DM841

Discrete Optimization

Part I

Lecture 2

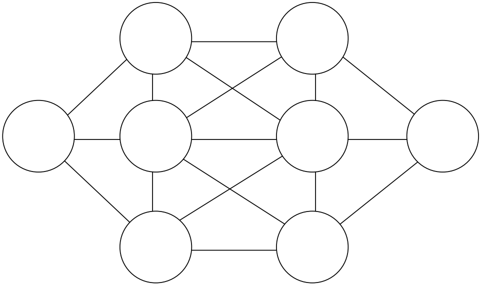
Solving Constraint Satisfaction Problems

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline



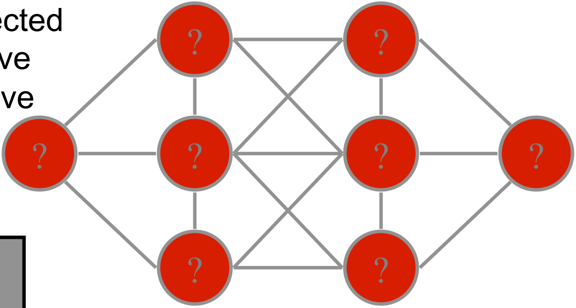
Put a different number in each circle (1 to 8) such that adjacent circles cannot take consecutive numbers

Constraint Programming
An Introduction
by example

Patrick Prosser
with the help of Toby Walsh, Chris Beck,
Barbara Smith, Peter van Beek, Edward Tsang, ...

A Puzzle

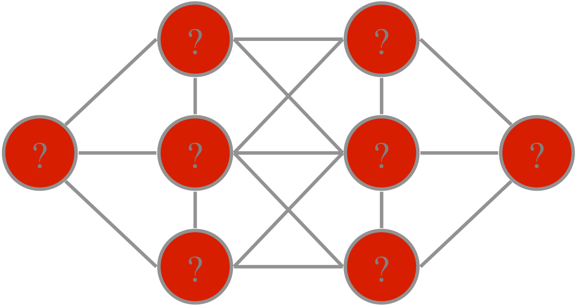
- Place numbers 1 through 8 on nodes
  - Each number appears exactly once
  - No connected nodes have consecutive numbers



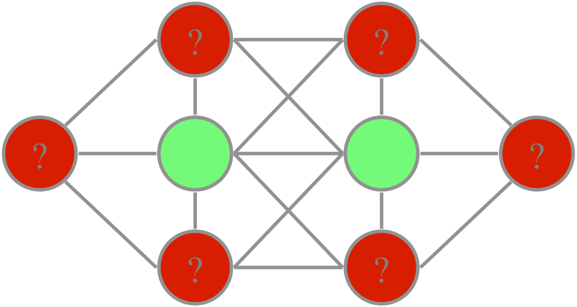
You have
8 minutes!

Heuristic Search

Which nodes are hardest to number?

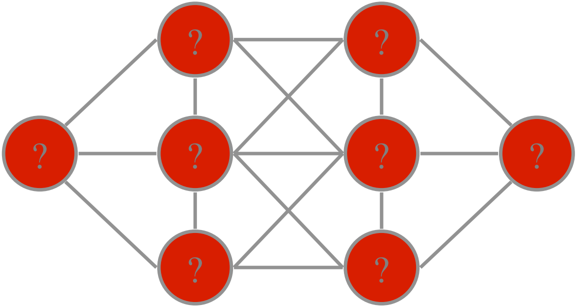


Heuristic Search



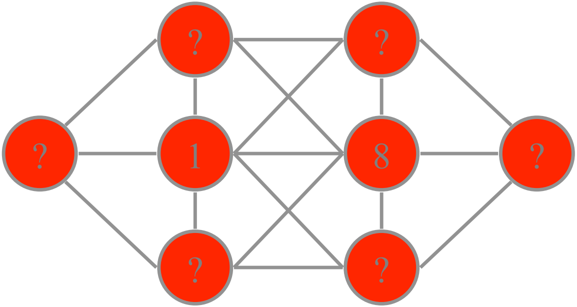
Heuristic Search

Which are the least constraining values to use?



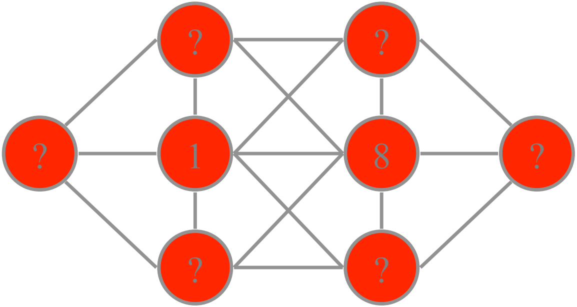
Heuristic Search

Values 1 and 8



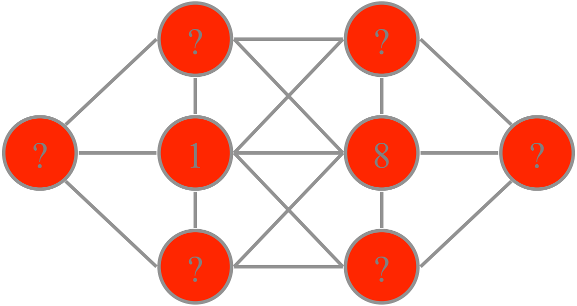
Heuristic Search

Values 1 and 8



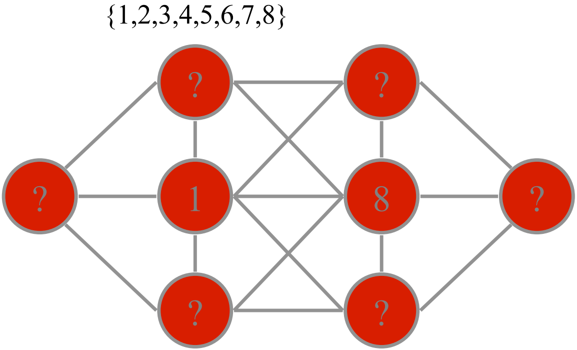
Symmetry means we don’t need to consider: 8 1

Inference/propagation

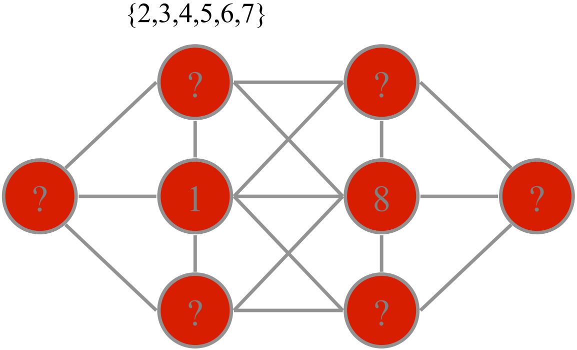


We can now eliminate many values for other nodes

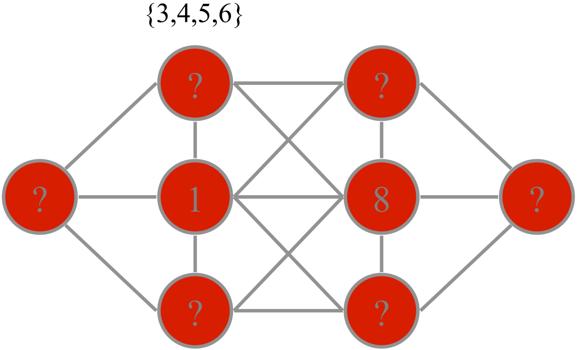
Inference/propagation



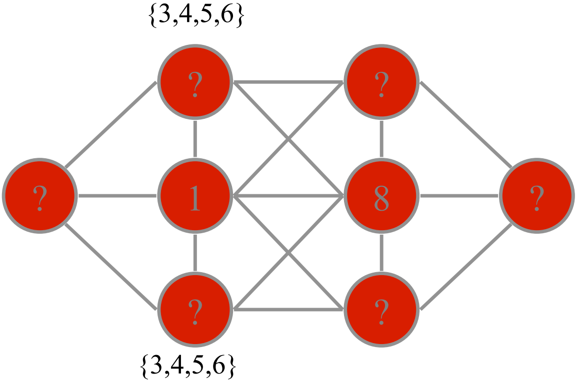
Inference/propagation



Inference/propagation

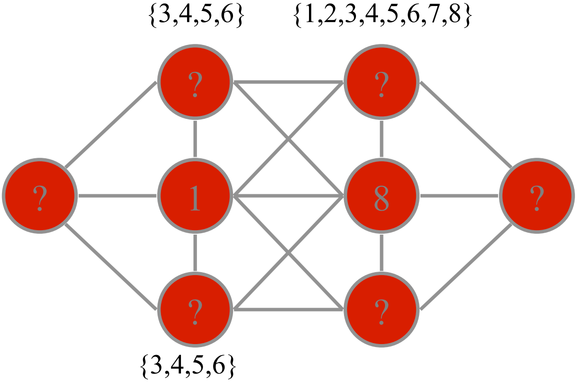


Inference/propagation

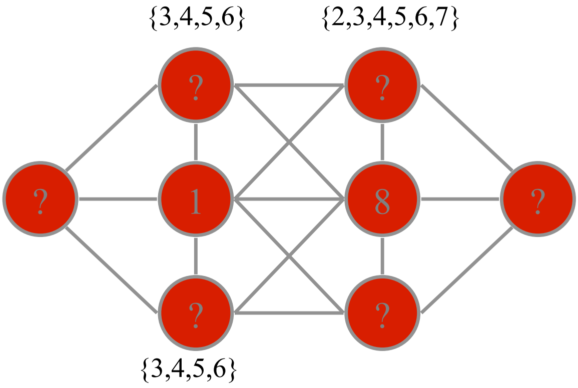


By symmetry

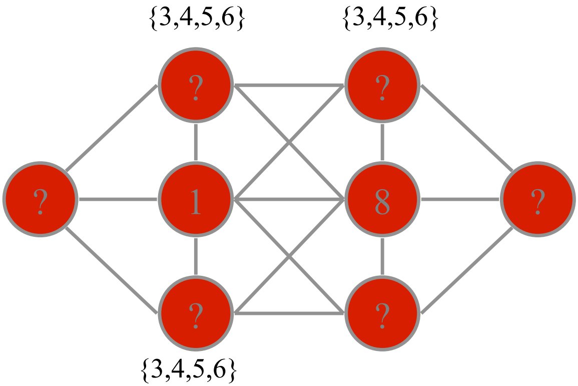
Inference/propagation



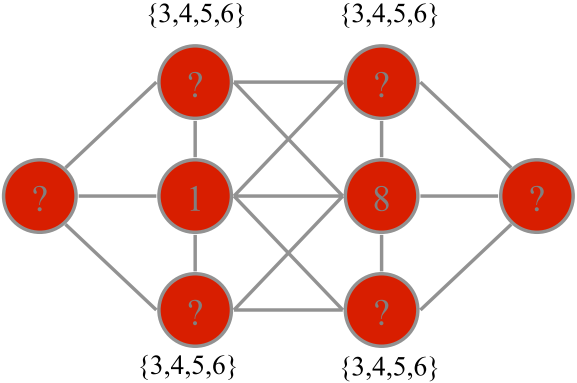
Inference/propagation



Inference/propagation

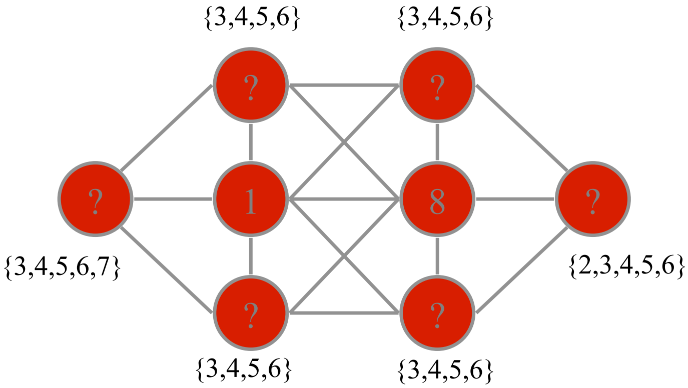


Inference/propagation

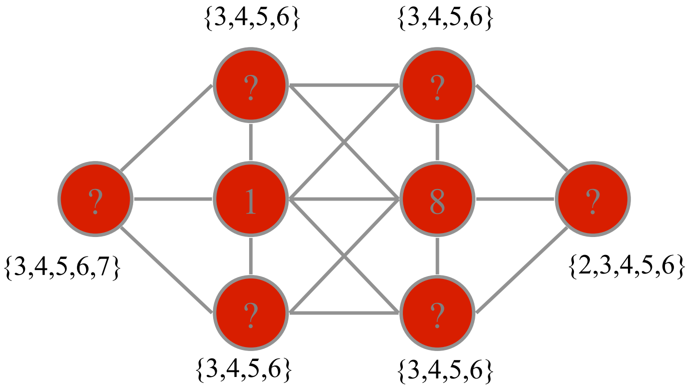


By symmetry

Inference/propagation

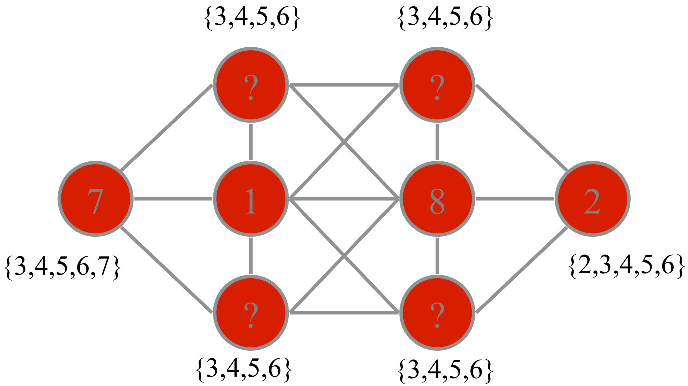


Inference/propagation



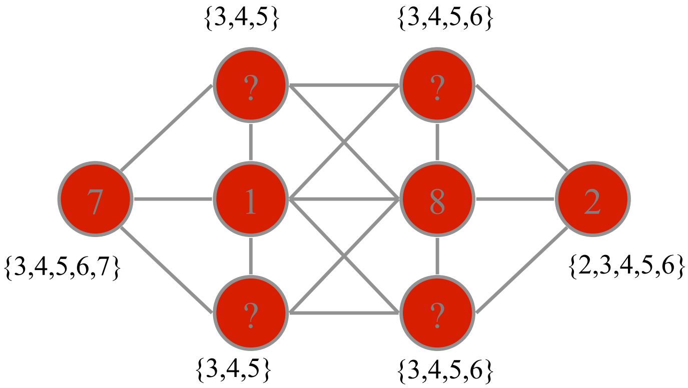
Value 2 and 7 are left in just one variable domain each

Inference/propagation



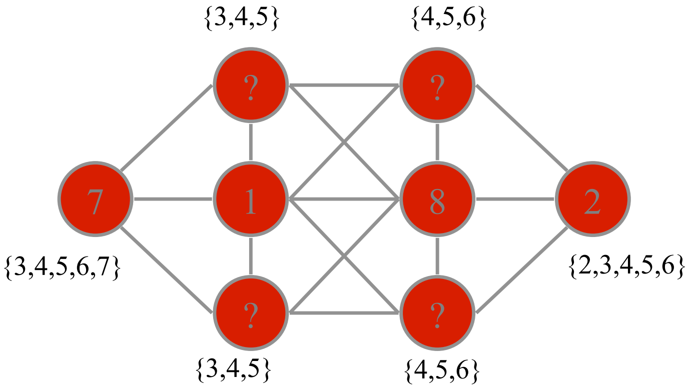
And propagate …

Inference/propagation



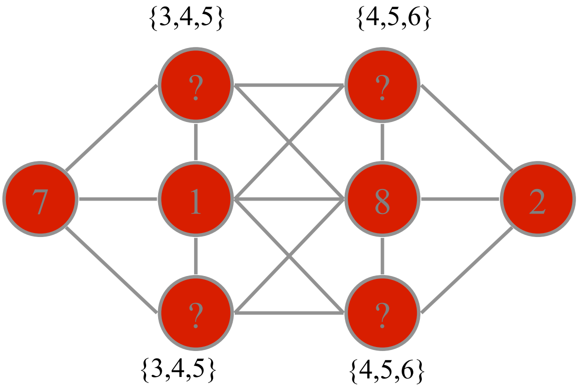
And propagate …

Inference/propagation



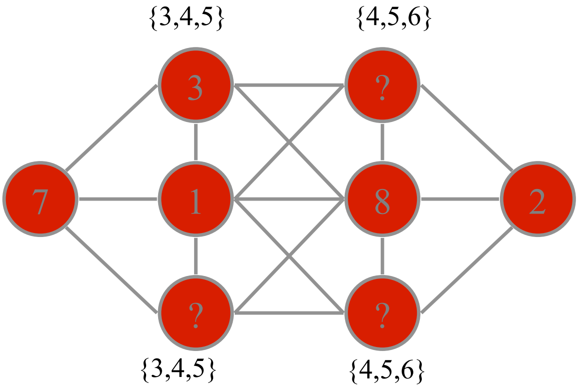
And propagate …

Inference/propagation



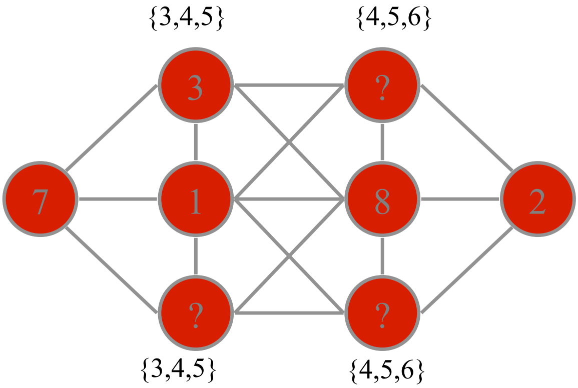
Guess a value, but be prepared to backtrack …

Inference/propagation



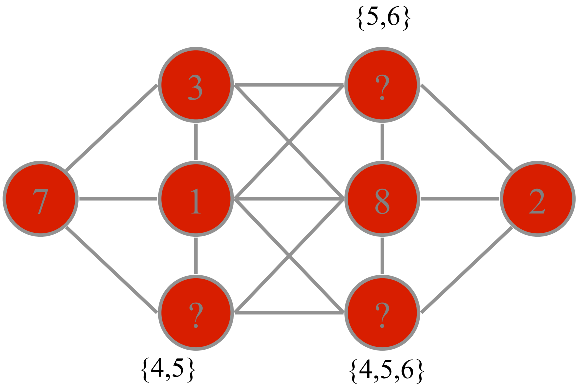
Guess a value, but be prepared to backtrack …

Inference/propagation



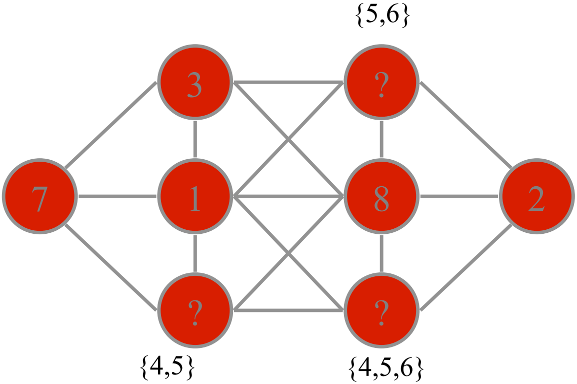
And propagate …

Inference/propagation



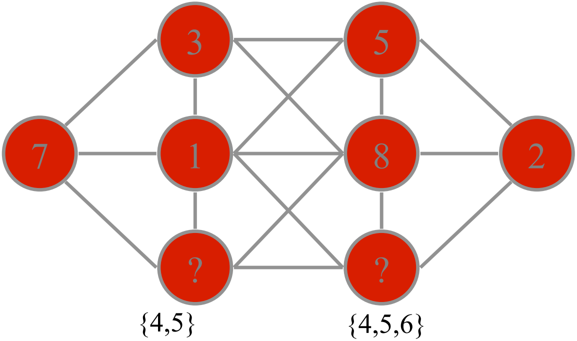
And propagate …

Inference/propagation



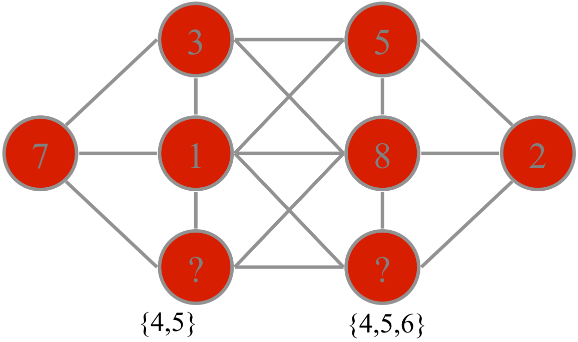
Guess another value …

Inference/propagation



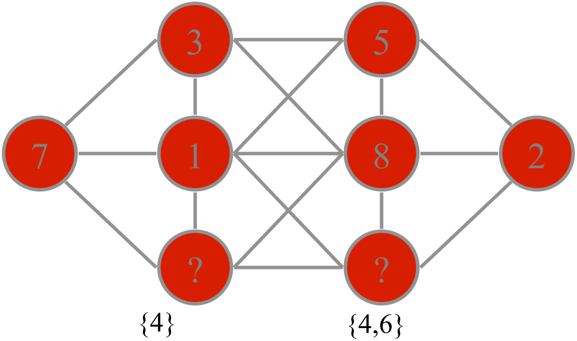
Guess another value …

Inference/propagation



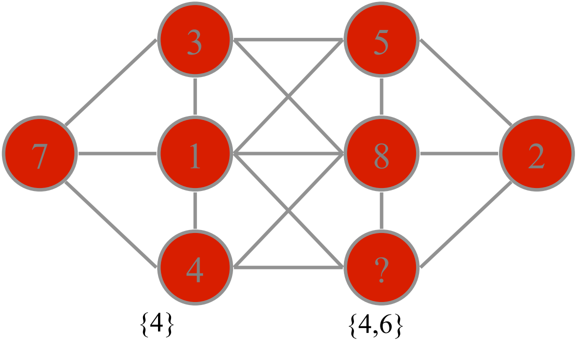
And propagate …

Inference/propagation



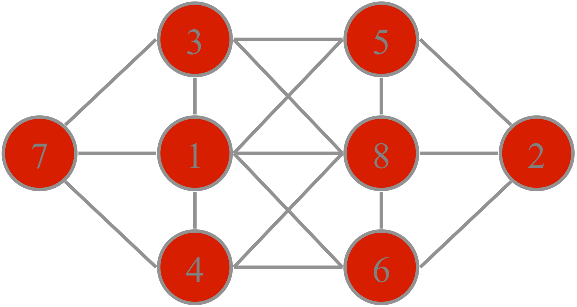
And propagate …

Inference/propagation



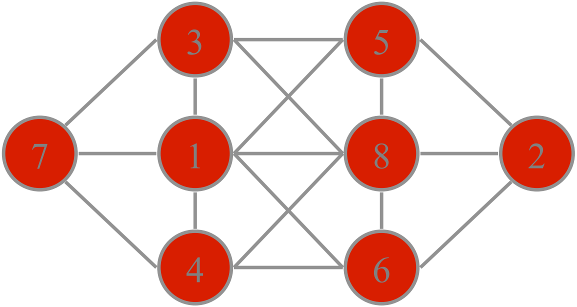
One node has only a single value left …

Inference/propagation



{6}

Solution

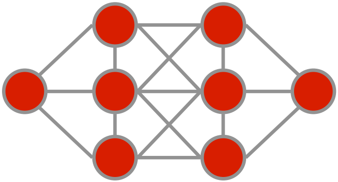


The Core of Constraint Computation

- Modelling
  - Deciding on variables/domains/constraints
- Heuristic Search
- Inference/Propagation
- Symmetry
- Backtracking

Hardness

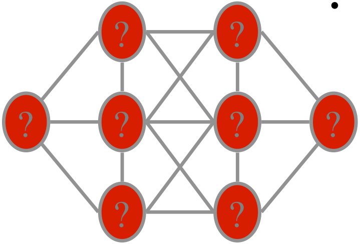
- The puzzle is actually a hard problem
  - NP-complete



Constraint programming

- Model problem by specifying constraints on acceptable solutions
  - define variables and domains
  - post constraints on these variables
- Solve model
  - choose algorithm
    - incremental assignment / backtracking search
    - complete assignments / stochastic search
  - design heuristics

Example CSP



- Variable, v_i for each node
- Domain of {1, …, 8}
- Constraints
  - All values used

  allDifferent(v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8)

  - No consecutive numbers for adjoining nodes

  $|v_1 - v_2| > 1$

  $|v_1 - v_3| > 1$

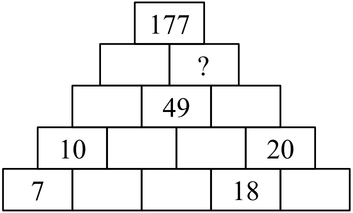
  …

more examples?

Do you know any constraint satisfaction problems?

To a man with a hammer, everything looks like a nail.

Scotsman 4/12/2003



In the pyramid above, two adjacent bricks added together give the value of the brick above. Find the value for the brick marked ?

Constraint Programming

Constraint Programming: an alternative approach to imperative programming and object oriented programming.

- Variables each with a finite set of possible values (domain)
- Constraint on a sequence of variables: a relationship on their domains

Constraint Satisfaction Problem: finite set of constraints

CP

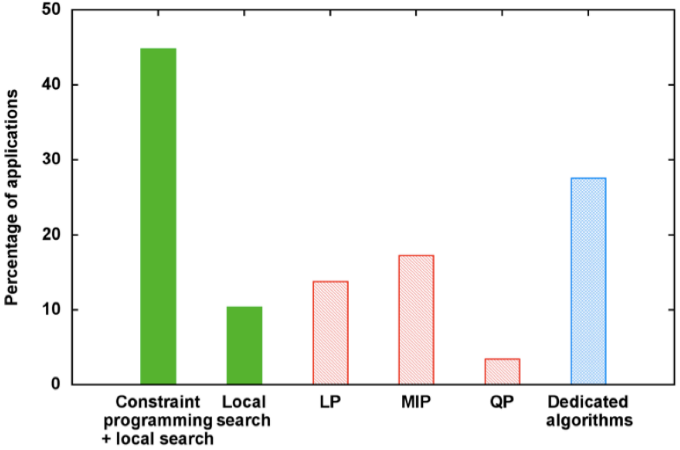
Constraint Programming	=	model (representation) +
		propagation (reasoning, inference) +
		search (reasoning, inference)

Applications

- Operation research (optimization problems)
- Graphical interactive systems (to express geometrical correctness)
- Molecular biology (DNA sequencing, 3D models of proteins)
- Finance
- Circuit verification
- Elaboration of natural languages (construction of efficient parsers)
- Scheduling of activities
- Configuration problem in form compilation
- Generation of coerent music programs [Anders and Miranda [2011]].
- Data bases
- ...
- http://hsimonis.wordpress.com/

Applications

Distribution of technology used at Google for optimization applications developed by the operations research team



[Slide presented by Laurent Perron on OR-Tools at CP2013]

List of Contents

Outline

Constraint Programming

The **domain** of a variable x, denoted $D(x)$, is a finite set of elements that can be assigned to x.

A **constraint** C on X is a subset of the Cartesian product of the domains of the variables in X, i.e., $C \subseteq D(x_1) \times \cdots \times D(x_k)$. A tuple $(d_1, \ldots, d_k) \in C$ is called a solution to C.
Equivalently, we say that a solution $(d_1, ..., d_k) \in C$ is an assignment of the value d_i to the variable x_i for all $1 \leq i \leq k$, and that this assignment satisfies C. If $C = \emptyset$, we say that it is inconsistent.

Extensional: specifies the good (or bad) tuples (values)
Intensional: specifies the characteristic function

Constraint Programming

Constraint Satisfaction Problem (CSP)

A CSP is a finite set of variables $\mathcal{X}$ with domain extension $\mathcal{D} = D(x_1) \times \cdots \times D(x_n)$, together with a finite set of constraints $\mathcal{C}$, each on a subset of $\mathcal{X}$. A **solution** to a CSP is an assignment of a value $d \in D(x)$ to each $x \in \mathcal{X}$, such that all constraints are satisfied simultaneously.

Constraint Optimization Problem (COP)

A COP is a CSP $\mathcal{P}$ defined on the variables $x_1, \ldots, x_n$, together with an objective function $f : D(x_1) \times \cdots \times D(x_n) \to Q$ that assigns a value to each assignment of values to the variables. An **optimal solution** to a minimization (maximization) COP is a solution d to $\mathcal{P}$ that minimizes (maximizes) the value of $f(d)$.

Task:

- determine whether the CSP/COP is consistent (has a solution):
- find one solution
- find all solutions
- find one optimal solution
- find all optimal solutions

Solving CSPs

- Systematic search:
  - choose a variable x_i that is not yet assigned
  - create a choice point, i.e. a set of mutually exclusive & exhaustive choices, e.g. $x_i = v$ vs $x_i \neq v$
  - try the first & backtrack to try the other if this fails
- Constraint propagation:
  - add $x_i = v$ or $x \neq v$ to the set of constraints
  - re-establish local consistency on each constraint
    $\rightsquigarrow$ remove values from the domains of future variables that can no longer be used because of this choice
  - fail if any future variable has no values left

Representing a Problem

- If a CSP $\mathcal{P} =< \mathcal{X}, \mathcal{DE}, \mathcal{C} >$ represents a problem P, then every solution of $\mathcal{P}$ corresponds to a solution of P and every solution of P can be derived from at least one solution of $\mathcal{P}$
- More than one solution of P can be represented by the same solution of $\mathcal{P}$, if modelling removes symmetry
- The variables and values of $\mathcal{P}$ represent entities in P
- The constraints of $\mathcal{P}$ ensure the correspondence between solutions
- The aim is to find a model $\mathcal{P}$ that can be solved as quickly as possible (Note that shortest run-time might not mean least search!)

Interactions with Search Strategy

Whether a model is better than another can depend on the search algorithm and search heuristics

- Let's assume that the search algorithm is fixed
  although different level of consistency can also play a role
- Let's also assume that choice points are always $x_i = v$ vs $x_i \neq v$
- Variable (and value) order still interact with the model a lot
- Is variable & value ordering part of modelling?

In practice it is.
but it depends on the modeling language used

Global Constraint: alldifferent

Global constraint:

set of more elementary constraints that exhibit a special structure when considered together.

alldifferent constraint

Let $x_1, x_2, \ldots, x_n$ be variables. Then:

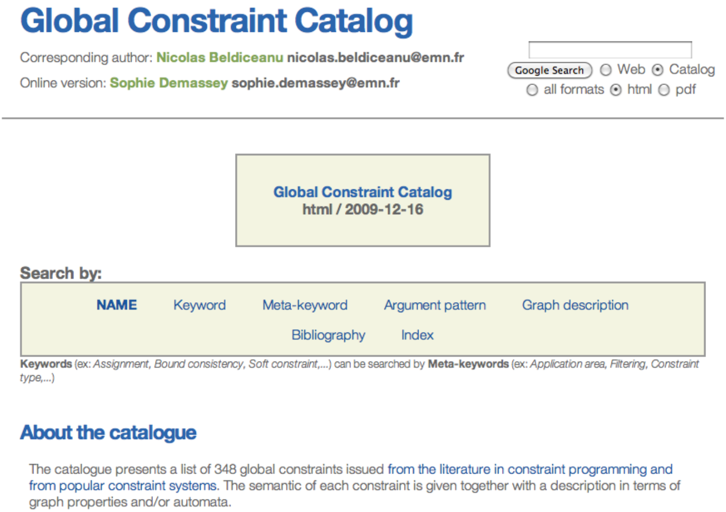
$$\texttt{alldifferent}(x_1, ..., x_n) = \{(d_1, ..., d_n) \mid \forall i, d_i \in D(x_i), \quad \forall i \neq j,\ d_i \neq d_j\}.$$

Constraint arity: number of variables involved in the constraint

Note: different notation and names used in the literature

Global Constraint Catalog

http://www.emn.fr/z-info/sdemasse/gccat/sec5.html



Global Constraint Catalog

Corresponding author: Nicolas Beldiceanu nicolas.beldiceanu@emn.fr

Online version: Sophie Demassey sophie.demassey@emn.fr

Google Search ○ Web ⦿ Catalog
○ all formats ⦿ html ○ pdf

Global Constraint Catalog
html / 2009-12-16

Search by:

NAME Keyword Meta-keyword Argument pattern Graph description
Bibliography Index

Keywords (ex: *Assignment, Bound consistency, Soft constraint,...*) can be searched by **Meta-keywords** (ex: *Application area, Filtering, Constraint type,...*)

About the catalogue

The catalogue presents a list of 348 global constraints issued from the literature in constraint programming and from popular constraint systems. The semantic of each constraint is given together with a description in terms of graph properties and/or automata.

Outline

Outline

Computational Models

Three main Computational Models to solve (combinatorial) constrained optimization problems:

- Mathematical Programming (LP, ILP, QP, SDP, ...)
- Constraint Programming (CSP as a model, SAT as a very special case)
- Local Search (... and Meta-heuristics)
- Others? Dynamic programming, dedicated algorithms, satisfiability modulo theory, answer set programming, etc.

Modeling

Modeling:

1. identify:
   - parameters
   - variables and domains
   - constraints
   - objective functions

   that formulate the problem

2. express what in point 1) in a way that allows the solution by available software

Variables

In MILP: real and integer (mostly binary) variables

In CP:

- finite domain integer (including Booleans),
- continuos with interval constraints
- structured domains: finite sets, multisets, graphs, ...

In LS: integer variables

Constraint Programming vs MILP

- In MILP we formulate problems as a set of linear inequalities
- In CP we describe substructures (so-called global constraints) and combine them with various combinators.
- Substructures capture building blocks often (but not always) comptuationally tractable by special-purpose algorithms
- CP models can:
  - be solved by the constraint engine
  - be linearized and solved by their MIP solvers;
  - be translated in CNF and sovled by SAT solvers;
  - be handled by local search
- In MILP the solver is often seen as a black-box
  In CP and LS solvers leave the user the task of programming the search.
- CP = model + propagation + search
  constraint propagation by domain filtering $\rightsquigarrow$ inference
  search = backtracking or branch and bound or local search

Outline

Example: Send More Money

Send + More = Money

You are asked to replace each letter by a different digit so that

	S	E	N	D	+
	M	O	R	E	=
M	O	N	E	Y	

is correct. Because S and M are the leading digits, they cannot be equal to the 0 digit.

Can you model this problem in MILP/CP?

Send More Money: CP model

SEND + MORE = MONEY

- $X_i \in \{0, \ldots, 9\}$ for all $i \in I = \{S, E, N, D, M, O, R, Y\}$

- Crypto constraint $\rightsquigarrow$ 1 equality constraint:

$$\begin{array}{cccccc} & 10^3 X_1 & +10^2 X_2 & +10 X_3 & +X_4 & + \\ & 10^3 X_5 & +10^2 X_6 & +10 X_7 & +X_2 & = \\ \hline 10^4 X_5 & +10^3 X_6 & +10^2 X_3 & +10 X_2 & +X_8 & \end{array}$$

- Each letter takes a different digit $\rightsquigarrow$ 1 inequality constraint

$$\texttt{alldifferent}([X_1, X_2, \ldots, X_8]).$$

(it substitutes 28 inequality constraints: $X_i \neq X_j, i, j \in I,\ i \neq j$)

- This is one model, not the model of the problem
- Many possible alternatives
- Choice often depends on the constraint system available
  Constraints available
  Reasoning attached to constraints
- Not always clear which is the best model

Send More Money: CP model (revisited)

- $X_i \in \{0, \ldots, 9\}$ for all $i \in I = \{S, E, N, D, M, O, R, Y\}$

- $$\begin{array}{cccccc} & 10^3 X_1 & +10^2 X_2 & +10 X_3 & +X_4 & + \\ & 10^3 X_5 & +10^2 X_6 & +10 X_7 & +X_2 & = \\ \hline 10^4 X_5 & +10^3 X_6 & +10^2 X_3 & +10 X_2 & +X_8 & \end{array}$$

- $$\text{alldifferent}([X_1, X_2, \ldots, X_8]).$$

- Redundant constraints (5 equality constraints)

$$\begin{aligned} X_4 + X_2 &= 10\, r_1 + X_8, \\ X_3 + X_7 + r_1 &= 10\, r_2 + X_2, \\ X_2 + X_6 + r_2 &= 10\, r_3 + X_3, \\ X_1 + X_5 + r_3 &= 10\, r_4 + X_6, \\ +r_4 &= X_5. \end{aligned}$$

Can we do better? Can we propagate something?

Send More Money: CP model
Gecode-python


```
from gecode import *

s = space()
letters = s.intvars(8,0,9)
S,E,N,D,M,O,R,Y = letters
s.rel(M,IRT_NQ,0)
s.rel(S,IRT_NQ,0)
s.distinct(letters)
C = [1000, 100, 10, 1,
     1000, 100, 10, 1,
     -10000, -1000, -100, -10, -1]
X = [S,E,N,D,
     M,O,R,E,
     M,O,N,E,Y]
s.linear(C,X, IRT_EQ, 0)
s.branch(letters, INT_VAR_SIZE_MIN, INT_VAL_MIN)
for s2 in s.search():
    print(s2.val(letters))
```

Send Most Money: CP model

Gecode-python

Optimization version:

$$\max \sum_{i \in I'} C_i X_i, \; I' = \{M, O, N, E, Y\}$$

```
from gecode import *

s = space()
letters = s.intvars(8,0,9)
S,E,N,D,M,O,T,Y = letters
s.rel(M,IRT_NQ,0)
s.rel(S,IRT_NQ,0)
s.distinct(letters)
C = [1000, 100, 10, 1,
     1000, 100, 10, 1,
     -10000, -1000, -100, -10, -1]
X = [S,E,N,D,
     M,O,S,T,
     M,O,N,E,Y]
s.linear(C,X,IRT_EQ,0)
money = s.intvar(0,99999)
s.linear([10000,1000,100,10,1],[M,O,N,E,Y], IRT_EQ, money)
s.maximize(money)
s.branch(letters, INT_VAR_SIZE_MIN, INT_VAL_MIN)
for s2 in s.search():
    print(s2.val(money), s2.val(letters))
```

Send More Money: CP model
MiniZinc

SEND-MORE-MONEY ≡ [DOWNLOAD]

```
include "alldifferent.mzn";

var 1..9: S;
var 0..9: E;
var 0..9: N;
var 0..9: D;
var 1..9: M;
var 0..9: O;
var 0..9: R;
var 0..9: Y;

constraint           1000 * S + 100 * E + 10 * N + D
                   + 1000 * M + 100 * O + 10 * R + E
       = 10000 * M + 1000 * O + 100 * N + 10 * E + Y;

constraint alldifferent([S,E,N,D,M,O,R,Y]);

solve satisfy;

output ["   ",show(S),show(E),show(N),show(D),"\n",
        "+  ",show(M),show(O),show(R),show(E),"\n",
        "= ",show(M),show(O),show(N),show(E),show(Y),"\n"];
```

H. Simonis' demo, slides 33-134

Domain Visualization

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Domain Visualization

Rows = Variables

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Domain Visualization

Columns = Values

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Domain Visualization

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M			Cells=		State					
O										
R										
Y										

Alldifferent Constraint

```
alldifferent(L),
```

- Built-in of `ic` library
- No initial propagation possible
- *Suspends*, waits until variables are changed
- When variable is fixed, remove value from domain of other variables
- *Forward checking*

Alldifferent Visualization

Uses the same representation as the domain visualizer

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Disequality Constraints

```
S #\= 0, M#\= 0,
```

Remove value from domain

$$S \in \{1..9\}, M \in \{1..9\}$$

Constraints solved, can be removed

Domains after Disequality

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y										

Equality Constraint

- Normalization of linear terms
  - Single occurence of variable
  - Positive coefficients
- Propagation

Normalization

	1000*S+	100*E+	10*N+	D
	+1000*M+	100*O+	10*R+	E
10000*M+	1000*O+	100*N+	10*E+	Y

Normalization

	1000*S+	100*E+	10*N+	D
	+**1000*M**+	100*O+	10*R+	E
10000*M+	1000*O+	100*N+	10*E+	Y

Normalization

	1000*S+	100*E+	10*N+	D
	+	100*O+	10*R+	E
9000*M+	1000*O+	100*N+	10*E+	Y

Normalization

	1000*S+	100*E+	10*N+	D
	+	**100*O**+	10*R+	E
9000*M+	**1000*O**+	100*N+	10*E+	Y

Normalization

	1000*S+	100*E+	10*N+	D
		+	10*R+	E
9000*M+	**900*O**+	100*N+	10*E+	Y

Cork Constraint Computation Centre

Normalization

	1000*S+	100*E+	**10*N+**	D
		+	10*R+	E
9000*M+	900*O+	**100*N+**	10*E+	Y

Normalization

	1000*S+	100*E+		D
		+	10*R+	E
9000*M+	900*O+	**90*N**+	10*E+	Y

Normalization

	1000*S+	**100*E+**		D
		+	10*R+	**E**
9000*M+	900*O+	90*N+	**10*E+**	Y

Normalization

	1000*S+	91*E+		D
		+	10*R	
9000*M+	900*O+	90*N+		Y

Simplified Equation

$$1000 * S + 91 * E + 10 * R + D = 9000 * M + 900 * O + 90 * N + Y$$

Propagation

$$1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9} =$$
$$9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}$$

Propagation

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{1000..9918} =$$

$$\underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..89919}$$

Propagation

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} =$$

$$\underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

Propagation

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} =$$

$$\underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

Deduction:

$$M = 1, S = 9, O \in \{0..1\}$$

Propagation

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} =$$

$$\underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

Deduction:

$$M = 1, S = 9, O \in \{0..1\}$$

Why? Skip

Consider lower bound for S

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Lower bound of equation is 9000
- Rest of lhs (left hand side) $(91 * E^{0..9} + 10 * R^{0..9} + D^{0..9})$ is atmost 918
- S must be greater or equal to $\frac{9000-918}{1000} = 8.082$
  - otherwise lower bound of equation not reached by lhs
- S is integer, therefore $S \geq \lceil \frac{9000-918}{1000} \rceil = 9$
- S has upper bound of 9, so $S = 9$

Consider upper bound of M

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Upper bound of equation is 9918
- Rest of rhs (right hand side) $900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}$ is at least 0
- M must be smaller or equal to $\frac{9918-0}{9000} = 1.102$
- M must be integer, therefore $M \leq \lfloor \frac{9918-0}{9000} \rfloor = 1$
- M has lower bound of 1, so $M = 1$

Consider upper bound of O

$$\underbrace{1000 * S^{1..9} + 91 * E^{0..9} + 10 * R^{0..9} + D^{0..9}}_{9000..9918} = \underbrace{9000 * M^{1..9} + 900 * O^{0..9} + 90 * N^{0..9} + Y^{0..9}}_{9000..9918}$$

- Upper bound of equation is 9918
- Rest of rhs (right hand side) $9000 * 1 + 90 * N^{0..9} + Y^{0..9}$ is at least 9000
- O must be smaller or equal to $\frac{9918-9000}{900} = 1.02$
- O must be integer, therefore $O \leq \lfloor \frac{9918-9000}{900} \rfloor = 1$
- O has lower bound of 0, so $O \in \{0..1\}$

Propagation of equality: Result

	0	1	2	3	4	5	6	7	8	9
S		-	-	-	-	-	-	-	-	✹
E										
N										
D										
M		✹	-	-	-	-	-	-	-	-
O			✖	✖	✖	✖	✖	✖	✖	✖
R										
Y										

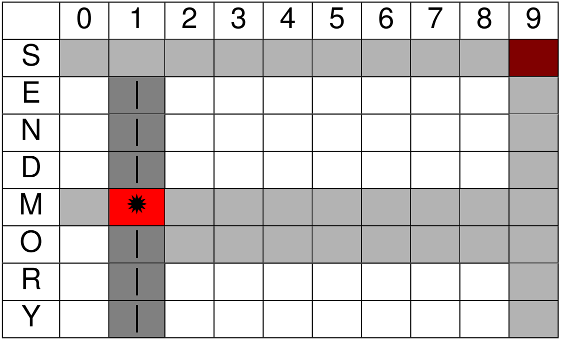
Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S		-	-	-	-	-	-	-	-	✹
E										
N										
D										
M		✹	-	-	-	-	-	-	-	-
O			✖	✖	✖	✖	✖	✖	✖	✖
R										
Y										

Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S										✸
E										|
N										|
D										|
M		✸								
O										
R										|
Y										|

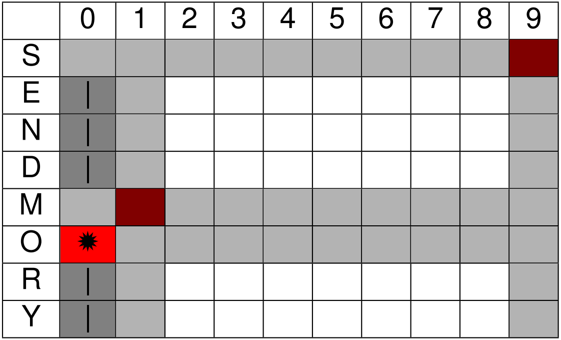
Propagation of alldifferent



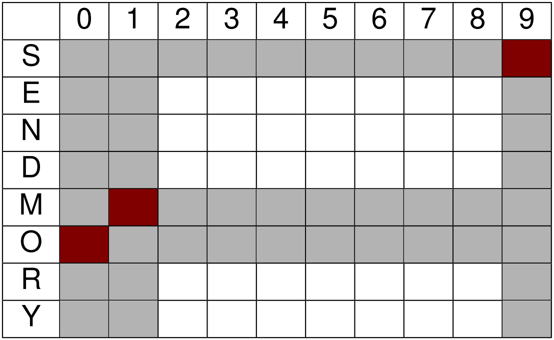
Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O	✸									
R										
Y										

Propagation of alldifferent



Propagation of alldifferent



$$O = 0, [E, R, D, N, Y] \in \{2..8\}$$

Waking the equality constraint

- Triggered by assignment of variables
- *or* update of lower or upper bound

Removal of constants

$$1000 * 9 + 91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} =$$
$$9000 * 1 + 900 * 0 + 90 * N^{2..8} + Y^{2..8}$$

Removal of constants

$$\mathbf{1000} * \mathbf{9} + 91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} =$$
$$\mathbf{9000} * \mathbf{1} + \mathbf{900} * \mathbf{0} + 90 * N^{2..8} + Y^{2..8}$$

Removal of constants

$$91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 90 * N^{2..8} + Y^{2..8}$$

Propagation of equality (Iteration 1)

$$\underbrace{91 * E^{2..8} + 10 * R^{2..8} + D^{2..8}}_{204..816} = \underbrace{90 * N^{2..8} + Y^{2..8}}_{182..728}$$

Propagation of equality (Iteration 1)

$$\underbrace{91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 90 * N^{2..8} + Y^{2..8}}_{204..728}$$

Cork Constraint Computation Centre

Propagation of equality (Iteration 1)

$$\underbrace{91 * E^{2..8} + 10 * R^{2..8} + D^{2..8} = 90 * N^{2..8} + Y^{2..8}}_{204..728}$$

$$N \geq 3 = \lceil \frac{204 - 8}{90} \rceil, E \leq 7 = \lfloor \frac{728 - 22}{91} \rfloor$$

Propagation of equality (Iteration 2)

$$91 * E^{2..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}$$

Propagation of equality (Iteration 2)

$$\underbrace{91 * E^{2..7} + 10 * R^{2..8} + D^{2..8}}_{204..725} = \underbrace{90 * N^{3..8} + Y^{2..8}}_{272..728}$$

Cork Constraint Computation Centre

Propagation of equality (Iteration 2)

$$\underbrace{91 * E^{2..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{272..725}$$

Propagation of equality (Iteration 2)

$$\underbrace{91 * E^{2..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{272..725}$$

$$E \geq 3 = \lceil \frac{272 - 88}{91} \rceil$$

Propagation of equality (Iteration 3)

$$91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}$$

Propagation of equality (Iteration 3)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8}}_{295..725} = \underbrace{90 * N^{3..8} + Y^{2..8}}_{272..728}$$

Propagation of equality (Iteration 3)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{295..725}$$

Propagation of equality (Iteration 3)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{3..8} + Y^{2..8}}_{295..725}$$

$$N \geq 4 = \lceil \frac{295 - 8}{90} \rceil$$

Propagation of equality (Iteration 4)

$$91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}$$

Propagation of equality (Iteration 4)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8}}_{295..725} = \underbrace{90 * N^{4..8} + Y^{2..8}}_{362..728}$$

Propagation of equality (Iteration 4)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{362..725}$$

Propagation of equality (Iteration 4)

$$\underbrace{91 * E^{3..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{362..725}$$

$$E \geq 4 = \lceil \frac{362 - 88}{91} \rceil$$

Propagation of equality (Iteration 5)

$$91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}$$

Propagation of equality (Iteration 5)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{386..725} = \underbrace{90 * N^{4..8} + Y^{2..8}}_{362..728}$$

Propagation of equality (Iteration 5)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{386..725}$$

Propagation of equality (Iteration 5)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{4..8} + Y^{2..8}}_{386..725}$$

$$N \geq 5 = \lceil \frac{386 - 8}{90} \rceil$$

Propagation of equality (Iteration 6)

$$91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}$$

Propagation of equality (Iteration 6)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8}}_{386..725} = \underbrace{90 * N^{5..8} + Y^{2..8}}_{452..728}$$

Propagation of equality (Iteration 6)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452..725}$$

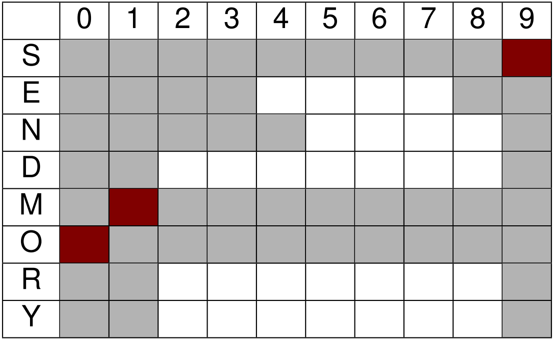
Propagation of equality (Iteration 6)

$$\underbrace{91 * E^{4..7} + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452..725}$$

$$N \geq 5 = \lceil \frac{452 - 8}{90} \rceil, E \geq 4 = \lceil \frac{452 - 88}{91} \rceil$$

No further propagation at this point

Domains after setup



Outline

1. Problem
2. Program
3. Constraint Setup
4. Search
   - Step 1
   - Step 2
   - Further Steps
   - Solution
5. Lessons Learned

`labeling` built-in

`labeling([S,E,N,D,M,O,R,Y])`

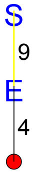
- Try variable is order given
- Try values starting from smallest value in domain
- When failing, backtrack to last open choice
- *Chronological Backtracking*
- *Depth First search*

Search Tree Step 1

Variable *S* already fixed

Step 2, Alternative $E = 4$

Variable $E \in \{4..7\}$, first value tested is 4



Assignment $E = 4$

	0	1	2	3	4	5	6	7	8	9
S										
E					✹	-	-	-		
N										
D										
M										
O										
R										
Y										

Propagation of $E = 4$, equality constraint

$$91 * 4 + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}$$

Propagation of $E = 4$, equality constraint

$$\underbrace{91 * 4 + 10 * R^{2..8} + D^{2..8}}_{386..452} = \underbrace{90 * N^{5..8} + Y^{2..8}}_{452..728}$$

Propagation of $E = 4$, equality constraint

$$\underbrace{91 * 4 + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452}$$

Propagation of $E = 4$, equality constraint

$$\underbrace{91 * 4 + 10 * R^{2..8} + D^{2..8} = 90 * N^{5..8} + Y^{2..8}}_{452}$$

$$N = 5, Y = 2, R = 8, D = 8$$

Result of equality propagation

	0	1	2	3	4	5	6	7	8	9
S										
E										
N						✹	-	-	-	
D			-	-	-	-	-	-	✹	
M										
O										
R			-	-	-	-	-	-	✹	
Y			✹	-	-	-	-	-	-	

Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S										
E										
N						✸	-	-	-	
D			-	-	-	-	-	-	✸	
M										
O										
R			-	-	-	-	-	-	✸	
Y			✸	-	-	-	-	-	-	

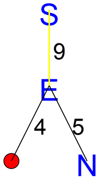
Propagation of alldifferent

	0	1	2	3	4	5	6	7	8	9
S									|	
E									|	
N						✸	-	-	|	
D			-	-	-	-	-	-	✸	
M									|	
O									|	
R			-	-	-	-	-	-	✸	
Y			✸	-	-	-	-	-	|	

Alldifferent fails!

Step 2, Alternative $E = 5$

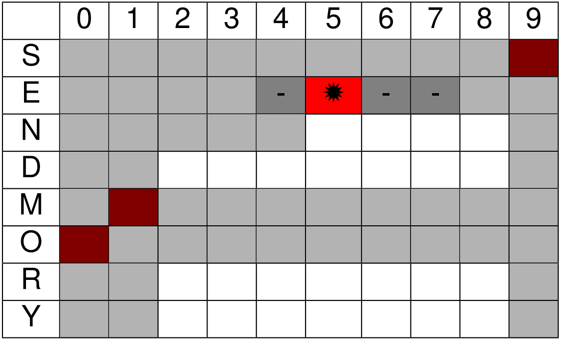
Return to last open choice, E, and test next value



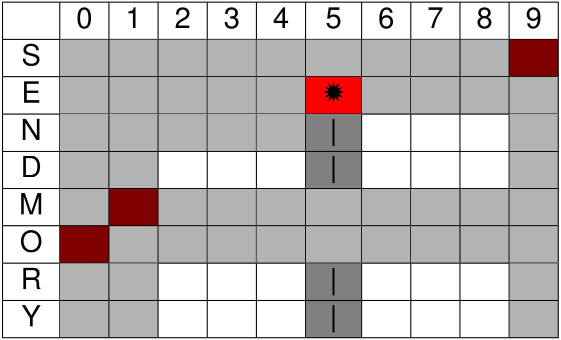
Assignment $E = 5$

	0	1	2	3	4	5	6	7	8	9
S										
E					-	✹	-	-		
N										
D										
M										
O										
R										
Y										

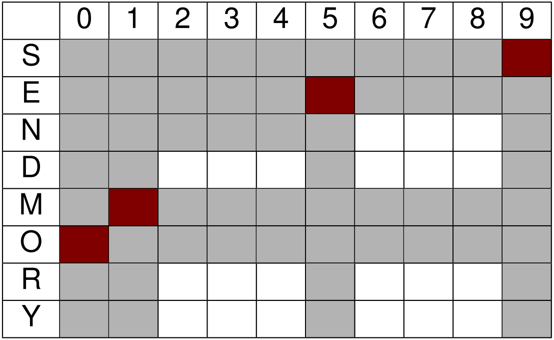
Propagation of alldifferent



Propagation of alldifferent



Propagation of alldifferent



$$N \neq 5, N \geq 6$$

Propagation of equality

$$91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}$$

Propagation of equality

$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8}}_{477..543} = \underbrace{90 * N^{6..8} + Y^{2..8}}_{542..728}$$

Propagation of equality

$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}}_{542..543}$$

Propagation of equality

$$\underbrace{91 * 5 + 10 * R^{2..8} + D^{2..8} = 90 * N^{6..8} + Y^{2..8}}_{542..543}$$

$$N = 6, Y \in \{2, 3\}, R = 8, D \in \{7..8\}$$

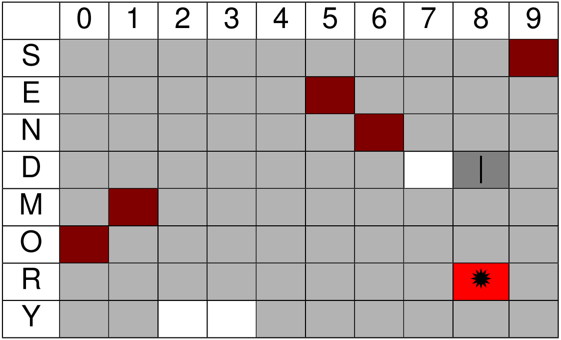
Result of equality propagation

	0	1	2	3	4	5	6	7	8	9
S										
E										
N							✸	-	-	
D			✖	✖	✖		✖			
M										
O										
R			-	-	-		-	-	✸	
Y					✖		✖	✖	✖	

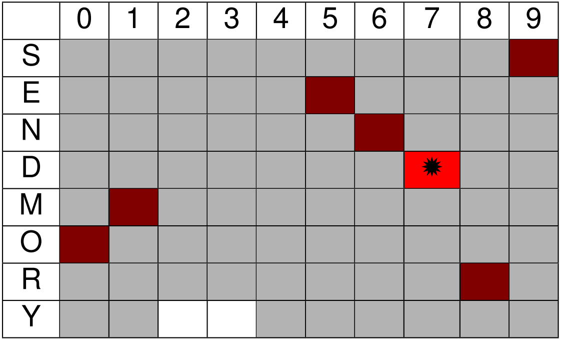
Propagation of `alldifferent`

	0	1	2	3	4	5	6	7	8	9
S										
E										
N							✹	-	-	
D			✖	✖	✖		✖			
M										
O										
R			-	-	-		-	-	✹	
Y					✖		✖	✖	✖	

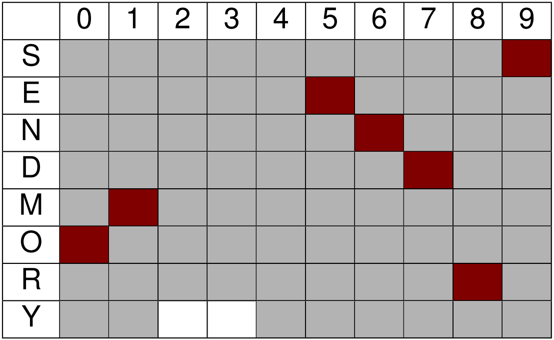
Propagation of `alldifferent`



Propagation of `alldifferent`



Propagation of `alldifferent`



$D = 7$

Propagation of equality

$$91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}$$

Propagation of equality

$$\underbrace{91 * 5 + 10 * 8 + 7}_{542} = \underbrace{90 * 6 + Y^{2..3}}_{542..543}$$

Propagation of equality

$$\underbrace{91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}}_{542}$$

Propagation of equality

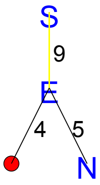
$$\underbrace{91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}}_{542}$$

$$Y = 2$$

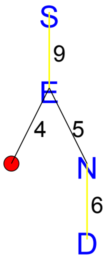
Last propagation step

	0	1	2	3	4	5	6	7	8	9
S										
E										
N										
D										
M										
O										
R										
Y			✸	-						

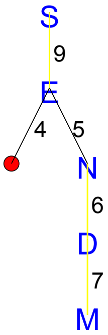
Further Steps: Nothing more to do



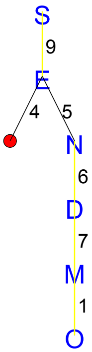
Further Steps: Nothing more to do



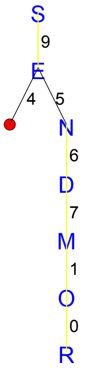
Further Steps: Nothing more to do



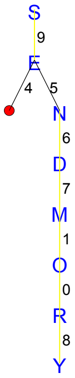
Further Steps: Nothing more to do



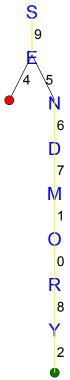
Further Steps: Nothing more to do



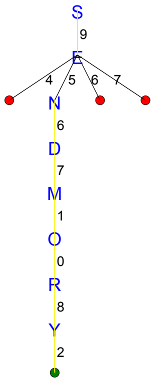
Further Steps: Nothing more to do



Further Steps: Nothing more to do



Complete Search Tree



Solution

	9	5	6	7
+	1	0	8	5
1	0	6	5	2

Strengths

- CP is excellent to explore highly constrained combinatorial spaces quickly
- Math programming is particulary good at deriving lower bounds
- LS is particualry good at derving upper bounds

Differences

- MILP models
  - impose modelling rules: linear inequalities and objectives
  - emphasis on tightness and compactness of LP, strength of bounds (remove dominated constraints)
- CP models
  - a large variety of algorithms communicating with each other: global constraints
  - more expressiveness
  - emphasis on exploiting substructres, include redundant constraints

Resume

- Constraint Satisfaction Problem
- Modelling in CP
- Examples, Send More Money, Sudoku

References

Anders T. and Miranda E.R. (2011). **Constraint programming systems for modeling music theories and composition**. *ACM Comput. Surv.*, 43(4), pp. 30:1–30:38.

Hooker J.N. (2011). **Hybrid modeling**. In *Hybrid Optimization*, edited by P.M. Pardalos, P. van Hentenryck, and M. Milano, vol. 45 of **Optimization and Its Applications**, pp. 11–62. Springer New York.

Smith B.M. (2006). **Modelling**. In *Handbook of Constraint Programming*, edited by F. Rossi, P. van Beek, and T. Walsh, chap. 11, pp. 377–406. Elsevier.

Williams H. and Yan H. (2001). **Representations of the all_different predicate of constraint satisfaction in integer programming**. *INFORMS Journal on Computing*, 13(2), pp. 96–103.