DM877

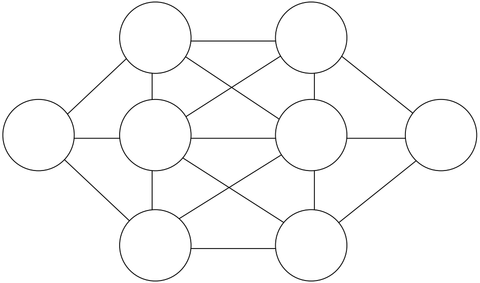
Discrete Optimization

Lecture 2
Constraint Programming Overview based on Examples

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline



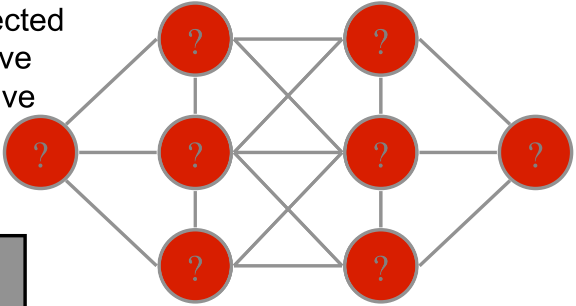
Put a different number in each circle (1 to 8) such that adjacent circles cannot take consecutive numbers

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

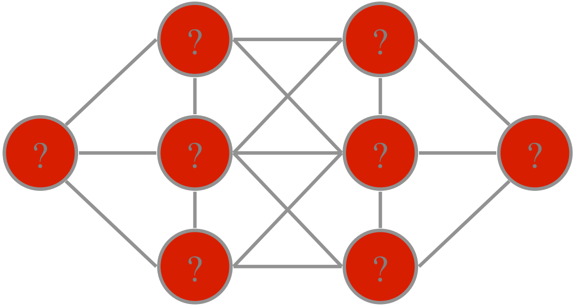
- Place numbers 1 through 8 on nodes
  - Each number appears exactly once
  - No connected nodes have consecutive numbers



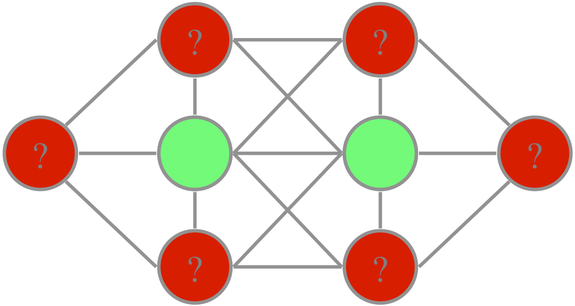
You have
8 minutes!

Heuristic Search

Which nodes are hardest to number?

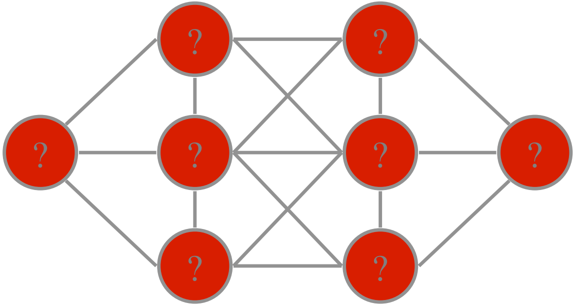


Heuristic Search



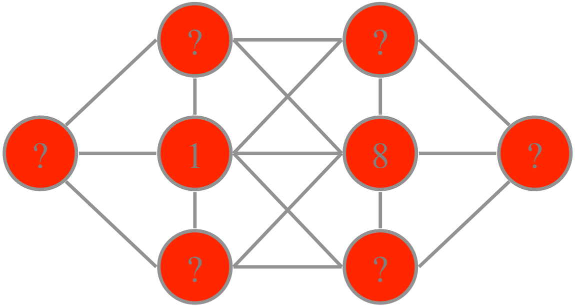
Heuristic Search

Which are the least constraining values to use?



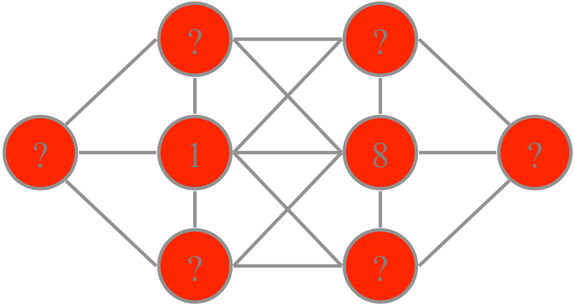
Heuristic Search

Values 1 and 8



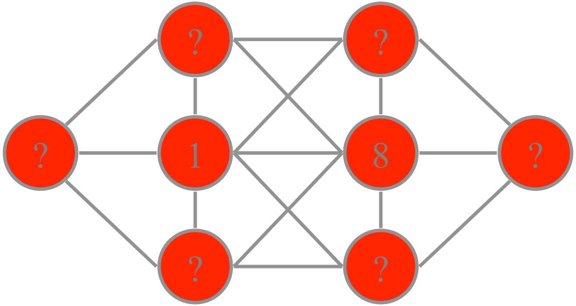
Heuristic Search

Values 1 and 8



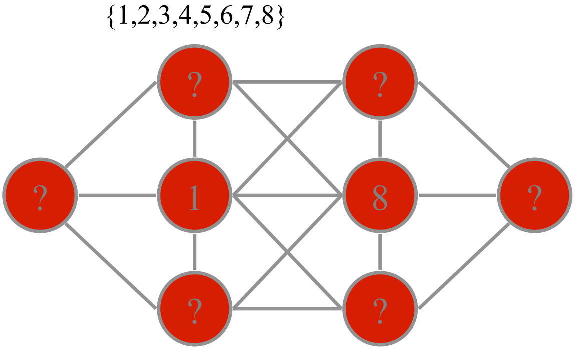
Symmetry means we don't need to consider: 8 1

Inference/propagation

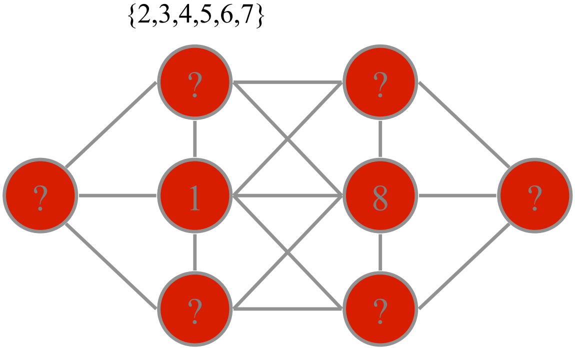


We can now eliminate many values for other nodes

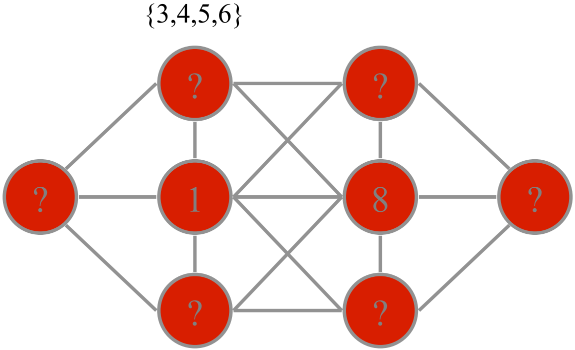
Inference/propagation



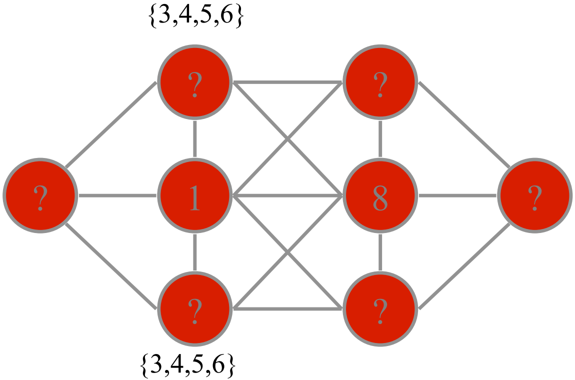
Inference/propagation



Inference/propagation

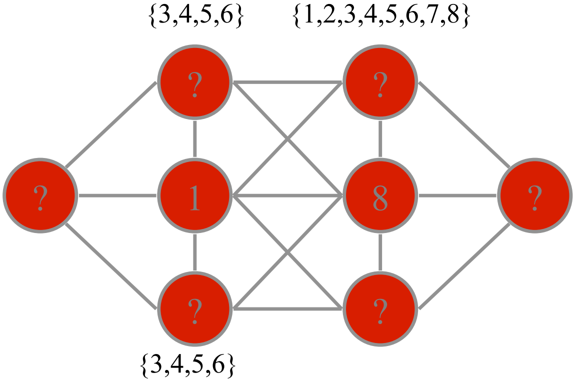


Inference/propagation

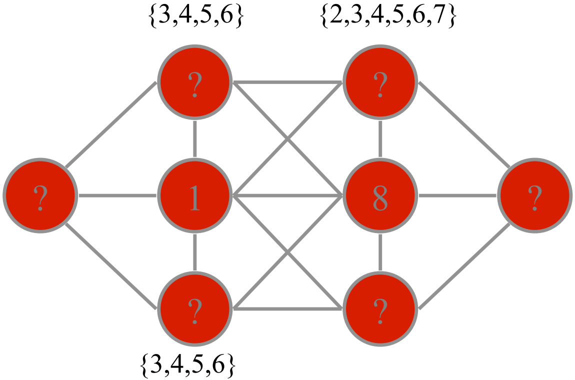


By symmetry

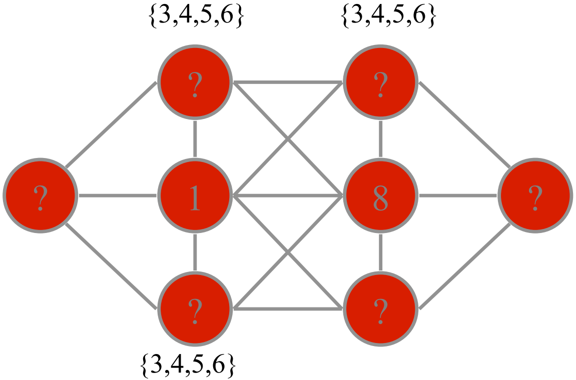
Inference/propagation



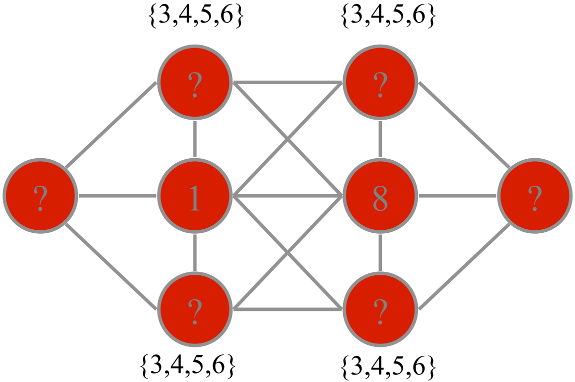
Inference/propagation



Inference/propagation

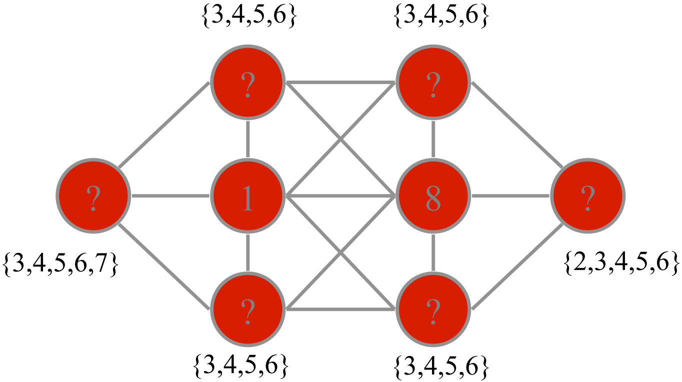


Inference/propagation

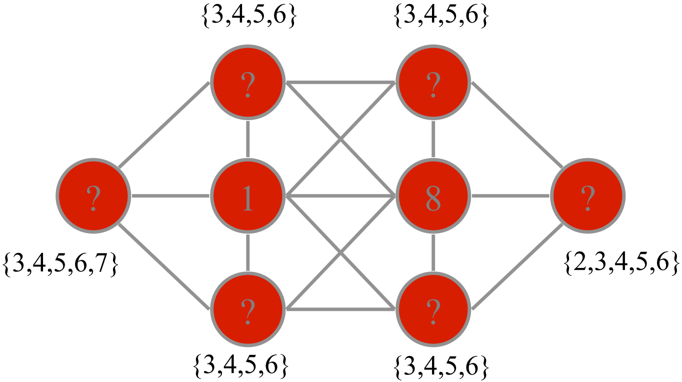


By symmetry

Inference/propagation

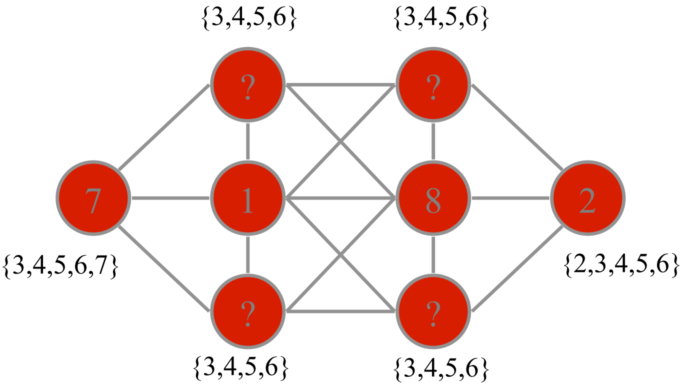


Inference/propagation



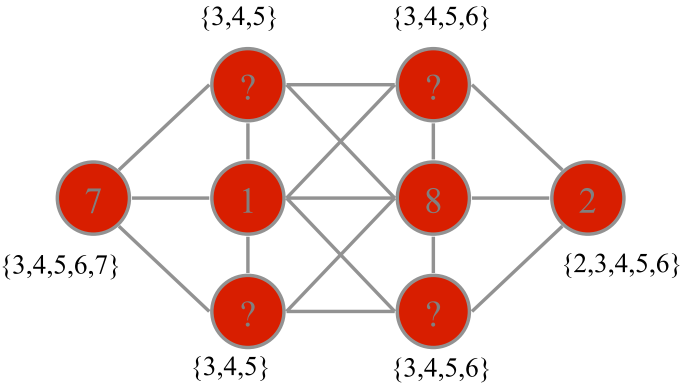
Value 2 and 7 are left in just one variable domain each

Inference/propagation



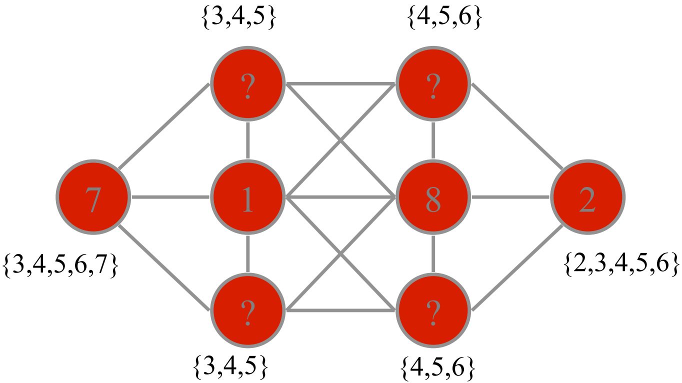
And propagate …

Inference/propagation



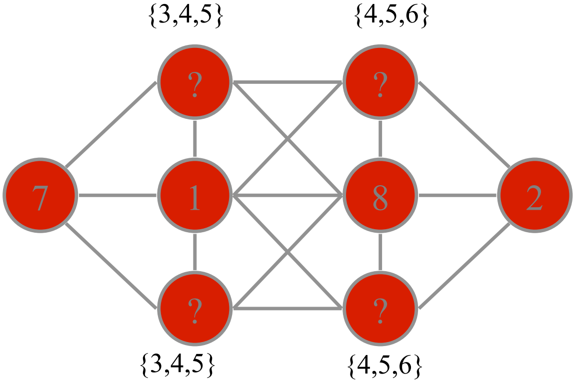
And propagate …

Inference/propagation



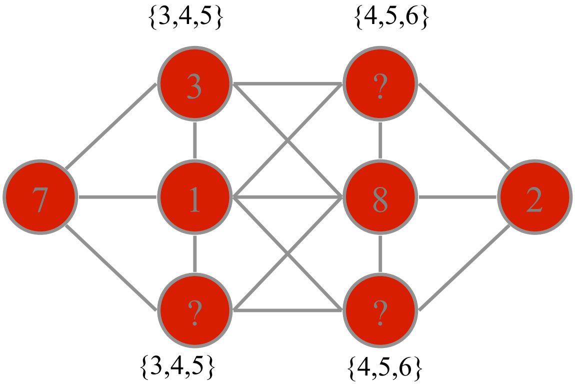
And propagate …

Inference/propagation



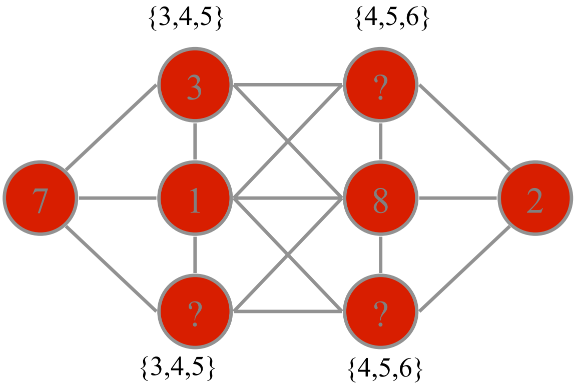
Guess a value, but be prepared to backtrack …

Inference/propagation



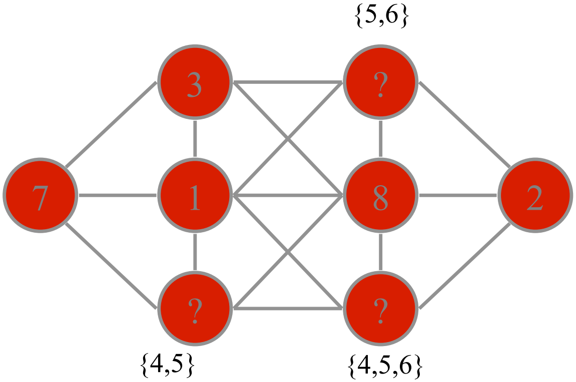
Guess a value, but be prepared to backtrack …

Inference/propagation



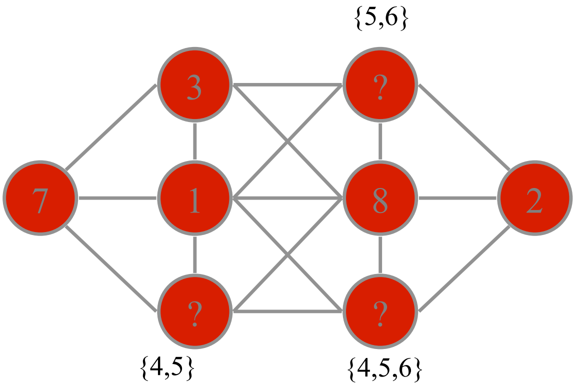
And propagate …

Inference/propagation



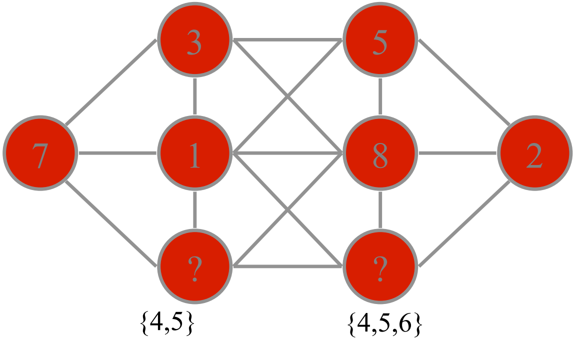
And propagate …

Inference/propagation



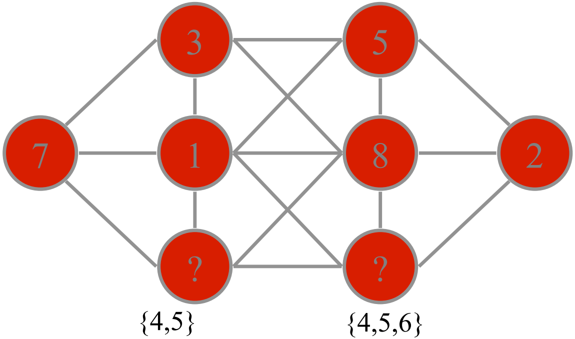
Guess another value …

Inference/propagation



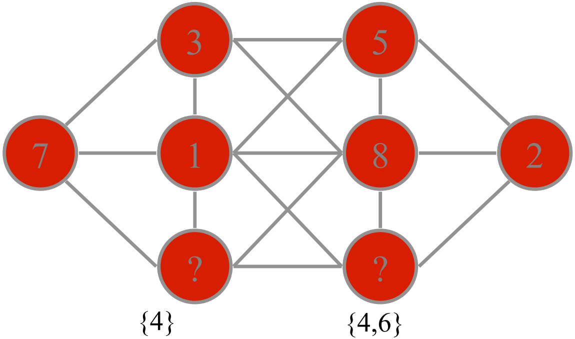
Guess another value …

Inference/propagation



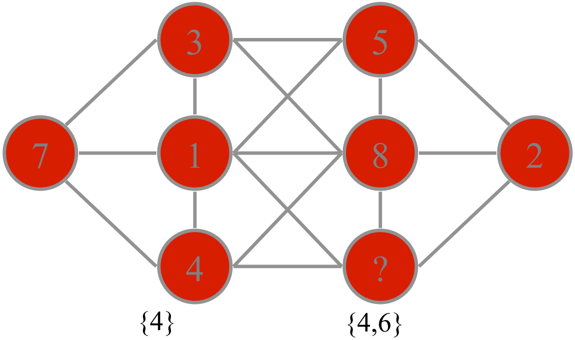
And propagate …

Inference/propagation



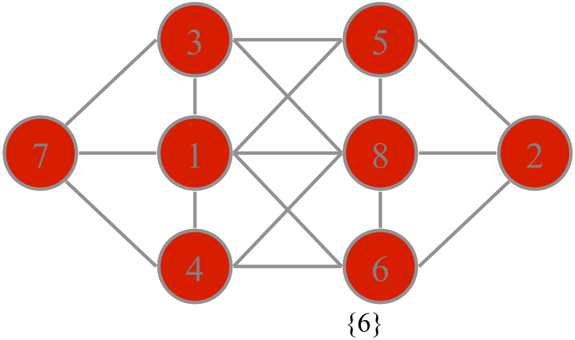
And propagate …

Inference/propagation

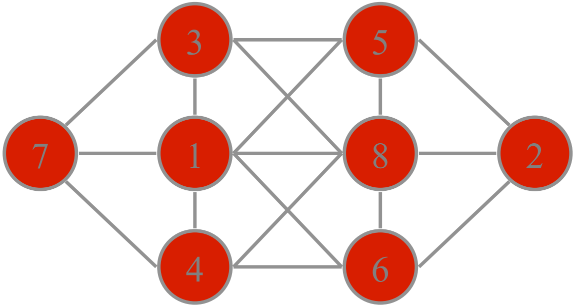


One node has only a single value left …

Inference/propagation



Solution

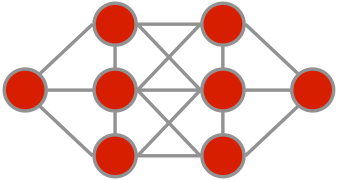


The Core of Constraint Computation

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

Hardness

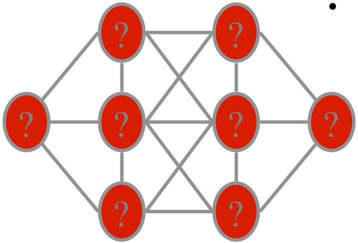
- The puzzle is actually a hard problem
  - NP-complete



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

    $|v_1 - v_2| > 1$

    $|v_1 - v_3| > 1$

    …

Outline

Constraint Programming - in a nutshell

- Declarative description of problems with
  - Variables which range over (finite) sets of values
  - Constraints over subsets of variables which restrict possible value combinations
  - A solution is a value assignment which satisfies all constraints
- Constraint propagation/reasoning
  - Removing inconsistent values for variables
  - Detect failure if constraint can not be satisfied
  - Interaction of constraints via shared variables
  - Incomplete
- Search
  - User controlled assignment of values to variables
  - Each step triggers constraint propagation
- Different domains require/allow different methods

Constraint Programming

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

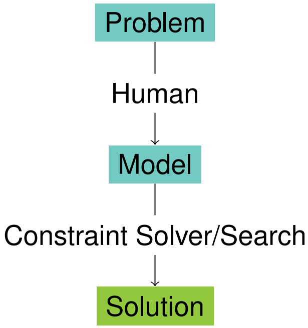
- Variables each with a finite set of possible values (domain)
- Constraint on a sequence of variables: a relationship on their domains

Constraint Satisfaction Problem: finite set of constraints

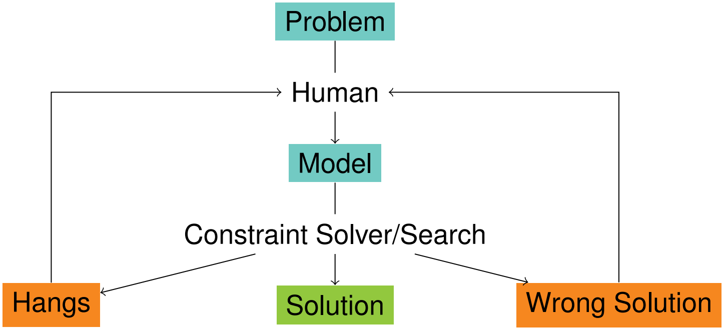
CP

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

Basic Process



More Realistic



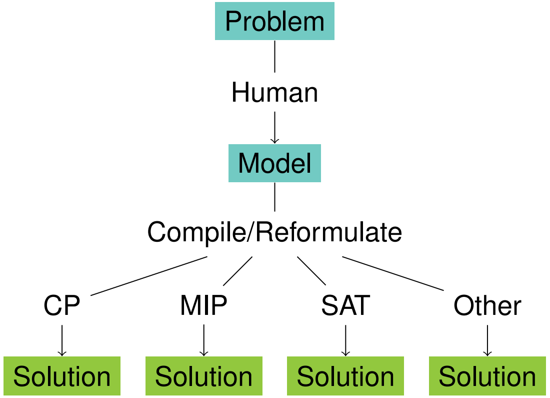
Dual Role of Model

- Allows Human to Express Problem
  - Close to Problem Domain
  - Constraints as Abstractions
- Allows Solver to Execute
  - Variables as Communication Mechanism
  - Constraints as Algorithms

Modelling Frameworks

- MiniZinc (NICTA, Australia)
- NumberJack (Insight, Ireland)
- Essence (UK)
- Allow use of multiple back-end solvers
- Compile model into variants for each solver
- A priori solver independent model(CP, MIP, SAT)

Framework Process



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
   - domains
   - constraints
   - objective function

   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 solved 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

Aims

- Example of Finite Domain Constraint Problem
- Models and Programs
- Constraint Propagation and Search
- Some Basic Constraints:
  linear arithmetic, alldifferent, disequality
- A Built-in search
- Visualizers for variables, constraints and search

Problem: 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.

Modelling

1. Parameters
2. Variables (ie, solution representation)
3. Domains (ie, allowed values for the variables)
4. Constraints

Later Objective Function

Model

- Each character is a variable, which ranges over the values 0 to 9.
- An alldifferent constraint between all variables, which states that two different variables must have different values. This is a very common constraint, which we will encounter in many other problems later on.
- Two disequality constraints (variable X must be different from value V) stating that the variables at the beginning of a number can not take the value 0.
- An arithmetic equality constraint linking all variables with the proper coefficients and stating that the equation must hold.

Send More Money: CP model

SEND + MORE = MONEY

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

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

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

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

- $X_M \neq 0$, $X_S \neq 0$

- Crypto constraint $\rightsquigarrow$ 1 equality constraint:

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

- 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

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]);
```

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))
```

Program Sendmory

```
:- module(sendmory).
:- export(sendmory/1).
:- lib(ic).
sendmory(L):-
        L = [S,E,N,D,M,O,R,Y],
        L :: 0..9,
        alldifferent(L),
        S #\= 0, M #\= 0,
        1000*S + 100*E + 10*N + D +
        1000*M + 100*O + 10*R + E #=
        10000*M + 1000*O + 100*N + 10*E + Y,
        labeling(L).
```

Question

But how does the program come up with a solution?

Constraint Setup

- Domain Definition
- Alldifferent Constraint
- Disequality Constraints
- Equality Constraint

The following slides are taken from H. Simonis: H. Simonis' demo, slides 33-134 and his tutorial at ACP2016.

Domain Definition

```
L = [S,E,N,D,M,O,R,Y],
L :: 0..9,
```

$$[S, E, N, D, M, O, R, Y] \in \{0..9\}$$

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

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 $O \in \{0..1\}$

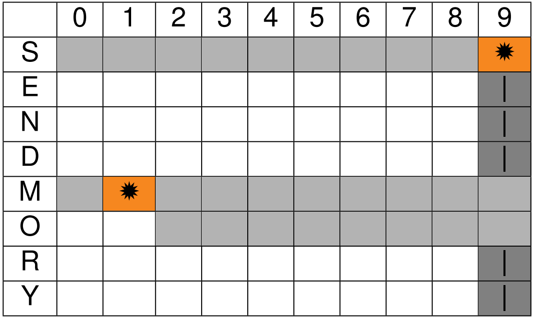
Propagation of equality: Result

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

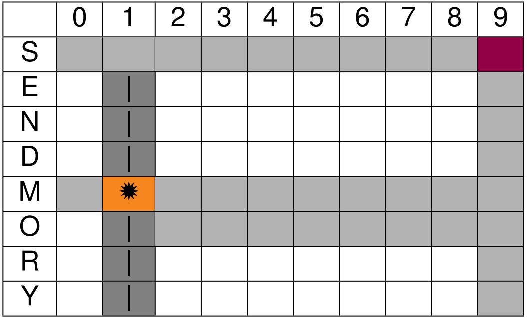
Propagation of alldifferent

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

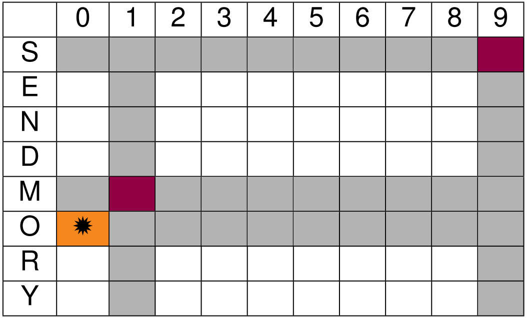
Propagation of alldifferent



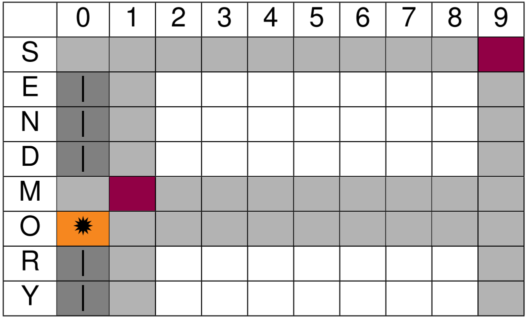
Propagation of alldifferent



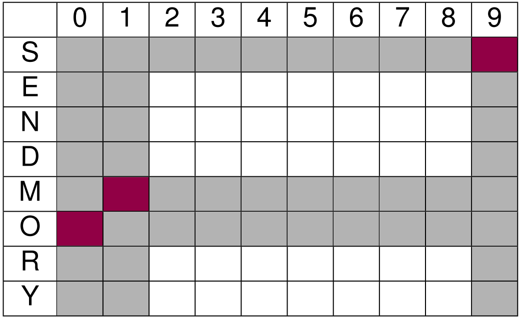
Propagation of alldifferent



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)

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

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)

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

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)

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

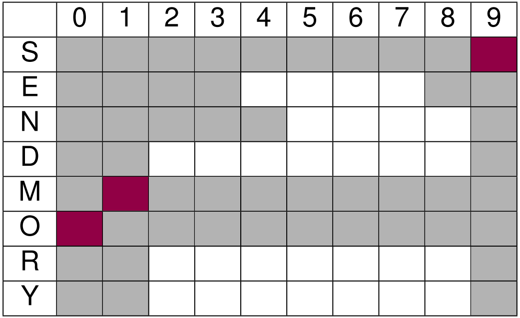
Propagation of equality (Iteration 6)

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

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

No further propagation at this point

Domains after setup



Outline

Problem

Program

Constraint Setup

Search

- Step 1
- Step 2
- Further Steps
- Solution

Points to Remember

`labeling` built-in

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

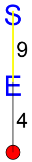
- Try variable is order given
- Try values starting from smallest value in domain
- When failing, backtrack to last open choice
- *Chronological Backtracking*
- *Depth First search*

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			✹	-	-	-	-	-	-	

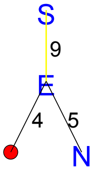
Propagation of alldifferent

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

Alldifferent fails!

Step 2, Alternative $E = 5$

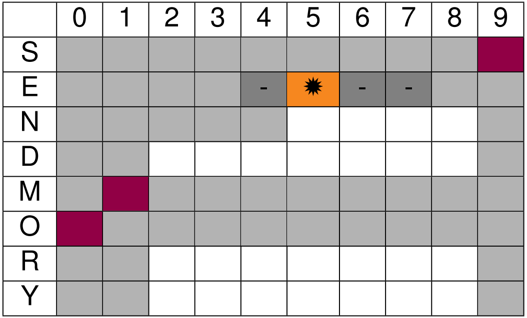
Return to last open choice, *E*, and test next value



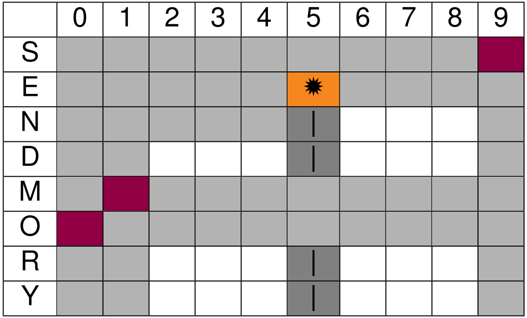
Assignment $E = 5$

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

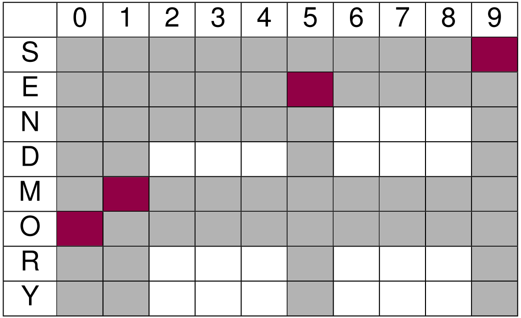
Propagation of alldifferent



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

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

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

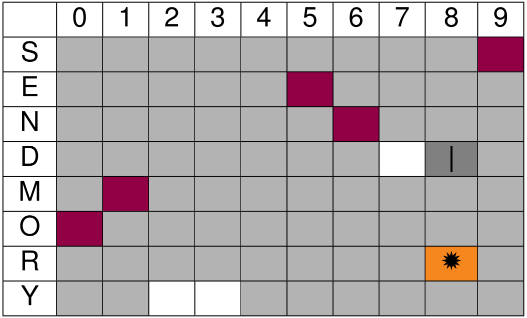
Result of equality propagation

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

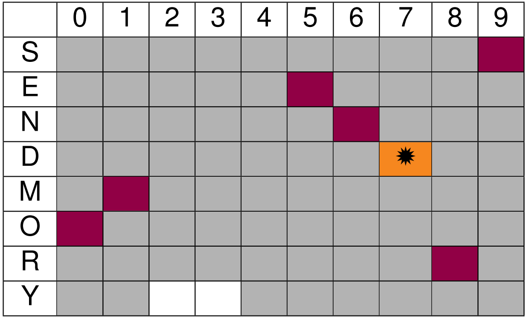
Propagation of `alldifferent`

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

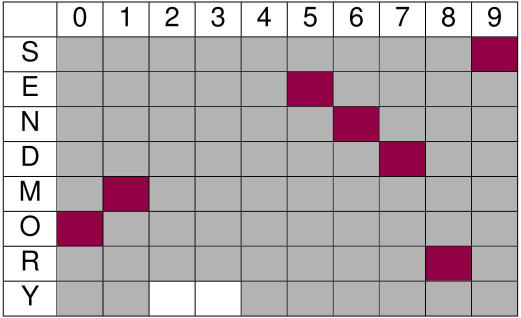
Propagation of `alldifferent`



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

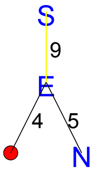
$$\underbrace{91 * 5 + 10 * 8 + 7 = 90 * 6 + Y^{2..3}}_{542}$$

$$Y = 2$$

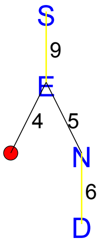
Last propagation step

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

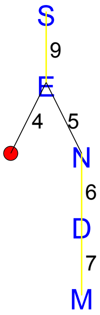
Further Steps: Nothing more to do



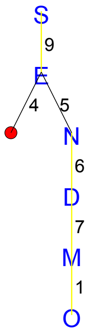
Further Steps: Nothing more to do



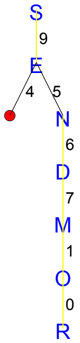
Further Steps: Nothing more to do



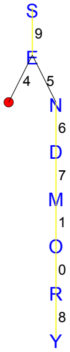
Further Steps: Nothing more to do



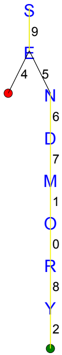
Further Steps: Nothing more to do



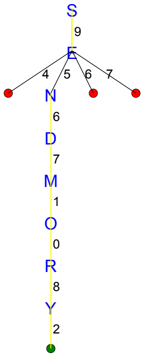
Further Steps: Nothing more to do



Further Steps: Nothing more to do



Complete Search Tree



Solution

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

Outline

Points to Remember

- Constraint models are expressed by:
  variables + constraints + parameters
- Problems can have many different models, which can behave quite differently. Choosing the best model is an art.
- Constraints can take many different forms.
- Propagation deals with the interaction of variables and constraints:
  It removes some values that are inconsistent with a constraint from the domain of a variable.
- Constraints only communicate via shared variables.

Points to Remember

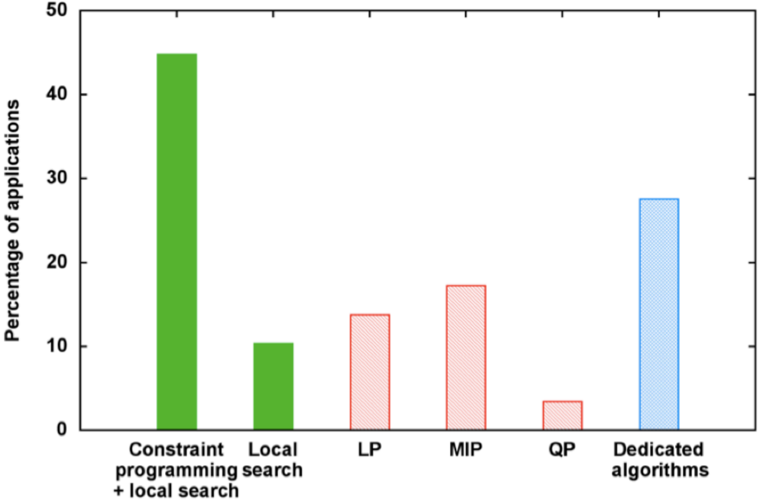
- Propagation is data driven, and can be quite complex even for small examples.
- Propagation usually is not sufficient, search may be required to find a solution.
- The default search uses chronological depth-first backtracking, systematically exploring the complete search space.
- The search choices and propagation are interleaved,
  after every choice some more propagation may further reduce the problem.

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



List of Contents

- Modeling with Finite Domain Integer Variables
- Introduction to MiniZinc
- Overview on global constraints
- Notions of local consistency
- Constraint propagation algorithms
- Filtering algorithms for global constraints
- Search
- Set variables
- Symmetries

Outline

1. An Initial Example

2. Constraint

3. Send More Money
   - Points to Remember
   - Modeling in MILP

Send More Money: ILP model 1

- $x_i \in \{0, \ldots, 9\}$ for all $i \in I = \{S, E, N, D, M, O, R, Y\}$
- $\delta_{ij} = \begin{cases} 0 & \text{if } x_i < x_j \\ 1 & \text{if } x_j < x_i \end{cases}$
- Crypto constraint:

$$\begin{array}{rrrrrl} & 10^3 x_1 & +10^2 x_2 & +10 x_3 & +x_4 & + \\ & 10^3 x_5 & +10^2 x_6 & +10 x_7 & +x_2 & = \\ \hline 10^4 x_5 & +10^3 x_6 & +10^2 x_3 & +10 x_2 & +x_8 & \end{array}$$

- Each letter takes a different digit:

$$\begin{aligned} x_i - x_j - 10\delta_{ij} &\leq -1, && \text{for all } i, j,\ i < j \\ x_j - x_i + 10\delta_{ij} &\leq 9, && \text{for all } i, j,\ i < j \end{aligned}$$

Send More Money: ILP model 2

- $x_i \in \{0, \ldots, 9\}$ for all $i \in I = \{S, E, N, D, M, O, R, Y\}$
- $y_{ij} \in \{0, 1\}$ for all $i \in I$, $j \in J = \{0, \ldots, 9\}$
- Crypto constraint:

$$\begin{array}{rrrrrr} & 10^3 x_1 & +10^2 x_2 & +10 x_3 & +x_4 & + \\ & 10^3 x_5 & +10^2 x_6 & +10 x_7 & +x_2 & = \\ \hline 10^4 x_5 & +10^3 x_6 & +10^2 x_3 & +10 x_2 & +x_8 & \end{array}$$

- Each letter takes a different digit:

$$\sum_{j \in J} y_{ij} = 1, \qquad \forall i \in I,$$

$$\sum_{i \in I} y_{ij} \leq 1, \qquad \forall j \in J,$$

$$x_i = \sum_{j \in J} j y_{ij}, \qquad \forall i \in I.$$

Send More Money: ILP model

The quality of these formulations depends on both the tightness of the LP relaxations and the number of constraints and variables (compactness)

- Which of the two models is tighter?
  project out all extra variables in the LP so that the polytope for LP is in the space of the x variables. By linear comb. of constraints:

Model 1

$$-1 \leq x_i - x_j \leq 10 - 1$$

Model 2

$$\sum_{j \in J} x_j \geq \frac{|J|\,(|J| - 1)}{2}, \qquad \forall J \subset I,$$

$$\sum_{j \in J} x_j \leq \frac{|J|\,(2k - |J|) + 1}{2}, \qquad \forall J \subset I.$$

- Can you find the convex hull of this problem?
  Williams and Yan [2001] prove that model 2 is facet defining

Suppose we want to maximize MONEY, how strong is the upper bound obtained with this formulation? How to obtain a stronger upper bound?

Send More Money: ILP model (**revisited**)

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

$$\begin{array}{rrrrrr} & 10^3 x_1 & +10^2 x_2 & +10 x_3 & +x_4 & + \\ & 10^3 x_5 & +10^2 x_6 & +10 x_7 & +x_2 & = \\ \hline 10^4 x_5 & +10^3 x_6 & +10^2 x_3 & +10 x_2 & +x_8 & \end{array}$$

- Each letter takes a different digit:

$$\sum_{j \in J} x_j \geq \frac{|J|\left(|J| - 1\right)}{2}, \qquad \forall J \subset I,$$

$$\sum_{j \in J} x_j \leq \frac{|J|\left(2k - |J|\right) + 1}{2}, \qquad \forall J \subset I.$$

But exponentially many!

Send More Money: CP model (revisited)

- $X_i \in \{0, \ldots, 9\}$ for all $i \in I = \{S, E, N, D, M, O, R, Y\}$
- $$\begin{array}{rrrrrl} & 10^3 X_1 & +10^2 X_2 & +10 X_3 & +X_4 & + \\ & 10^3 X_5 & +10^2 X_6 & +10 X_7 & +X_2 & = \\ \hline 10^4 X_5 & +10^3 X_6 & +10^2 X_3 & +10 X_2 & +X_8 & \end{array}$$
- $$\text{alldifferent}([X_1, X_2, \ldots, X_8]).$$
- 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 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))
```

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.