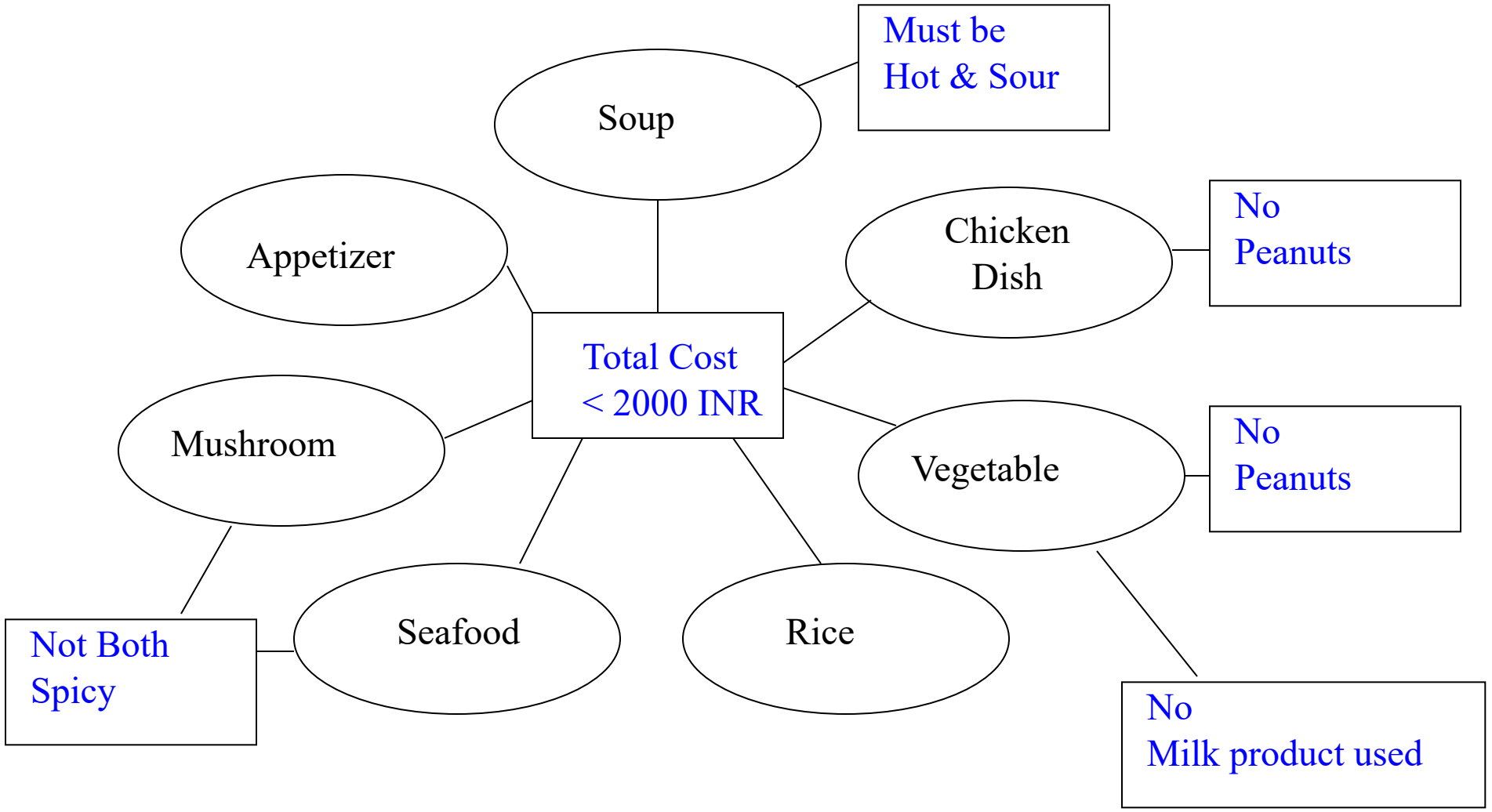
ARTIFICIAL INTELLIGENCE (AI)
CONSTRAINT SATISFACTION PROBLEM (CSP)

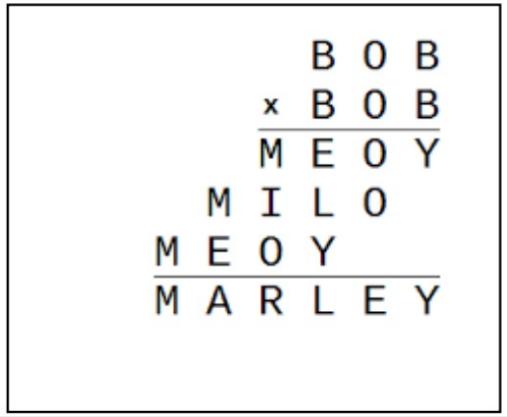
Shreya Ghosh
Assistant Professor, Computer Science and Engineering
IIT Bhubaneswar

CONSTRAINT SATISFACTION PROBLEMS

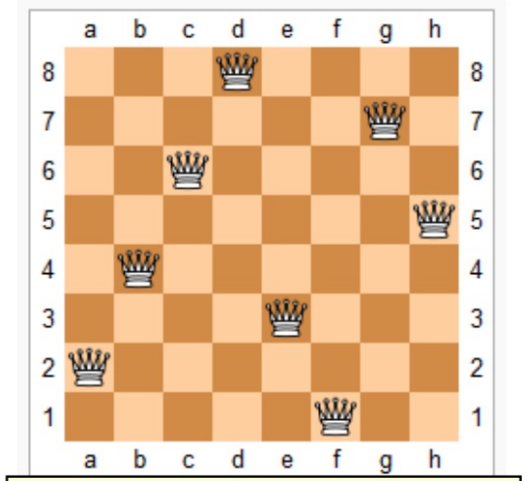


Constraint Network

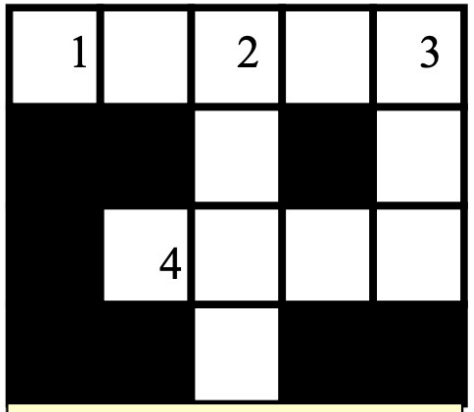
Constraint Satisfaction Problems (CSPs)



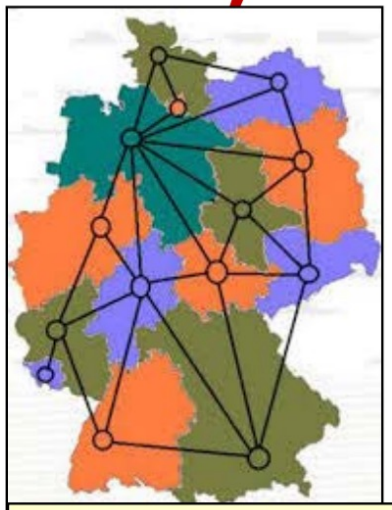
CRYPTARITHMETIC PUZZLE



N-QUEENs

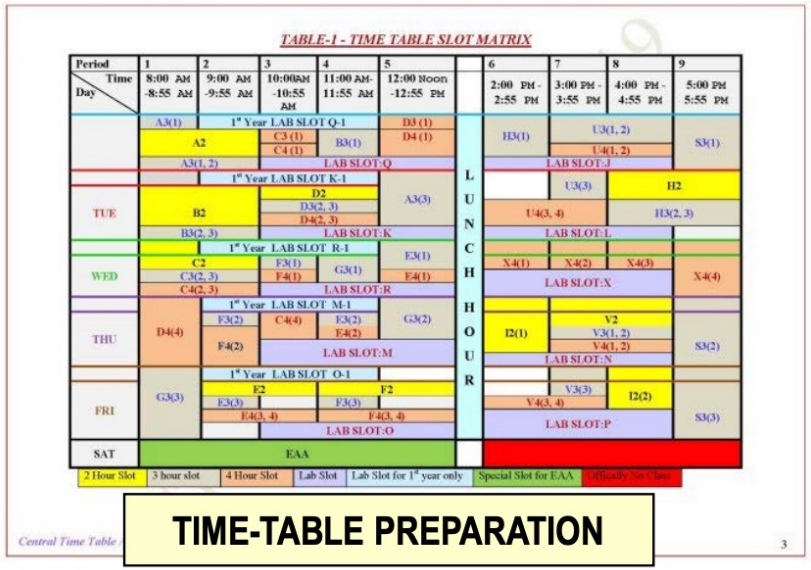


CROSSWORD PUZZLE

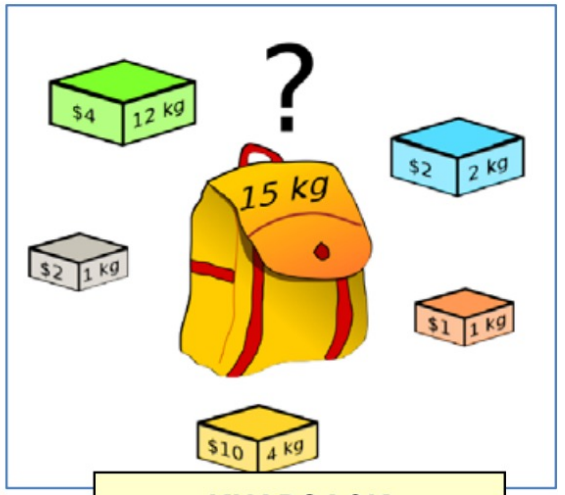


MAP COLOURING

AIRLINE GATE SCHEDULING



TIME-TABLE PREPARATION

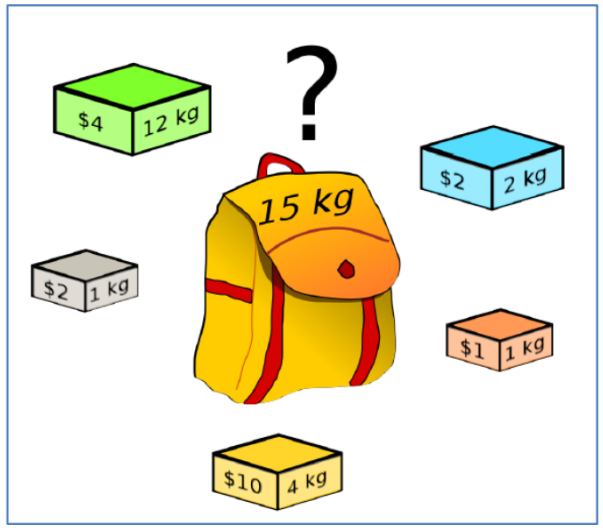


KNAPSACK

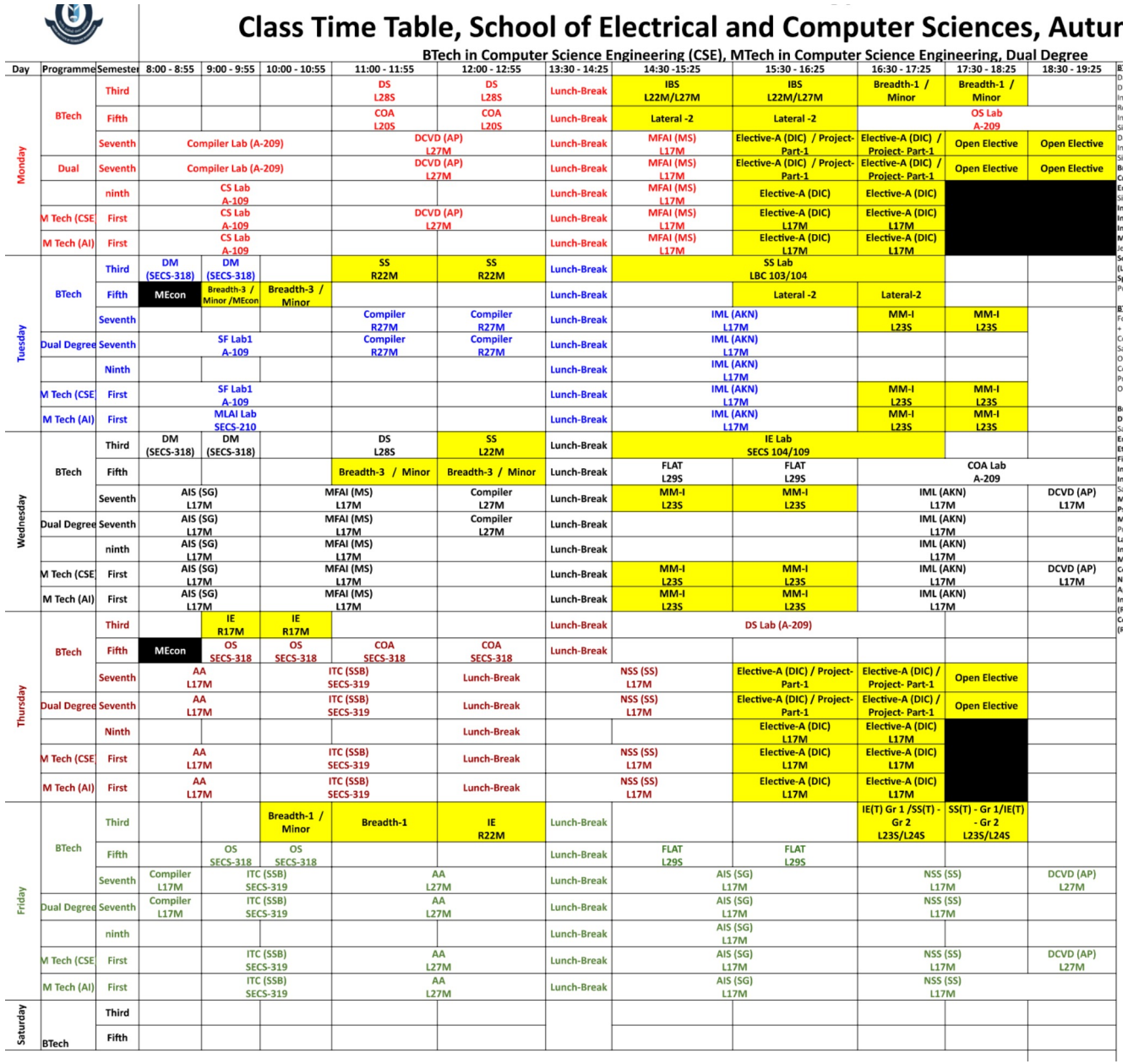
CSP Graph for Airline Gate Scheduling

Flight No	Dep Time	G Start	G End
F1	7:00	6:15	7:15
F2	8:30	7:45	8:45
F3	7:45	7:00	8:00
F4	9:45	9:00	10:00
F5	10:00	9:15	10:15
F6	9:00	8:15	9:15
F7	11:00	10:15	11:15

Formulating CSPs: Knapsack



1. VARIABLES
2. DOMAINS
3. SATISFACTION CONSTRAINTS
4. OPTIMIZATION CRITERIA
5. SOLUTION



Class Time Table, School of Electrical and Computer Sciences, Autur

BTech in Computer Science Engineering (CSE), MTech in Computer Science Engineering, Dual Degree

Day	Programme	Semester	8:00 - 8:55	9:00 - 9:55	10:00 - 10:55	11:00 - 11:55	12:00 - 12:55	13:30 - 14:25	14:30 -15:25	15:30 - 16:25	16:30 - 17:25	17:30 - 18:25	18:30 - 19:25
Monday	BTech	Third				DS L28S	DS L28S	Lunch-Break	IBS L22M/L27M	IBS L22M/L27M	Breadth-1 / Minor	Breadth-1 / Minor	
		Fifth				COA L20S	COA L20S	Lunch-Break	Lateral -2	Lateral -2	OS Lab A-209		
		Seventh	Compiler Lab (A-209)			DCVD (AP) L27M		Lunch-Break	MFAI (MS) L17M	Elective-A (DIC) / Project-Part-1	Elective-A (DIC) / Project- Part-1	Open Elective	Open Elective
	Dual	Seventh	Compiler Lab (A-209)			DCVD (AP) L27M		Lunch-Break	MFAI (MS) L17M	Elective-A (DIC) / Project-Part-1	Elective-A (DIC) / Project- Part-1	Open Elective	Open Elective
		ninth	CS Lab A-109					Lunch-Break	MFAI (MS) L17M	Elective-A (DIC)	Elective-A (DIC)		
	M Tech (CSE	First	CS Lab A-109			DCVD (AP) L27M		Lunch-Break	MFAI (MS) L17M	Elective-A (DIC) L17M	Elective-A (DIC) L17M		
	M Tech (AI)	First	CS Lab A-109					Lunch-Break	MFAI (MS) L17M	Elective-A (DIC) L17M	Elective-A (DIC) L17M		
Tuesday	BTech	Third	DM (SECS-318)	DM (SECS-318)		SS R22M	SS R22M	Lunch-Break	SS Lab LBC 103/104				
		Fifth	MEcon	Breadth-3 / Minor /MEcon	Breadth-3 / Minor			Lunch-Break		Lateral -2	Lateral-2		
		Seventh				Compiler R27M	Compiler R27M	Lunch-Break	IML (AKN) L17M		MM-I L23S	MM-I L23S	
	Dual Degree	Seventh	SF Lab1 A-109			Compiler R27M	Compiler R27M	Lunch-Break	IML (AKN) L17M				
		Ninth						Lunch-Break	IML (AKN) L17M				
	M Tech (CSE	First	SF Lab1 A-109					Lunch-Break	IML (AKN) L17M		MM-I L23S	MM-I L23S	
	M Tech (AI)	First	MLAI Lab SECS-210					Lunch-Break	IML (AKN) L17M		MM-I L23S	MM-I L23S	
Wednesday	BTech	Third	DM (SECS-318)	DM (SECS-318)		DS L28S	SS L22M	Lunch-Break	IE Lab SECS 104/109				
		Fifth				Breadth-3 / Minor	Breadth-3 / Minor	Lunch-Break	FLAT L29S	FLAT L29S	COA Lab A-209		
		Seventh	AIS (SG) L17M		MFAI (MS) L17M		Compiler L27M	Lunch-Break	MM-I L23S	MM-I L23S	IML (AKN) L17M		DCVD (AP) L17M
	Dual Degree	Seventh	AIS (SG) L17M		MFAI (MS) L17M		Compiler L27M	Lunch-Break			IML (AKN) L17M		
		ninth	AIS (SG) L17M		MFAI (MS) L17M			Lunch-Break			IML (AKN) L17M		
	M Tech (CSE	First	AIS (SG) L17M		MFAI (MS) L17M			Lunch-Break	MM-I L23S	MM-I L23S	IML (AKN) L17M		DCVD (AP) L17M
	M Tech (AI)	First	AIS (SG) L17M		MFAI (MS) L17M			Lunch-Break	MM-I L23S	MM-I L23S	IML (AKN) L17M		
Thursday	BTech	Third		IE R17M	IE R17M			Lunch-Break	DS Lab (A-209)				
		Fifth	MEcon	OS SECS-318	OS SECS-318	COA SECS-318	COA SECS-318	Lunch-Break					
		Seventh	AA L17M		ITC (SSB) SECS-319		Lunch-Break		NSS (SS) L17M	Elective-A (DIC) / Project-Part-1	Elective-A (DIC) / Project- Part-1	Open Elective	
	Dual Degree	Seventh	AA L17M		ITC (SSB) SECS-319		Lunch-Break		NSS (SS) L17M	Elective-A (DIC) / Project-Part-1	Elective-A (DIC) / Project- Part-1	Open Elective	
		Ninth					Lunch-Break			Elective-A (DIC) L17M	Elective-A (DIC) L17M		
	M Tech (CSE	First	AA L17M		ITC (SSB) SECS-319		Lunch-Break		NSS (SS) L17M	Elective-A (DIC) L17M	Elective-A (DIC) L17M		
	M Tech (AI)	First	AA L17M		ITC (SSB) SECS-319		Lunch-Break		NSS (SS) L17M	Elective-A (DIC) L17M	Elective-A (DIC) L17M		
Friday	BTech	Third			Breadth-1 / Minor	Breadth-1	IE R22M	Lunch-Break			IE(T) Gr 1 /SS(T) - Gr 2 L23S/L24S	SS(T) - Gr 1/IE(T) - Gr 2 L23S/L24S	
		Fifth		OS SECS-318	OS SECS-318			Lunch-Break	FLAT L29S	FLAT L29S			
		Seventh	Compiler L17M	ITC (SSB) SECS-319		AA L27M		Lunch-Break	AIS (SG) L17M		NSS (SS) L17M		DCVD (AP) L27M
	Dual Degree	Seventh	Compiler L17M	ITC (SSB) SECS-319		AA L27M		Lunch-Break	AIS (SG) L17M		NSS (SS) L17M		
		ninth						Lunch-Break	AIS (SG) L17M				
	M Tech (CSE	First		ITC (SSB) SECS-319		AA L27M		Lunch-Break	AIS (SG) L17M		NSS (SS) L17M		DCVD (AP) L27M
	M Tech (AI)	First		ITC (SSB) SECS-319		AA L27M		Lunch-Break	AIS (SG) L17M		NSS (SS) L17M		
Saturday	BTech	Third											
		Fifth											

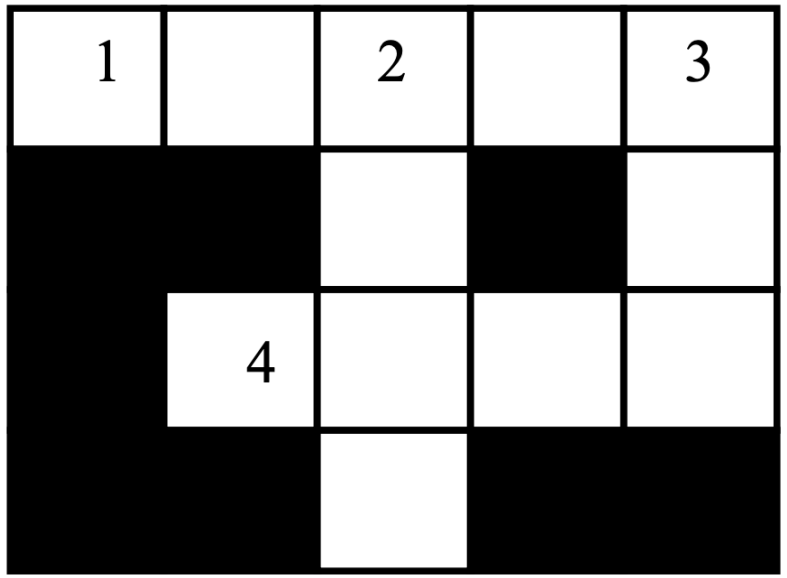
Slots, Rooms, Subjects, Teachers, Students

Room-Slots: Subjects
Subjects: L-T-P, Teachers, Students

Multi-layered constraints

Intricate Optimization

CSP Graph for Crossword



Word List:

astar, happy, hello, hoses, live, load, loom, peal, peel, save, talk, ant, oak, old

Constraint Propagation Steps

- **Constraints**
  - Unary Constraints or Node Constraints
  - Binary Constraints or Edges between CSP Nodes
  - Higher order or Hyper-Edges between CSP Nodes
- **Node Consistency**
  - For every Variable V_i, remove all elements of D_i that do not satisfy the Unary Constraints for the Variable
  - First Step is to reduce the domains using Node Consistency
- **Arc Consistency**
  - For every element x_ij of D_i, for every edge from V_i to V_j, remove x_ij if it has no consistent value(s) in other domains satisfying the Constraints
  - Continue to iterate using Arc Consistency till no further reduction happens.
- **K-Consistency or Path Consistency**
  - For every element y_ij of D_i, choose a Path of length L with L variables, use a consistency checking method similar to above to reduce domains if possible

Arc Consistency Algorithm AC-3

```
AC-3(csp) // inputs - CSP with variables, domains, constraints
1.   queue ← local variable initialized to all arcs in csp
2.   while queue is not empty do
3.       (Xi , Xj) ← pop(queue)
4.       if Revise(csp, Xi, Xj) then
5.           if size of Di = 0 then return false
6.           for each Xk in Xi.neighbors-{Xj} do
7.               add (Xk , Xi) to queue
8.   return true
```

```
Revise(csp, Xi, Xj)
1.   revised ← false
2.   for each x in Di do
3.       if no value y in Dj allows (x, y) to satisfy constraint between Xi and Xj then
4.           delete x from Di
5.           revised ← true
6.   return revised
```

Backtracking Algorithm for CSP

CSP-BACKTRACKING({})

CSP-BACKTRACKING(a)

- If a is complete then return a
- X ← select unassigned variable
- D ← select an ordering for the domain of X
- For each value v in D do
  - If v is consistent with a then
    - Add (X= v) to a
    - result ← CSP-BACKTRACKING(a)
    - If result ≠ *failure* then return result
- Return *failure*

Strategies for CSP Search Algorithms

- Initial Constraint Propagation
- Backtracking Search
  - Variable Ordering
    - Most Constrained Variable / Minimum Remaining Values
    - Most Constraining Variable
  - Value Ordering
    - Least Constraining Value leaving maximum flexibility
  - Dynamic Constraint Propagation Through Forward Checking
    - Preventing useless Search ahead
  - Dependency Directed Backtracking
- SAT Formulations and Solvers
- Optimization
  - Branch-and-Bound
  - SMT Solvers, Constraint Programming
- Learning, Memoizing, etc
- CSP Problems are NP-Hard in General

<DIY> FLIGHT SCHEDULING PROBLEM IN SAT FORM