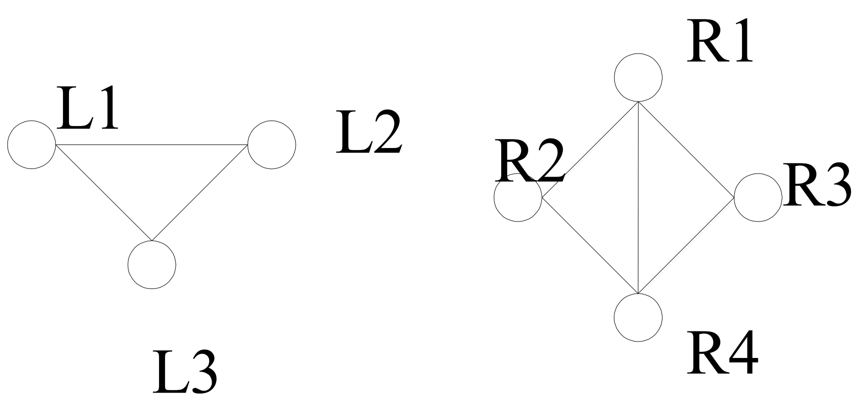
Dynamic Backtracking

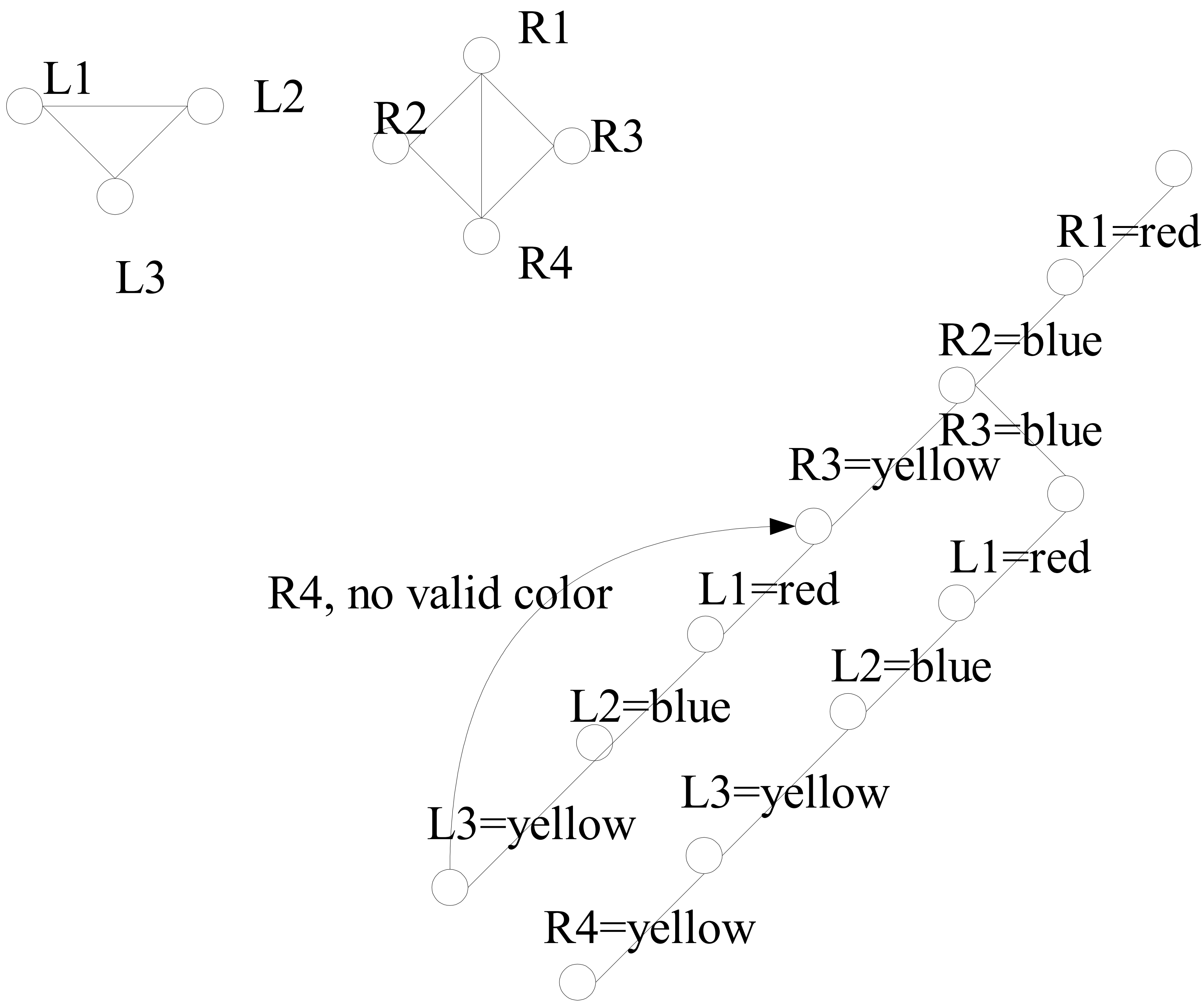
Constraint satisfaction problems

- Consists of:
  - a set of variables
  - A domain of values each variable can take
  - A set of constraints that the variables must satisfy.

Why is dynamic backtracking interesting?

- Consider a map coloring, with 3 colors, on a disconnected map.





L1
L2
L3
R1
R2
R3
R4
R1=red
R2=blue
R3=blue
R3=yellow
L1=red
L1=red
R4, no valid color
L2=blue
L2=blue
L3=yellow
L3=yellow
R4=yellow

- To do backjumping, we already need to save some information on what was already set, so we can tell what the problem was, so we could backtrack to it.
- However, we need to add some more bookkeeping to allow us to be able to tell whether or not a choice was made based on the choice we are jumping back to.

Definitions:

- Partial solution: the set of assignments of variables we currently have, which currently violates no constraints.
- Eliminating Explanations
  - An eliminating explanation for a variable i is an ordered pair (v,E) where v is a value that i can't take, and E is a set of variables that have been set, which cause v to be eliminated.

Elimination mechanism

- An elimination mechanism, denoted by $\varepsilon(P,i)$, is a function that takes two arguments. The first is the partial solution we currently have, and the second is a variable i that we have not set yet.
- This function returns the set of eliminating explanations for i.

Elimination mechanism continued

- An elimination mechanism must satisfy 3 conditions.:
  - Correct: if the value v is not represented in ε(P,i), then every constraint must be satisfied when we set i=v, and add i to the partial solution.
  - Complete: if P is a partial solution which could be extended to a solution using the variable i set to value v, then for any partial solution extended form P where v is eliminated for i, then the variable that caused the problem must be one of the variables that was chosen to extend P.

Properties of eliminating mechanisms

- Concise: For every variable i, partial solution P, and value v, there can be at most one element of the form (v,E) in ε(P,i)

Backjumping

- Now would be a good time to show how backjumping works, using this new notation.

Backjumping algorithm

- Start with no variables set, and all the sets of eliminating explanations empty.
- When we have set every variable, we are finished.
- We now want to choose an unset variable, i. Set $E_i = \varepsilon(P,i)$.
- If there is a value v that has not been eliminated by anything in E_i, set i to v.

Backjumping continued

- If no values are left, let E contain all the variables included in the explanations for each value.
- If E is empty, no solution exists. Otherwise, take the last variable, j, that we set which is in E. Let v be what we set j to. Unset j and all the variables we set after j. Add $(v, E \cap var(P))$ to E_j.
- Now return to the previous slide, and consider the variable j.

Dynamic backtracking

- We start with an empty solution set, and set all the sets of eliminating explanations to empty.
- If we ever have every variable set to a value, then we have our solution.
- We now will select a variable, i, that we have not set yet. Now we will add everything from $\varepsilon(P,i)$ to E_i.

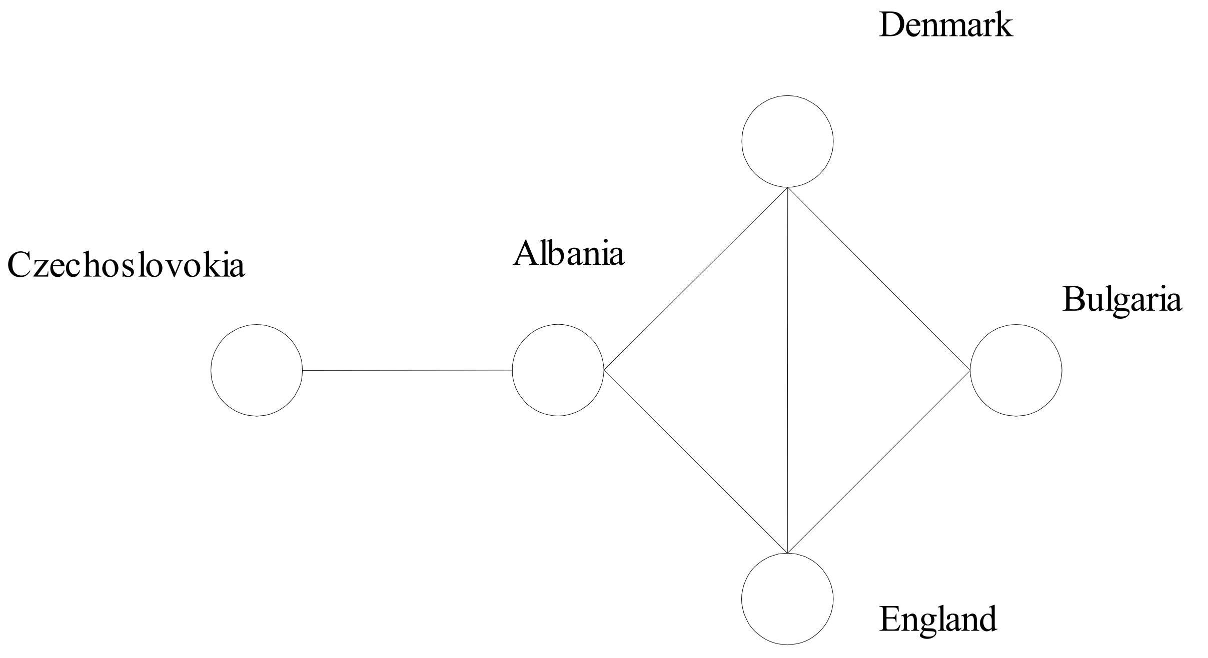
Dynamic Backtracking continued

- If there is a value v that i can take not in E_i, set i to be v, and choose another variable to look at.
- Otherwise, let E be the set of all the variables mentioned in the explanations in E_i.

Dynamic backtracking continued

- If E is empty, then there is no solution. Otherwise, let (j,v) be the last variable that we set, that occurs in E. Now, remove (j,v) from our solution and for every variable k that was set after j, remove any explanation that involved j. Also add (v, $E \cap var(P)$) to E_j, where var(P) denotes all the variables currently set.

Map



Example

- Set Albania to red

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria				
Czechoslovakia				
Denmark				
England				

Example

- Set Bulgaria to yellow

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovakia				
Denmark				
England				

Example

- Set Czechoslovakia to blue

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovak	Blue	A		
Denmark				
England				

We put A in the red column, because Albania is the reason Czechoslovakia can't be red

Example

- Set Denmark to blue

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovak	Blue	A		
Denmark	Blue	A	B	
England				

We put A in the red column, because Albania is the reason Denmark can't be red, and B in the yellow column, because Bulgaria is the reason Denmark can't be yellow.

Example

- Consider England

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovak	Blue	A		
Denmark	Blue	A	B	
England		A	B	D

Since all possible colors England could take are eliminated, we must step back.

Example

- Unset Denmark,

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovak	Blue	A		
Denmark		A	B	
England		A	B	

remove Denmark from any of the eliminating explanations

Example

- Add A,B to the explanation for blue

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Yellow			
Czechoslovak	Blue	A		
Denmark		A	B	A,B
England		A	B	

Example

- Backtrack to Bulgaria

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria			A	
Czechoslovak	Blue	A		
Denmark		A		
England		A		

Remove all explanations that involved Bulgaria, and add Albania as a reason for Bulgaria not being yellow.

Example

- Color Bulgaria Red

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Red		A	
Czechoslovak	Blue	A		
Denmark		A		
England		A		

Example

- Color Denmark Blue

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Red		A	
Czechoslovak	Blue	A		
Denmark	Blue	A,B		
England		A		

Example

- Color England Yellow

Country	Color	Red	Yellow	Blue
Albania	Red			
Bulgaria	Red		A	
Czechoslovak	Blue	A		
Denmark	Blue	A,B		
England	Yellow	A,B		D

Some experimentation

- Dynamic backtracking has been incorporated in a crossword puzzle generation program.
- The program was run on the problem of generating 19 puzzles of size 2x2 to 13x13. They attempted each puzzle 100 times using backjumping, and also using dynamic backtracking.
- When the program backtracked 1000 times, the run was considered a failure.

Results

Frame	Dynamic backtracking	Backjumping	Frame	Dynamic backtracking	Backjumping
1	100	100	11	100	98
2	100	100	12	100	100
3	100	100	13	100	100
4	100	100	14	100	100
5	100	100	15	99	14
6	100	100	16	100	26
7	100	100	17	100	30
8	100	100	18	61	0
9	100	100	19	10	0
10	100	100			