Assignment : 2

Zarin Tasnim Haider

ID : 21301380

Section : 08

CSE : 422

Conceptual Questions

1) A constraint Satisfaction problem (CSP) is a computational problem where variables constrained by permissible values.

Components:

i) Variables: Entities needing values.

ii) Domains: Allowable values per variable.

iii) Constraints: Define variable relationships, limiting feasible value combinations.

2) Backtracking Algorithm explores variable assignment backtracking upon dead ends. It undoes recent assignments, tries alternatives until solution or exhaustion. ~~For example,~~

3) Forward checking in CSPs involves examing the remaining domains of variables after a variable is assigned a value, and it removes value from these domains that are no longer consistent with constraints. Forward checking enhance the basic backtracking algorithm by reducing the search space early on, potentialy preventing the

exploration of branches that lead to dead end. For example: if assigning a value to a variable reduces the domain of another variable to an empty set, it indicates that the current assignment cannot lead to a solution.

(4) Heuristics aid CSP solving by guiding variable and value selection. Common heuristics include MRV and MCV for variable selection, and LCV and first fail for value selection in backtracking algorithms.

(5) Global constraints in CSPs apply to multiple variables, while binary constraints involve pairs of variables. An example of global constraint is the "All-different" constraint, ensuring all variables in its scope take distinct values, which can simplify modeling and reduce search space.

⑥ Constraints in CSP difine relationships between variables, shaping its structure and complexity. The nature of constraints impacts difficulty: complex constraints increase intricacy, while simpler ones ease problem-solving, Overly restrictive constraints may limit feasible solutions, heightening difficulties, whereas looser constraints offer more flexibility, potentially simplifying the problem.

⑦ Heuristic impact the search process in solving CSPs by guiding the selection of variable and values, as well as the exploration of the solution space. While heuristics can improve search efficiency by focusing on promising regions of the search space, relying too heavily on heuristics may lead to suboptimal solutions or increased computational complexity in some cases.

(8) The efficieny of the backtracking algo with and without forward checking depends on the specific problem instance and the nature of the constraints. Forward checking can significally improve backtracking performance when applied judiciously especialy in problems with large search space or complex constraints. It helps to prune the seach space early, reducing the likelihood of exploring fruitless paths and potentially leading to faster convergence towards a solution.

Scanned with CamScanner

Ans to the Q. NO-01

Courses = {C1, C2, C3, C4, C5, C6, C7, C8, C9, C10}

Days = {D1, D2, D3, D4, D5}

Time slots = {T_1, T_2, T3, T4}

Instructor = {I1, I2, I3, I4, I5}

Domains :- Day-time and instruction takes values.

Constant :

i) No inst can teach not more than two course.

ii) Avoid 2 CSE courses on the same day.

	T_1	T_2	T_3	T_4
D_1	$C_1 I_1$		$C_2 I_2$	
D2		$C_3 I_1$		
D3	$C_5 I_3$		$C_6 I_5$	
D4		$C_7 I_4$		$C_8 I_5$
D5	$C_9 I_3$		$C_{10} I_4$	

Scanned with CamScanner

Ans to the Q. No-2

Variables:

Start_time (Product, Stage)

Domains:

Each stage can be start on the hour from 4 Am to 8 pm. with which stage lasting 2 hours.

Constraints

1) Each stage can only handle one Product at a time.

2) Each product must go through the stages in the order: mixing → baking → packaging

3) Time Limitation must be ready for sale by 8 am for bread and cookies and by 12 pm for cake.

Time	4-6 am	6-8 am	8-10 am	10-12 pm
Bread	Mixing	Baking	Packaging	
Cake	"	"	"	
Cookies	"	"	"	

Ans to the ans 3

Variable:
Start_time = (Project, task)

Domains

Each task can start in 4 hour blocks from 9 am to 5 pm over 2days.

Constraints

→ Tasks for each project must be completed in order : coding → testing → deployment.

→ One one project can be in testing stage at any given time.

→ No project can have more than one task scheduled per day.

Simulation

Day 1

Time	9-1pm	1-5 pm
Project A	coding	testing
Project B		Coding

Day 2

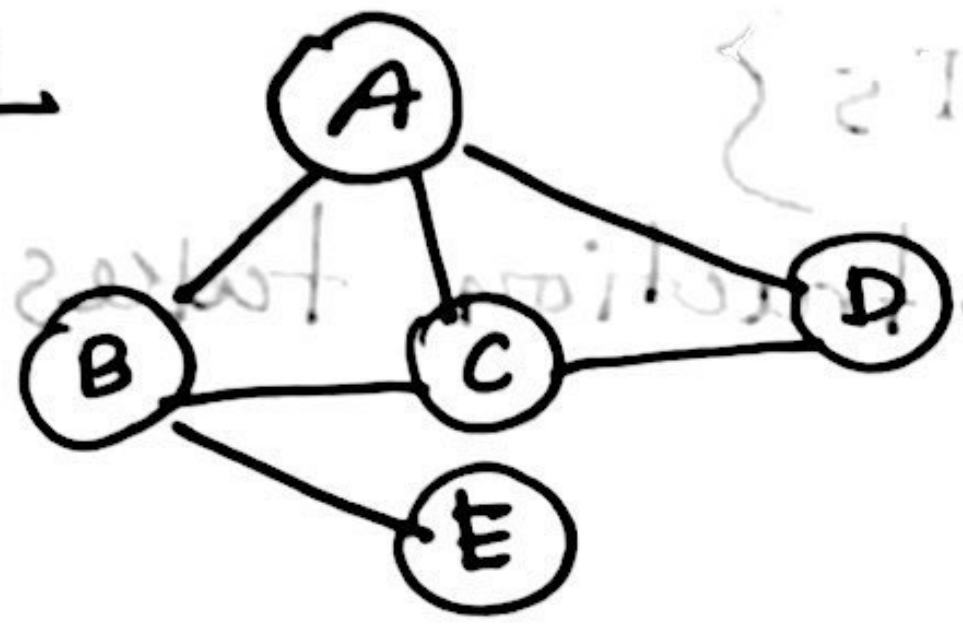
Time	9-1pm	1-5p
Project A	Deployment	
Project B	testing	Deployment

Ans to the Q. NO 4

Variable = { A, B, C, D, E }

Domains = { Red , Green, Blue , Yellow, Orange }

Graph



[No. two adjacent region will share same color]

Simulation

A B C D E

RGBYO ~~R~~GBYO ~~R~~GBYO ~~RGBYO~~ RGBYO RG~~BYO~~

R GBYO ~~GBYO~~ GBYO

A	B	C	D	E
RGBYO	~~R~~GBYO	~~R~~GBYO	~~R~~GBYO	RGBYO
R	GBYO	GBYO	GBYO	RGBYO
R	G	BYO	G~~B~~YO	RBYO
R	G	B	GYO	RBYO
R	G	B	G	RBYO
R	G	B	G	R

Ans to the Q. NO 5

Variable = { course , days , time , labroom , TA }

Domains = { 0 , 1 }
- 1 → scheduled
- 0 → not "

Constraints

1) TA can't take two labs at the same time.
2) Each course can be scheduled only once.
3) Each lab = 2 hours.
4) One lab = 1 sesson.
5) Software base coure must be held in labs with that specific software

	T_1	T_2
D1	C1	C2
D2	C3	C5
D3	C4	
D4		
D5		

Ans to the Q. NO - 6

sub = { math, Science , English }

teacher = { Mr. A , Mr. B , Mr. C }

Students Group = { G_1 , G_2 , G_3 }

slots = { T_1 , T_2 , T_3 }

T_1 = 9 am

T_2 = 10 am

T_3 = 11 am

Simulation

	T_1	T_2	T_3
G_1	1	1	1
G_2	1	1	0
G_3	0	1	1

0 → doesn't do the class

1 → does the class

constaints
i) G1 needs to attend all 3 sub.
ii) G2 needs to attend Math and Science
iii) G3 " " " Endish and 1[illegible]

Ans to the Q. NO - 7

workshop = {W1, W2, W3, W4}

time slots = {T1, T2, T3}

Domains = {AG1, AG2, AG3}

T_1 = 9-10:30 Am

T_2 = 11-12:30 pm

T_3 = 2-3:30 pm

Constains

AG1 wants to attend W_1 and W2.

AG2 wants to attend W2 and W3.

AG3 " " " W3 " W4.

	T_1	T_2	T_3
W1	AG1		
W2	AG3	AG1	AG2
W3	AG2	AG3	AG1
W4		AG2	AG3

Ans to the Q·NO - 8

Variable : $\{P_1, P_2, P_3, P_4, P_5, P_6\}$

Domain : $\{A, B, C, D, E, F\}$

Constraints

~~BA~~

Book A next to Book B.

Book C " " Book D.

Book E at one end.

P_1	P_2	P_3	P_4	P_5	P_6
ABCDEF	ABCDEF	ABCD~~E~~F	ABCD~~E~~F	ABCD~~E~~F	ABCD~~E~~F
E	~~A~~BCD~~E~~F	~~A~~B~~CDF~~	A~~BCDF~~	~~AB~~CDF	~~AB~~CDF
E	CDF	B	A	CDF	CDF
E	CDF	B	A	~~CDF~~	~~CD~~F
E	C~~DF~~	B	A	D	F
E	C	B	A	D	F

Book Problems :

6.1

with 3 colors

1st Node = 3 possibilities

2nd " = 2 "

Everyother nodes can be colored by one color.

Tasmanis isn't connected, so it will take any.

∴ Total solution : $3 \times 2 \times 3 = 18$

4 color

1st node = 4 possibilities

2nd " = 3 "

3rd " = 2 "

Remaining = 1 "

Tasmania = any color

Total solution = $4 \times 3 \times 2 \times 3 = 96$

2 color

No solution.

As each node has 2 or more adjacent, so the constraint can't be satisfied.

6.2

a) Variable will be corresponding to each of the chess board.

b) Domain for each : {empty, occupits}

c) Every pair of squares separated by a knight's move is constrained. Both squares in each pair can't be occupied simultaneously.

- The entire set of square is constraind such that the total number of occupied squares should be K.

d) For simulated annealing, the successor function must completely connect the space for random restart. The good state must be reachable by hillclimbing for some initial state. Two basic classes of solution,

i) Ensuring no attack. Action are to move a knight to any unattacked square.

ii) Allow attacks but try to get rid of them. Action are to move a knight to any square.

a) Using Depth First Search we can solve this. Each successor will fill in a word in the puzzle with one of the words in the dictionary.

Heuristics

Let the slot length be in desending order because normally there are less longer words.
And it is better to go one word at a time to minimize the number of steps.

b) As a CSP, the variables should be words. If we choose letter as variables the the cost for searching and filling a slot will increase.

6.4

a) For large rectangle: width = W height = H.

Each small rectangle has 4 variables: x, y, w, h

The constraints :

$R_i, x \geq 0$

$R_i x + R_i, w \leq W$

$R_i, y \geq 0$

$R_i, y + R_i, h \leq H$

b) The could be 3 variables for each class.

i) One with time.

ii) one with classroom.

iii) One with teachers.

constraints only one class can be in same classroom at the same time, teachers can teach one class at a time.

c) Have one variable for each step one the tour. The requirement is neighbouring cities to be connected by roads. All diff. constraint variables have diff value.

6.7

The problem can be represented as CSP by taking a variable for each color, pet, drink, country and cigarette brand. The values of each variable will be 1-5 indicating each house number. Besides, ease of expressing a problem, the other reason to choose this representation is the efficiency of finding a solution.

Another representation is to have five variables for each house. One with the domain of color, one with pet and so on.

6.8

a) $A1 = R$

b) $H = R$ conflicts with $A1$

c) $H = G$

D) $A4 = R$

E) $F1 = R$

f) $A2 = R$. conflicts with $A1$, $A2 = G$ conflicts with H, so $A2 = B$

g) $F2 = R$.

h) $A_3 = R$ conflicts with A_4, $A_3 = G$ conflicts with H.
$A_3 = B$ " " A_2.
∴ Back track.
Conflict set is $\{A_2, H, A_1\}$ so jump to A_2 and add $\{H, A_4\}$ to A_2's conflict set.

i) A_2 has more values. So again backtrack. Conflict set $\{A_1, H, A_4\}$ so jump back to A_4, Add $\{A_1, H\}$ to A_4's conflict set.

j) $A_4 = G$ conflicts with h, so $A_4 = B$

k) $F_1 = R$.

l) $A_2 = R$ conflicts with A_1, $A_2 = G$ conflicts with H, so, $A_2 = B$.

m) $F_L = R$

n) $A_3 = R$.

o) $T = R$ conflicts with F_1 and F_2, ~~T~~ $T = G$ conflicts with G, so $T = B$.

p) Success.

6.16

- A constraint is a restriction on the possible values of two or more variables.
- Backtracking search is a form of dfs in which there is a single pepresentation of the state that gets updated for each successor and then must be restored when a dead end is reached.
- Arc consistent is a directed arc from variable A to B if for ~~each~~ every value in the current domain of A, there is some consistent value of B.
- Backjumping is jumping back more than one level when dead end is reached.
- Min-conflicts is a heuristic which says that when given a variable to modify, choose the value the conflicts with the fewest number of other variable.
- Cycle cutset is a set of variables which when removed from the constraint graph make it acyclic.