CSP: Solving by Search

Pedro Meseguer
IIIA-CSIC
Bellaterra, Spain

Overview

Systematic Search

- Search Tree
- Backtracking (BT)
- Conflict-Directed Backjumping (CBJ)
- Dynamic Backtracking (DB)
- Heuristics

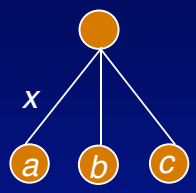
Local search

- Breakout

Search Tree

State space: explored as a tree

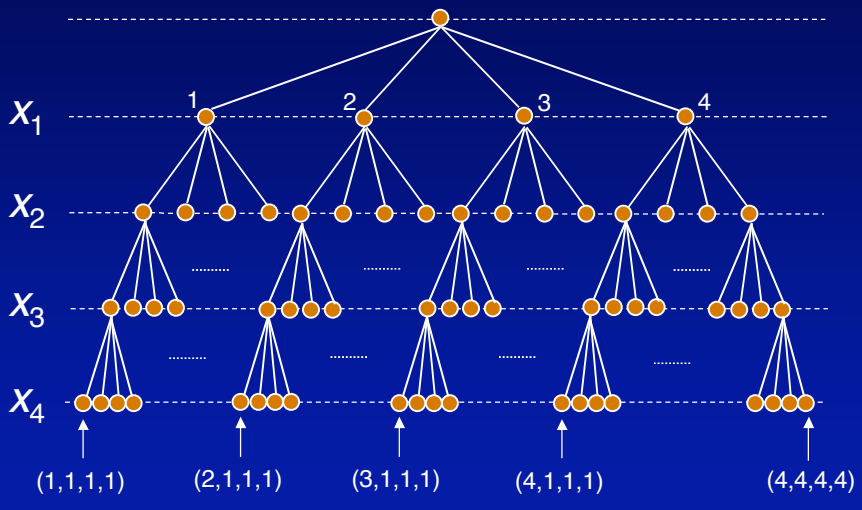
- root: empty
- one variable per level
- sucessors of a node:
  - one sucessor per value of the next level variable
  - meaning: variable value



Tree:

- each branch defines an assignment
- depth *n* (number of variables)
- branching factor *d* (domain size)

Search tree for 4-queens



Backtracking Algorithm

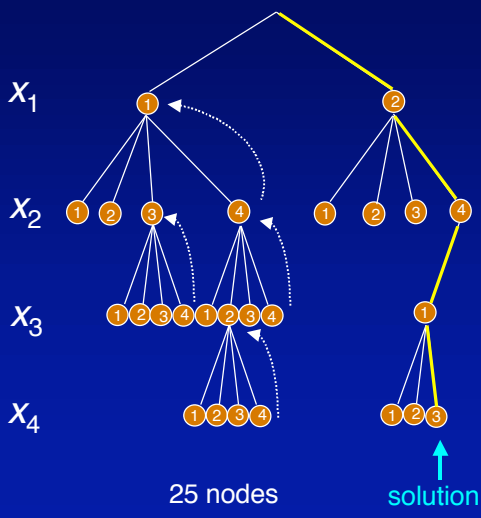
Depth-first tree traversal (DFS)

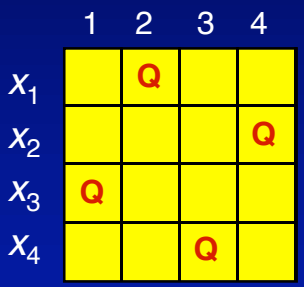
At each node:

- check every completely *assigned* constraint
- if consistent, continue DFS
- otherwise, prune current branch
- continue DFS

Complexity: $O(d^n)$

Backtracking on 4-queens

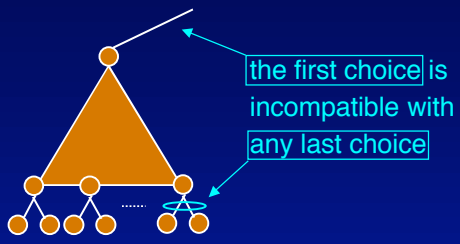




Problems of Backtracking

Thrashing:

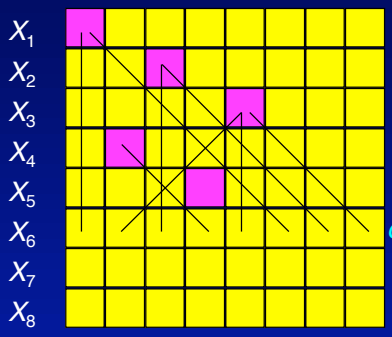
- the same failure can be rediscovered an exponential number of times



Solutions:

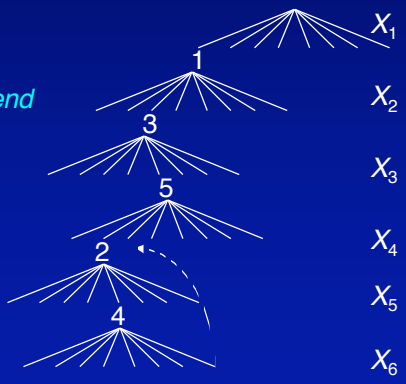
- check not completely assigned constraints: *propagation*
- non-chronological backtracking: *backjumping*

Non-chronological Backtracking



dead-end

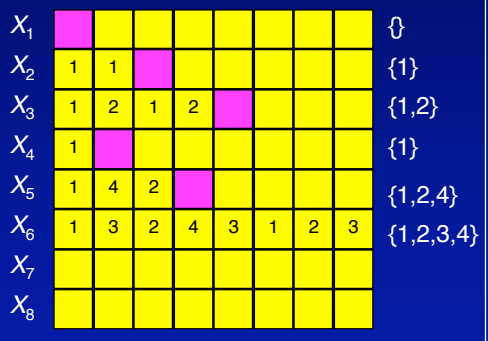
Backtrack on a culprit variable: X_4



Changing X_5 does NOT remove the dead-end

Conflict Set

CS(x_k): assig. variables in conflict with some value of x_k



- Backtrack: jumps to the *last* variable in CS(x_k)
- CS(x_k) is *backed-up*:

x_k □ x_i

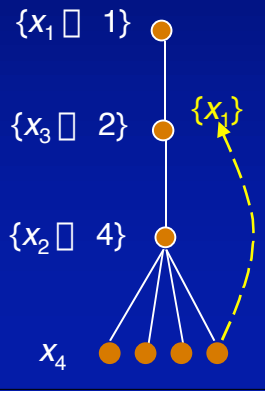
CS(x_i)=CS(x_i) ∪ CS(x_k)-x_i

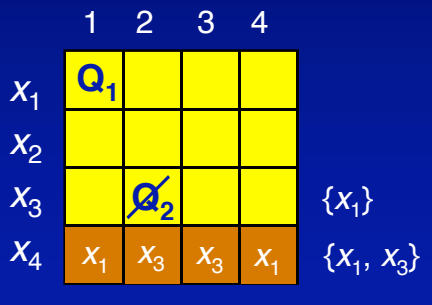
- x_k conflicts with var. before x_i are passed to x_i

Conflict-Directed Backjumping

Non-chronological backtracking:

- jumps to the last variable in the conflict-set
- the conflict set is backed-up
- intermediate decisions are *removed*





Nogoods

Nogood: subset of incompatible assignments

Example: map colouring, x_1, x_2, x_3 adjacent, $D = \{a,b\}$

$(x_1 = a \land x_3 = a)$ or equivalently

$x_1 = a \land x_3 \neq a$

lhs — *rhs*

Nogood resolution:

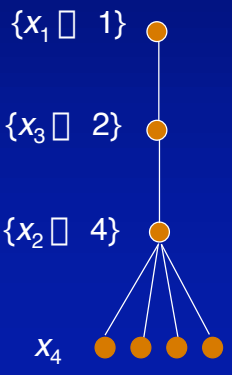
$x_1 = a \land x_3 \neq a$
$x_2 = b \land x_3 \neq b$

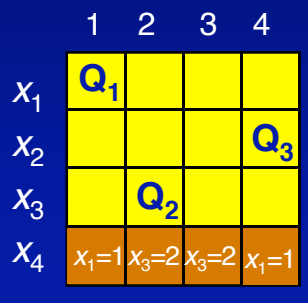
$\Rightarrow$ $x_1 = a \land x_2 \neq b$ (*last variable in lhs*)

Dynamic Backtracking

Non-chronological backtracking:

- one nogood per each incompatible value
- empty domain: new *ng* by nogood resolution
- backtrack to the variable in *rhs*(*ng*)





$x_1{=}1 \land x_4 \neq 1$
$x_3{=}2 \land x_4 \neq 2$
$x_3{=}2 \land x_4 \neq 3$
$x_1{=}1 \land x_4 \neq 4$

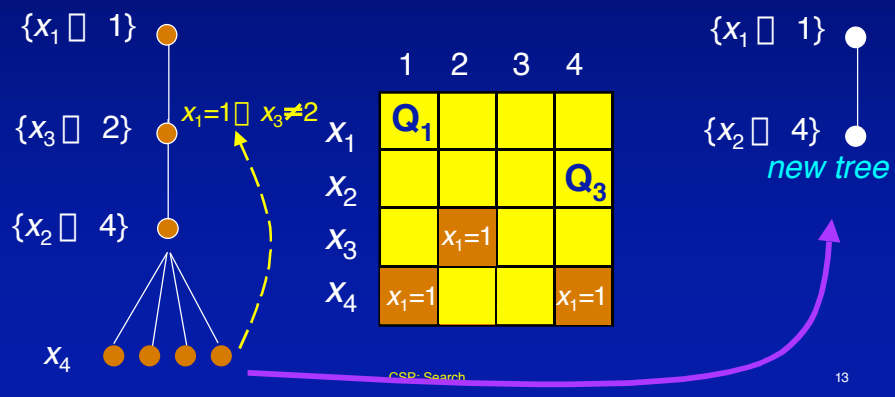
$\Downarrow$

ng: $x_1{=}1 \land x_3 \neq 2$

Dynamic Backtracking (II)

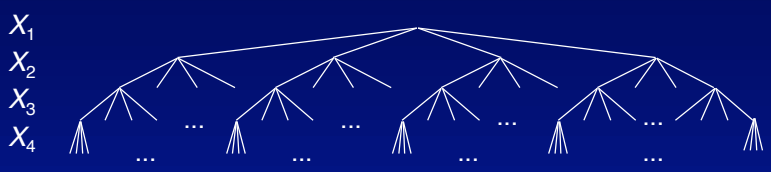
Non-chronological backtracking:

- jumps to the last decision *responsible* for the dead-end
- intermediate decisions are *NOT* removed



Variable Ordering

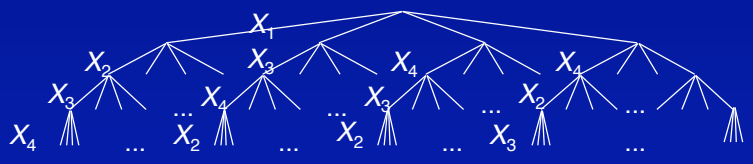
Static variable ordering: variable associated with level



Not needed for an exhaustive tree.

Dynamic variable ordering:

- each branch considers all vars / different ordering per branch



Variable Selection

Node *q*, what is the next variable to assign?

1. There is a solution in *succ*(*q*): *any var is fine*
2. There is no solution in *succ*(*q*): *assign var that sooner detects that there is no solution*

What is more often?

- Except trivial problems, case 2
- *Biggest* effort: escaping from problems without sol.

Fail-first: first the variable that sooner detects failure

Heuristics

Min-domain:

- First the variable with less compatible values with the current partial solution
- Minimize the search tree

Max-degree:

- First the variable involved in more constraints
- Maximize constraint propagation

Combination:

- First the variable with min domain/degree