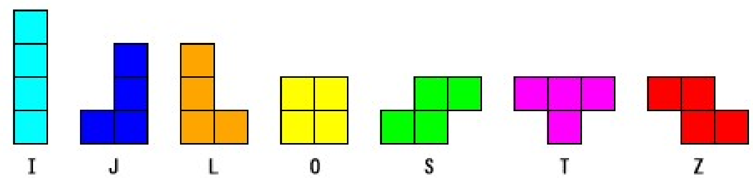
Files cannot be broken into pieces
Naïve solution: Examine all 2^n subsets
Dynamic programming solution
Various heuristic aids
Approx. solutions (say, 90% of optimal)

Off-line game of Tetris

We have a rectangular bin and a sequence of tetrominos

Find optimal play to maximize the number of pieces used

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