DM826 – Spring 2014

Modeling and Solving Constrained Optimization Problems

Lecture 2
Overview on CP and Examples

Combination

Simplification


Contradiction

Redundancy

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Resume

- First example: Sudoku
  first experience on modelling in MILP and CP
- SAT models
  - impose modelling rules (propositional calculus)
- 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

CP

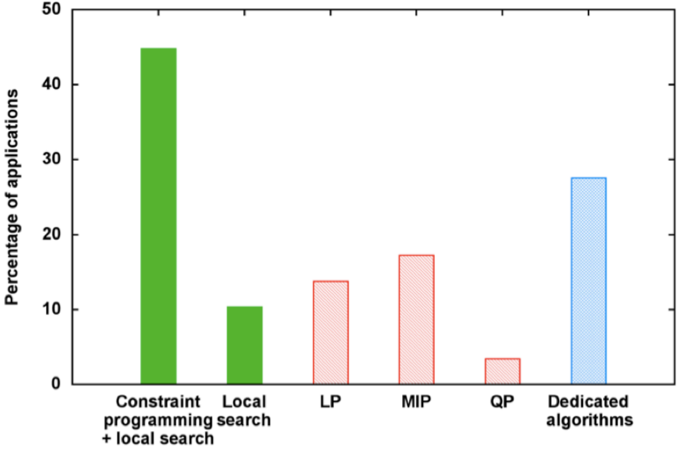
Constraint Programming = model (representation) + propagation (reasoning, inference) + search (reasoning, inference)

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
- 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

- Modeling
- Overview on global constraints
- Introduction to Gecode
- Notions of local consistency
- Constraint propagation algorithms
- Filtering algorithms for global constraints
- Search
- Set variables
- Symmetries
- Logic-Based Benders Decomposition and/or Large Neighborhood Search

Outline

Constraint Programming

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

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.

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 $x_i, \forall 1 \leq i \leq k$, and that this assignment satisfies C.

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 $x_i, \forall 1 \leq i \leq k$, and that this assignment satisfies C.
If $C = \emptyset$, we say that it is inconsistent.

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 $x_i, \forall 1 \leq i \leq k$, and that this assignment satisfies C.
If $C = \emptyset$, we say that it is inconsistent.

Extensional: specifies the satisfying tuples
Intensional: specifies the characteristic function

Constraint Programming

Constraint Satisfaction Problem (CSP)

A CSP is a finite set of variables X, together with a finite set of constraints C, each on a subset of X. A **solution** to a CSP is an assignment of a value $d \in D(x)$ to each $x \in X$, such that all constraints are satisfied simultaneously.

Constraint Programming

Constraint Satisfaction Problem (CSP)

A CSP is a finite set of variables X, together with a finite set of constraints C, each on a subset of X. A **solution** to a CSP is an assignment of a value $d \in D(x)$ to each $x \in X$, such that all constraints are satisfied simultaneously.

Constraint Optimization Problem (COP)

A COP is a CSP 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 P that minimizes (maximizes) the value of $f(d)$.

Task:

- determine whether the CSP/COP is consistent (has a solution):
- find one solutions
- 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 $\mathcal{P}$ can represent the same solution of P, if modelling introduces 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.

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:

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

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:

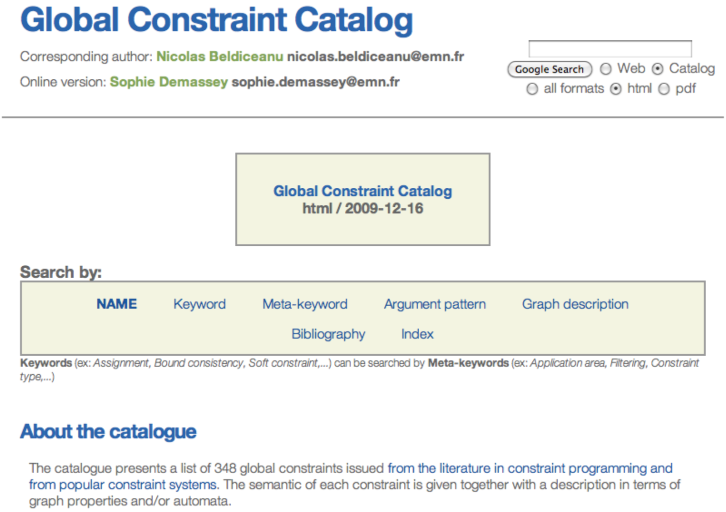
$$\texttt{alldifferent}(x_1, ..., x_n) = \{(d_1, ..., d_n) \mid \forall i, d_i \in D(x_i), \quad \forall i \neq j,\ d_i \neq d_j\}.$$

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.

Example: Send More Money

Send + More = Money

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

$$\begin{array}{cccccc} & S & E & N & D & + \\ & M & O & R & E & = \\ \hline M & O & N & E & Y & \end{array}$$

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

Can you model this problem as an IP?

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:

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

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:

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

- Each letter takes a different digit:

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

Send More Money: ILP model 2

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

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

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

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

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?

- Can you find the convex hull of this problem?

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

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

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

- Can you find the convex hull of this problem?

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

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

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

- 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)

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

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

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

Send More Money: ILP model (revisited)

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

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

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

But exponentially many!

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:

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

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

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

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

Send More Money: CP model (revisited)

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

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

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

- Redundant constraints (5 equality constraints)

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

Send More Money: CP model (revisited)

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

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

-

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

- 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?

Constraint Reasoning

Combination

Simplification

Contradiction

Redundancy

ILP model + CP propagation

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

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

- $$\begin{aligned} \sum_{j \in J} y_{ij} &= 1, && \forall i \in I, \\ \sum_{i \in I} y_{ij} &\leq 1, && \forall j \in J, \\ x_i &= \sum_{j \in J} j y_{ij}, && \forall i \in I. \end{aligned}$$

- Propagation adds valid inequalities:

$$LB(X_i) \leq x_i \leq UB(X_i) \text{ for all } i \in I$$

.

- H. Simonis' demo, slides 42-56

The convex hull of `alldifferent`

Convex Hull of of alldifferent

Given a set $I = \{1, \ldots, n\}$ (variable indices) and a set $D = \{0, \ldots, k\}$ with $k \geq n$, we consider

$$\text{alldifferent}([x_1, \ldots, x_n]), \text{ with } 0 \leq x_i \leq k.$$

all the facets of the previous ILP formulation for the `alldifferent` constraint are

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

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

[Williams and Yan [2001]]

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:

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

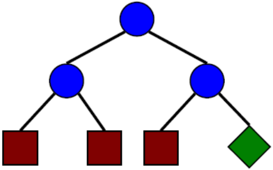
```
from gecode import *

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]

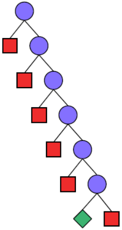
```
include "alldifferent.mzn";

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"];
```

Gecode - gist



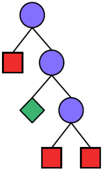
The inner nodes (blue circles) are choices, the red square leaf nodes are failures, and the green diamond leaf node is a solution.

Lexicographic



```
    SEND
+   MORE
--------
= MONEY

    9567
+   1085
--------
=  10652
```

First-fail



```
    SEND
+   MORE
--------
= MONEY

    9567
+   1085
--------
=  10652
```

Exercise

Can you try to solve:

$$\texttt{Gerald} + \texttt{Donald} = \texttt{Robert}$$

Viewpoints

A viewpoint is a pair $< \mathcal{X}, \mathcal{DE} >$, i.e. a set of variables and their domains

- Given a viewpoint, the constraints have to restrict the solutions of $\mathcal{P}$ to solutions of P
  - So the constraints are (to some extent) decided by the viewpoint
  - Different viewpoints give very different models
- We can combine viewpoints - more later
- Good rule of thumb: choose a viewpoint that allows the constraints to be expressed easily and concisely
  - will propagate well, so problem can be solved efficiently

Modelling

- Different views to the problem
- Adding implied constraints
- Auxiliary variables to make it easier to state constraints and improve constraint propagation

A Puzzle Example


```
 SEND +
 MORE =
MONEY
```

Two representations

- The first yields initially a weaker constraint propagation. The tree has 23 nodes and the unique solution is found after visiting 19 nodes
- The second representation has a tree with 29 nodes and the unique solution is found after visiting 23 nodes

A Puzzle Example

$$\begin{array}{r} \texttt{SEND}\ + \\ \texttt{MORE}\ = \\ \texttt{MONEY} \end{array}$$

Two representations

- The first yields initially a weaker constraint propagation. The tree has 23 nodes and the unique solution is found after visiting 19 nodes
- The second representation has a tree with 29 nodes and the unique solution is found after visiting 23 nodes

However for the puzzle $\texttt{GERALD} + \texttt{DONALD} = \texttt{ROBERT}$ the situation is reverse. The first has 16651 nodes and 13795 visits while the second has 869 nodes and 791 visits

$\leadsto$ Finding the best model is an empirical science

Guidelines

Rules of thumbs for modelling (to take with a grain of salt):

- use representations that involve less variables and simpler constraints for which constraint propagators are readily available

- use constraint propagation techniques that require less preprocessing (ie, the introduction of auxiliary variables) since they reduce the search space better.

  Disjunctive constraints may lead to an inefficient representation since they can generate a large search space.

- use global constraints

References

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.