Search techniques in Constraint Programming

Hadrien Cambazard (from doctoral class of Emmanuel Hebrard, LAAS, Toulouse)

G-SCOP, LAAS-CNRS
Toulouse

Outline

1. **Backtrack Search**
   - Variable Ordering
   - Value Ordering (Branching)
   - Restarts & Randomization

2. **Discrepancy Search techniques**

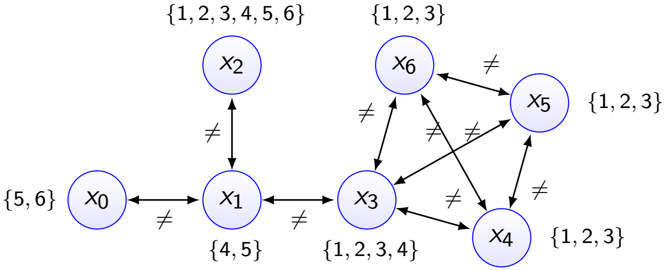
Search Algorithms

- Tree Search
  - Given a decision (a constraint) for example $x = v$, either:
    - ⋆ there exists a solution where $x = v$
    - ⋆ there exists a solution where $x \neq v$
    - ⋆ there is no solution
  - Typical decisions (fix a value, split a domain, enforce a precedence, ...)
  - Exploring both branches gives a complete algorithm
    - ⋆ In practice, it is often possible to detect inconsistencies after assigning only a few variables
    - ⋆ Before each decision, there is a propagation step to enforce local consistencies

Variable Ordering

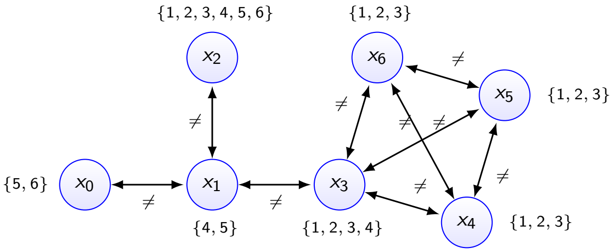
- The order in which variables are explored matters!
  - During search, most of the time is spent in unsatisfiable sub-trees
  - Detecting failure early (fail first)
  - Start with variables that are most likely to "fail"
    - Smallest domain size (less freedom, smaller branching factor)
    - Maximum degree (most constrained variable)
    - Minimum $\frac{\text{domain size}}{\text{degree}}$
    - ...

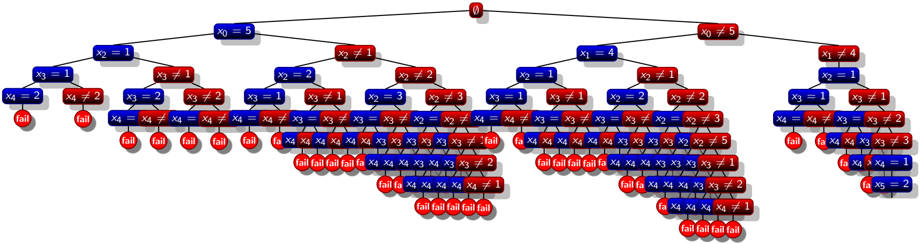
Coloring



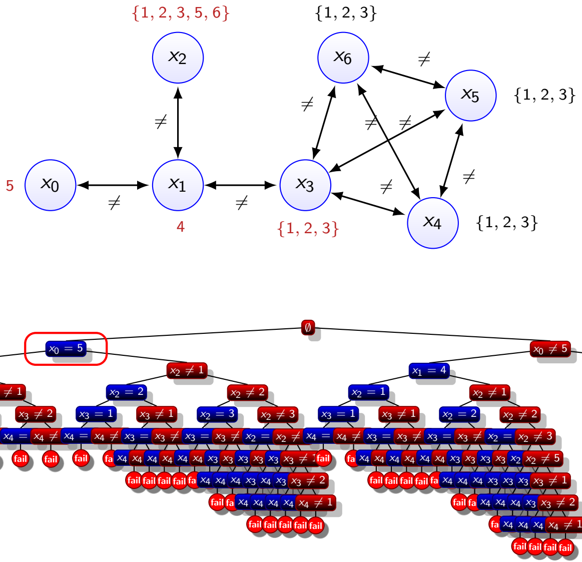
$\{1,2,3,4,5,6\}$
x_2
$\{1,2,3\}$
x_6
$\neq$
x_5
$\{1,2,3\}$
$\neq$
$\neq$
$\neq$
$\neq$
$\{5,6\}$
x_0
x_1
x_3
$\neq$
$\neq$
$\neq$
$\neq$
x_4
$\{1,2,3\}$
$\{4,5\}$
$\{1,2,3,4\}$

Lexicographic

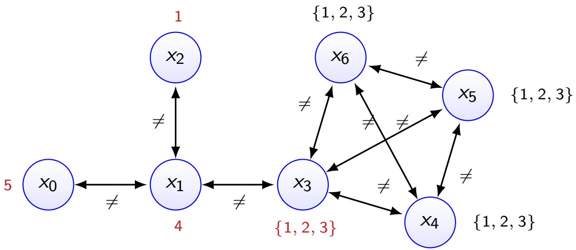


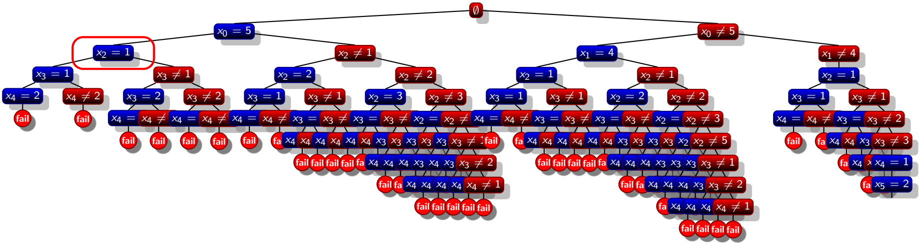


Lexicographic

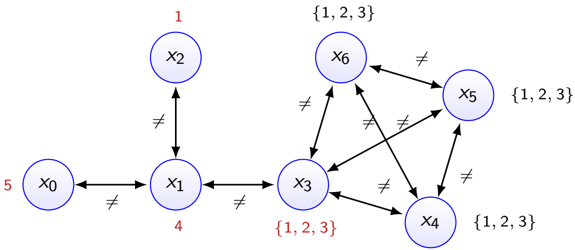


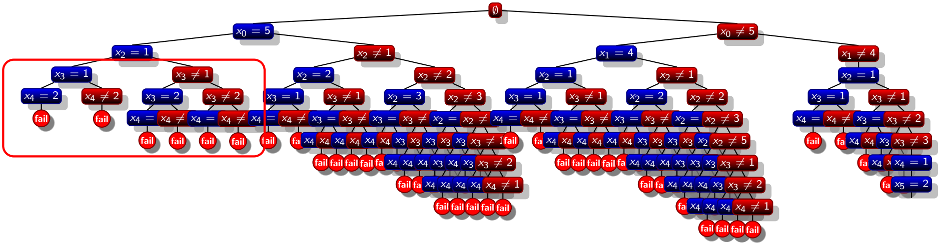
Lexicographic



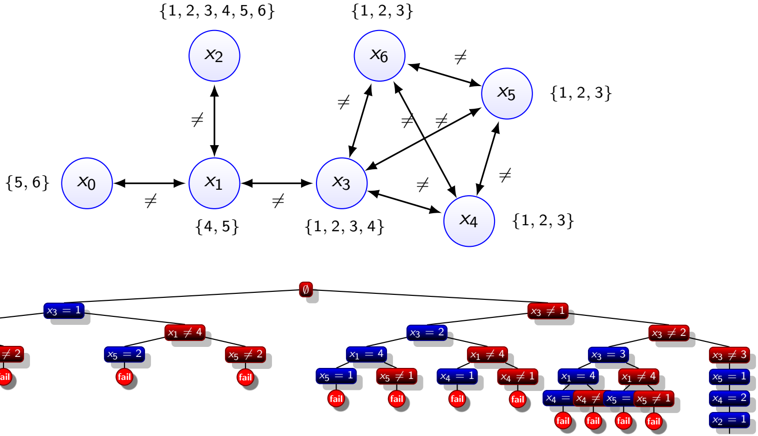


Lexicographic

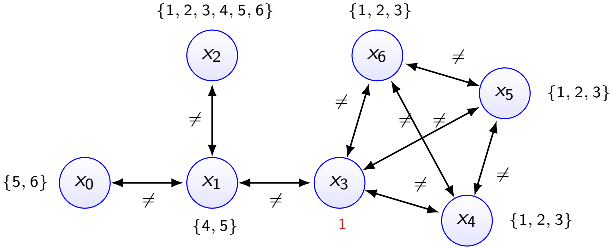


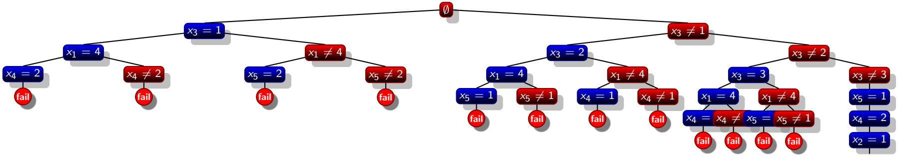


Maximum Degree

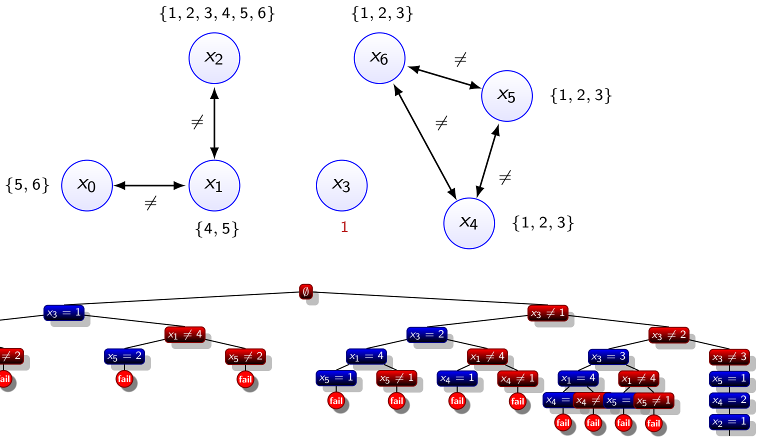


Maximum Degree

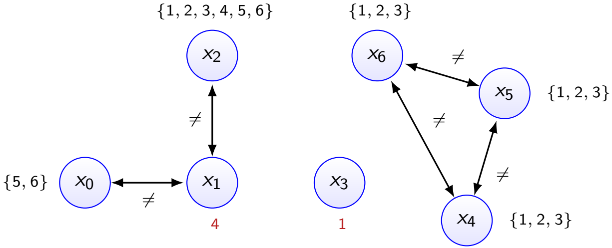


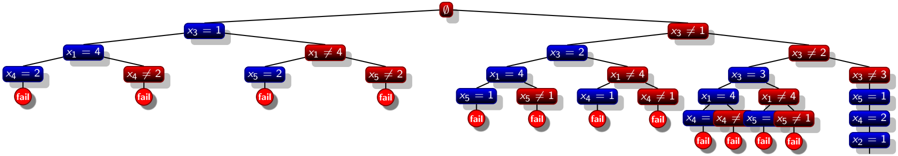


Maximum Degree

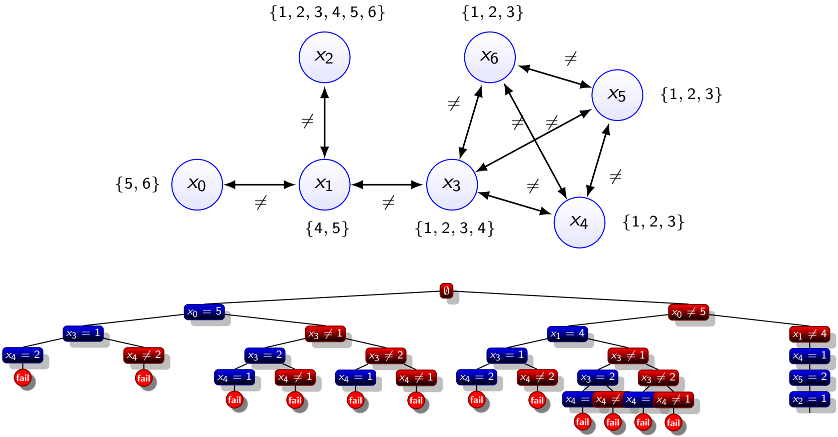


Maximum Degree

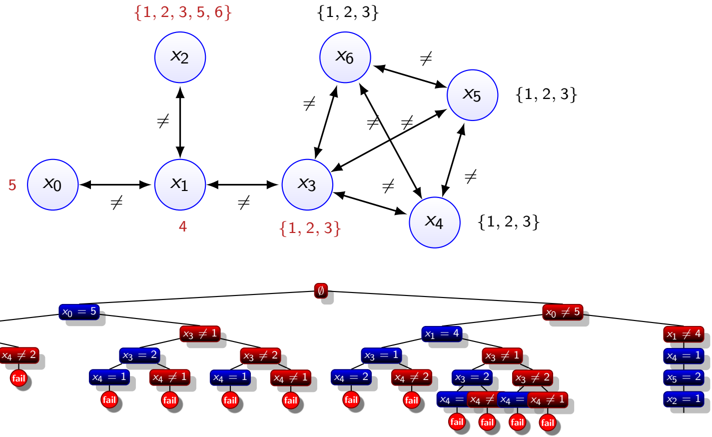




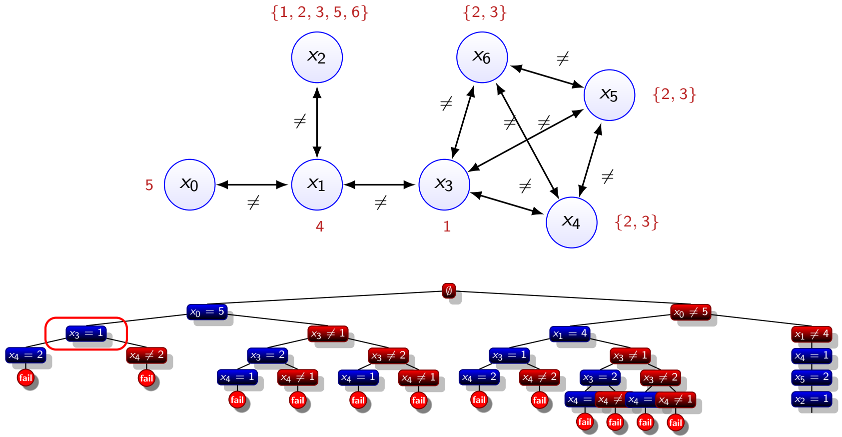
Minimum Domain



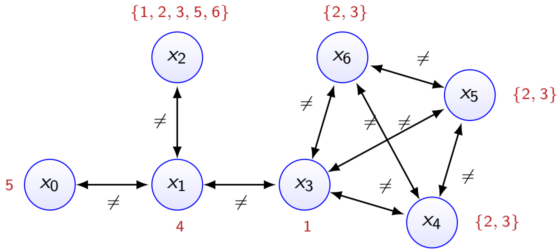
Minimum Domain

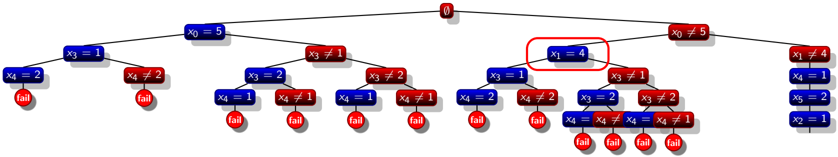


Minimum Domain

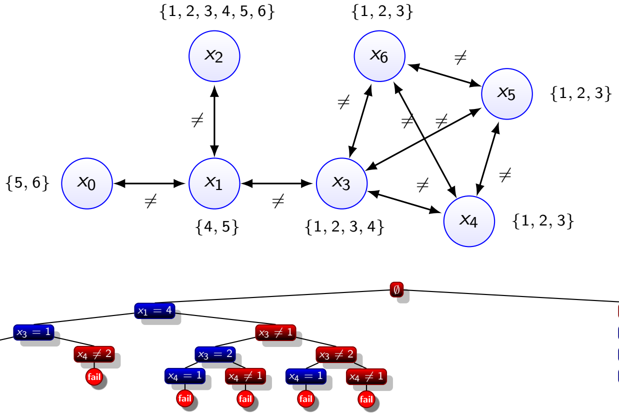


Minimum Domain

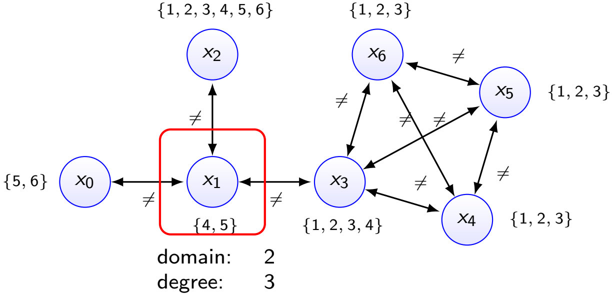


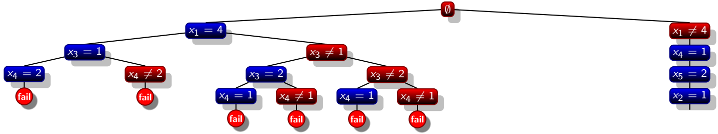


Minimum Domain / Degree

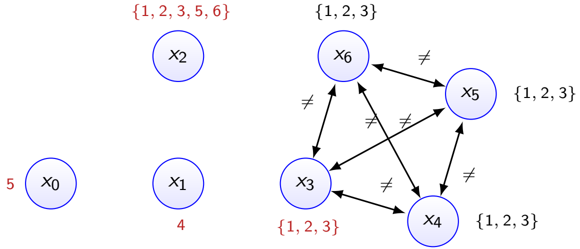


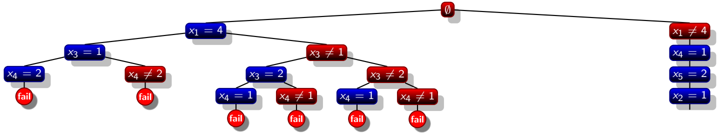
Minimum Domain / Degree



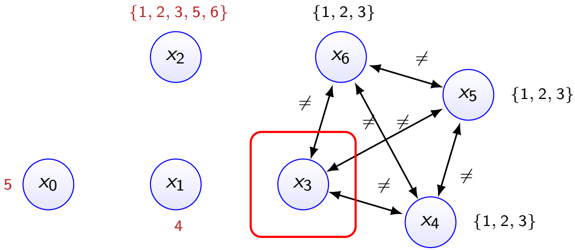


Minimum Domain / Degree

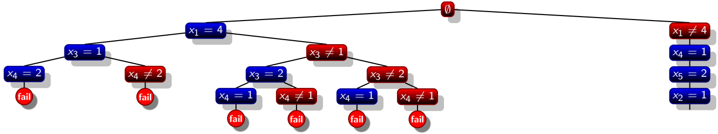




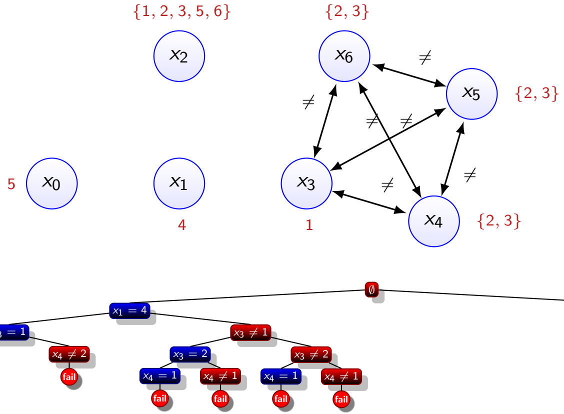
Minimum Domain / Degree



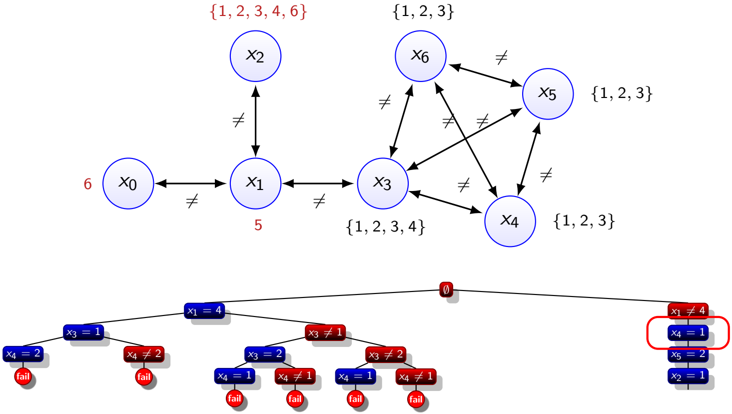
domain: 3
degree: 3



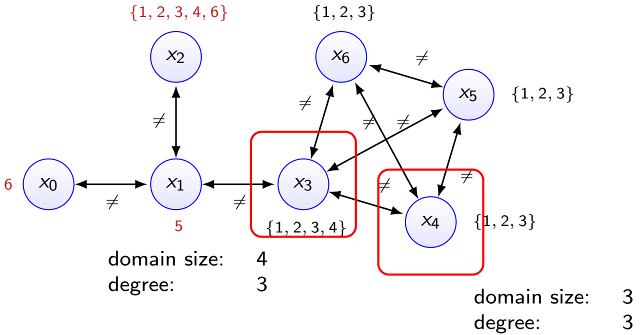
Minimum Domain / Degree

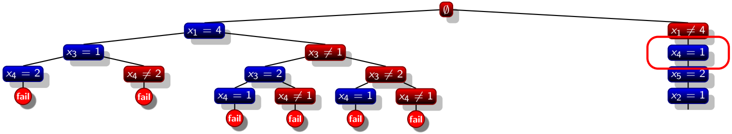


Minimum Domain / Degree



Minimum Domain / Degree





Weighted Degree Heuristic

- The weight of a constraint is initialised to 1
- When propagating, if a contradiction is detected on that constraint, its weight is increased by 1
- The weight of a variable is equal to the sum of the weights of its neighboring constraints

$$\mathrm{weight}(x) = \sum_{c \text{ s.t. } x \in \mathcal{X}(c)} \mathrm{weight}(c)$$

- Domain over weigthed degree:
  - The variable that minimizes the ratio $\frac{\text{domain size}}{\textit{weight}}$ is selected first

Impact

- The impact $I(x, v)$ of a pair (x, v) is initialised to 0
- The size of the search space of a network $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ is bounded by $\prod_{x \in \mathcal{X}} |\mathcal{D}(x)|$
- When the decision $x = v$ is succesful
  - Let b be the size of the search space of $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ (before the decision)
  - Let a be the size of the search space of $\mathcal{P}' = \mathrm{AC}(\mathcal{X}, \mathcal{D}, \mathcal{C} \cup (x = v))$ (after the decision)
  - The impact of this decision corresponds to the reduction of the search space: $\frac{b-a}{b}$
  - The impact of (x, v) is the average impact computed for all decisions $x = v$.
- The weight of a variable x is $\sum_{v \in \mathcal{D}(x)} \frac{1}{I(x,v)}$, the variable with minimum weight is selected first

Value Ordering

- Often considered less important than variable ordering
  - Most of the search effort is spent in unsatisfiable subtrees
  - Value ordering does not really matter in an unsatisfiable subtree since it must be fully explored
- In satisfaction problems:
  - Choosing a satisfiable subtree when branching (promise: choose the least constrained value)
- In optimisation problems:
  - Choosing the assignment that is likely to give the best objective value in order to get good upper bounds quickly

Heavy Tail Behavior

- Given a collection of instances of a problem, we often observe a few exceptionally hard instances
  - These instances are rare, but take exceptionally longer to solve
  - Large impact on the mean runtime for a given set
  - As opposed to normal distributions, the mean does not stabilize when the size of the sample grows
    - ★ When the sample grows, the mean runtime is skewed up
    - ★ Heavy tail behavior
- Not a characteristic of the instance, the same behavior behavior is observed if we run several times the same instance while varying some parameter of the solver
  - For some combination instance / solver parameters, we get trapped into an exponential subtree

Heavy Tail Behavior

- Randomization:
  - Add some randomized parameter in variable or value selection (for instance to break ties)
  - Given the same random seed the solver will explore the same tree, however it will never explore two identical subproblems in the same way
- Restarting:
  - After a given limit r, for instance in number of explored nodes: restart from scratch
- Randomization + restarts eliminates the huge variance in solver performance
  - And therefore reduces the mean runtime when a heavy tail behavior could be observed

- Which limit r should we use?
  - Geometric: $r_i = f^i b$
    - ★ $b = 100, f = 2$: 100, 200, 400, 800, ...
  - Luby:
    - ★ $r_i = 2^{i-1} b$ if $i = 2^k - 1$
    - ★ $r_{i-2^{k-1}+1} b$ if $2^{k-1} \leq i \leq 2^k - 1$
    - ★ $b = 10$: 10,10,20,10,10,20,40,10,10,20,10,10,20,40,80 ...

Outline

1. Backtrack Search
2. **Discrepancy Search techniques**

Limits of heuristics

- Idea of Limited Discrepancy Search:
  - A heuristic is never perfect
  - But if we trust it, we want to search first as close as possible to the advised decisions.
- Idea of Depth Bounded Discrepency Search:
  - Early choices are less informed
  - A heuristic is generally less reliable at the top of the tree

Limited Discrepancy Search