Symbolic-Numeric Algorithms for Constraint Solving:

Overview, Challenges, Applications, Future

Martine Ceberio

University of Texas at El Paso

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Context: Continuous Constraints

- Continuous constraints:
  - *Same as discrete ones, except for the domains of their variables: continuous*

Context: Continuous Constraints

- Continuous constraints:
  - *Same as discrete ones, except for the domains of their variables: continuous*
- Example:
  - *a chemistry problem:*

$$
\begin{aligned}
14 * z1^2 + 6 * z1 * z2 + 5 * z1 - 72 * z2^2 - 18 * z2 &= 850 * z3 - 2.0e - 9, \\
0.5 * z1 * z2^2 + 0.01 * z1 * z2 + 0.13 * z2^2 + 0.04 * z2 &= 4.0e4, \\
0.03 * z1 * z3 + 0.04 * z3 &= 850
\end{aligned}
$$

$$z_i \in [-1000; +1000]$$

Solving Continuous Constraints

- Enumeration is not an option

Solving Continuous Constraints

- Enumeration is not an option
- We use intervals to guarantee a complete search:
  - *The search goes from the complete search space down to solutions*

Solving Continuous Constraints

- Enumeration is not an option
- We use intervals to guarantee a complete search:
  - *The search goes from the complete search space down to solutions*
- What do we do with these intervals during the search to find solutions?

Solving Continuous Constraints

- Enumeration is not an option
- We use intervals to guarantee a complete search:
  - *The search goes from the complete search space down to solutions*
- What do we do with these intervals during the search to find solutions?
  - *Intervals are contracted (possibly discarded) along the search*

Symbolic-Numeric Algorithms

- Numeric part of the algorithms = the interval computations to contract the domains

Symbolic-Numeric Algorithms

- Numeric part of the algorithms = the interval computations to contract the domains
- Symbolic part of the algorithms = the design of these contractors, and more

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c:\; f(x)=0$*, where* $x \in D$*.*

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c: \ f(x) = 0$, *where* $x \in D$.

$$N(D) \ = \ D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f'}(D)} \right)$$

Approaches to domain contraction (1)

Let us assume we want to <u>find zeros</u>.

- Historically: the **<u>Newton</u>** method

  Let's assume $c: \; f(x) = 0$*, where* $x \in D$*.*

$$N(D) \; = \; D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f'}(D)} \right)$$

- Limitations:

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c: \ f(x) = 0$*, where* $x \in D$*.*

$$N(D) \ = \ D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f'}(D)} \right)$$

- Limitations:
  - What if $0 \in \boldsymbol{f'}(D)$?

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c: \ f(x) = 0$*, where* $x \in D$*.*

$$N(D) \ = \ D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f'}(D)} \right)$$

- Limitations:
  - What if $0 \in \boldsymbol{f'}(D)$? $\rightsquigarrow$ no contraction

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c:\ f(x)=0$*, where* $x \in D$*.*

$$N(D) \ = \ D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f}'(D)} \right)$$

- Limitations:
  - What if $0 \in \boldsymbol{f}'(D)$? $\rightsquigarrow$ no contraction
- Reasons for this:

Approaches to domain contraction (1)

Let us assume we want to find zeros.

- Historically: the **Newton** method

  Let's assume $c: \; f(x) = 0$, *where* $x \in D$.

$$N(D) \;=\; D \cap \left(\boldsymbol{m}(D) - \frac{\boldsymbol{f}(m(D))}{\boldsymbol{f'}(D)} \right)$$

- Limitations:
  - What if $0 \in \boldsymbol{f'}(D)$? $\rightsquigarrow$ no contraction
- Reasons for this:
  - D is so large that f is not monotonic on D
  - $\boldsymbol{f'}(D)$ is overestimated, hence including a 0 while it is not part of the range

Approach to domain contraction (2)

Let us assume we have a <u>linear constraint</u> to solve.

- Let our constraint be:

$$\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_n} x_n = \boldsymbol{b}$$

Approach to domain contraction (2)

Let us assume we have a *<u>linear constraint</u>* *to solve.*

- Let our constraint be:

$$\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_n} x_n = \boldsymbol{b}$$

- Natural approach: the Gauss-Seidel method:

$$x_i = \frac{\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_{i-1}} x_{i-1} + \boldsymbol{a_{i+1}} x_{i+1} + \ldots + \boldsymbol{a_n} x_n}{\boldsymbol{a_i}}$$

Approach to domain contraction (2)

Let us assume we have a linear constraint to solve.

- Let our constraint be:

$$\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_n} x_n = \boldsymbol{b}$$

- Natural approach: the Gauss-Seidel method:

$$x_i = \frac{\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_{i-1}} x_{i-1} + \boldsymbol{a_{i+1}} x_{i+1} + \ldots + \boldsymbol{a_n} x_n}{\boldsymbol{a_i}}$$

= Symbolic inversion on x_i for a numerical contraction

Approach to domain contraction (2)

Let us assume we have a <u>linear constraint</u> to solve.

- Let our constraint be:

$$\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_n} x_n = \boldsymbol{b}$$

- Natural approach: the Gauss-Seidel method:

$$x_i = \frac{\boldsymbol{a_1} x_1 + \boldsymbol{a_2} x_2 + \ldots + \boldsymbol{a_{i-1}} x_{i-1} + \boldsymbol{a_{i+1}} x_{i+1} + \ldots + \boldsymbol{a_n} x_n}{\boldsymbol{a_i}}$$

= Symbolic inversion on x_i for a numerical contraction

- Limitation: What if the constraint is not linear?

Approach to domain contraction (3)

Let us assume we have a <u>nonlinear constraint</u> to solve.

- Let our constraint be:

$$cos(x) + 3ztan(y) = 2 \;\rightsquigarrow\; x = cos^{-1}(2 - 3ztan(y))$$

= symbolic inversion of (nonlinear) constraints

Approach to domain contraction (3)

Let us assume we have a nonlinear constraint to solve.

- Let our constraint be:

$$cos(x) + 3ztan(y) = 2 \;\rightsquigarrow\; x = cos^{-1}(2 - 3ztan(y))$$

  = symbolic inversion of (nonlinear) constraints

- Different approaches to such inversion:
  - Kearfott 1991: decomposition of the constraints into primitive, easily invertible constraints
  - Ceberio & Granvilliers 2000: recursive inversion of terms
  - Hansen & Walster 2003: inversion of f as $f = g - h$ where g is easy to invert

Inverted constraints

- Provide with an approximation of the domain of each variable = projection of the constraint (subset) onto each variable's dimension

Inverted constraints

- Provide with an approximation of the domain of each variable = projection of the constraint (subset) onto each variable's dimension
  - *Example:* $x^2 + y^2 - 1 = 0$ *with* $x,\ y\ \in [0, 2]$
    $y = (1 - x^2)^{1/2} = (1 - [0, 4])^{1/2} = [-1, 1]$

Inverted constraints

- Provide with an approximation of the domain of each variable = projection of the constraint (subset) onto each variable's dimension
  - *Example:* $x^2 + y^2 - 1 = 0$ *with* $x,\ y\ \in [0, 2]$
    $y = (1 - x^2)^{1/2} = (1 - [0, 4])^{1/2} = [-1, 1]$
    $\leadsto$ *since we know that* $y \in [0, 2]$, *now we know that the consistent part of this domain is* $[0, 1]$
    We can do the same with x *and narrow it scope to* $[0, 1]$

Inverted constraints

- Provide with an approximation of the domain of each variable = projection of the constraint (subset) onto each variable's dimension
  - *Example:* $x^2 + y^2 - 1 = 0$ *with* $x,\ y\ \in [0, 2]$
    $y = (1 - x^2)^{1/2} = (1 - [0, 4])^{1/2} = [-1, 1]$
    $\leadsto$ *since we know that* $y \in [0, 2]$, *now we know that the consistent part of this domain is* $[0, 1]$
    We can do the same with x *and narrow it scope to* $[0, 1]$
- “One" approximation per variable and per constraint
  - *Each constraint contributes to reducing the domain of each variable*

Inverted constraints

- Provide with an approximation of the domain of each variable = projection of the constraint (subset) onto each variable's dimension
  - *Example:* $x^2 + y^2 - 1 = 0$ *with* $x,\ y\ \in [0, 2]$
    $y = (1 - x^2)^{1/2} = (1 - [0, 4])^{1/2} = [-1, 1]$
    $\rightsquigarrow$ *since we know that* $y \in [0, 2]$*, now we know that the consistent part of this domain is* $[0, 1]$
    We can do the same with x *and narrow it scope to* $[0, 1]$
- “One" approximation per variable and per constraint
  - *Each constraint contributes to reducing the domain of each variable*
- Problem with this approach:
  - *Result = intersection of approximations*
  - *What we aim at = approximation of intersection*

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Challenges

- Weaknesses of interval computations:

  Overestimation leads to slower solving processes

- Weaknesses of the domain contractions ($\equiv$ constraint consistency techniques):

  The locality of reasonings leads to slower solving processes

Approaches (1)

Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations

Approaches (1)

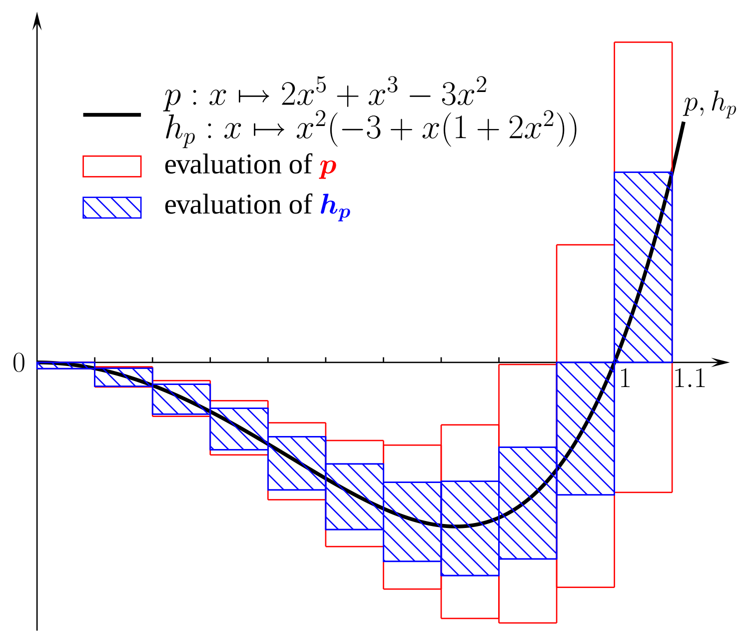
Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*

Approaches (1)

Efficiency of interval computations:

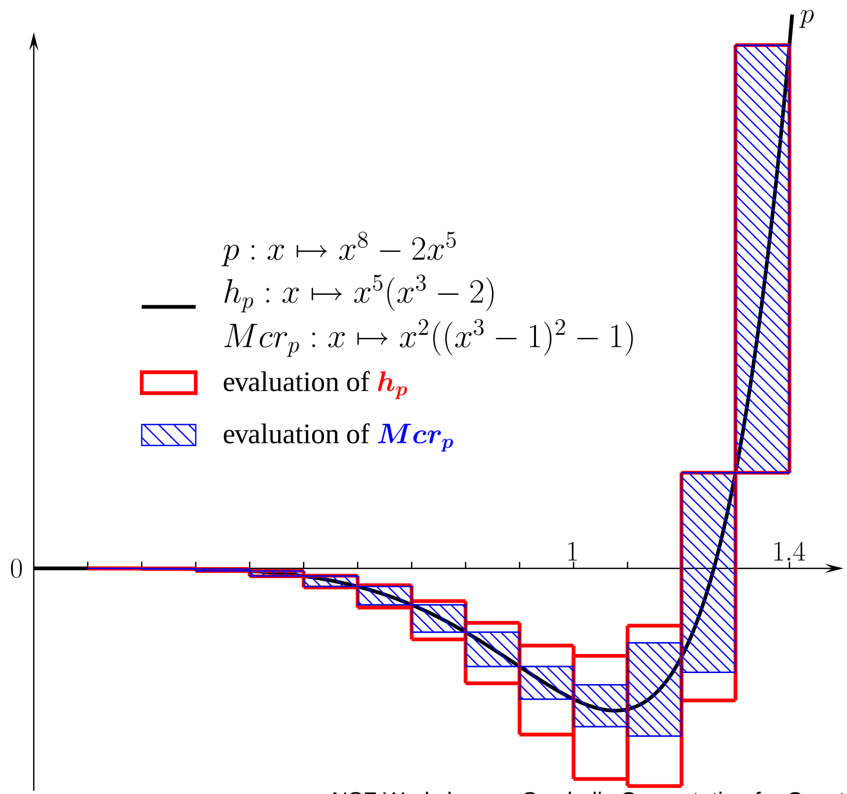
- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*



Approaches (1)

Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*



Approaches (1)

Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*
  - *But: still not exact (NP-hard to achieve)*

Approaches (1)

Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*
  - *<u>But:</u> still not exact (NP-hard to achieve)*
- Otherwise: shaving technique
  - *Decomposition of intervals into smaller ones that are evaluated*
  - *Relies on the inclusion property of interval arithmetic*
  - *Achieves better interval evaluations*

Approaches (1)

Efficiency of interval computations:

- Overestimation: syntax-dependent evaluations
- Therefore: symbolic pre-processing techniques
  - *Horner; Hong & Stahl; Ceberio & Granvilliers*
  - *<u>But:</u> still not exact (NP-hard to achieve)*
- Otherwise: shaving technique
  - *Decomposition of intervals into smaller ones that are evaluated*
  - *Relies on the inclusion property of interval arithmetic*
  - *Achieves better interval evaluations*
  - *<u>But:</u> expensive*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *“the more the better”*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *“the more the better”*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$C : \begin{cases} c_1 : & x + y + x^2 + xy + y^2 & = & 0 \\ c_2 : & x + t + xy + t^2 + x^2 & = & 0 \\ c_3 : & y + z + x^2 + z^2 & = & 0 \\ c_4 : & x + z + x^2 + y^2 + z^2 + xy & = & 0 \end{cases}$$

defined over $E = [-100, 100]^4$,
4 solutions reached in 140ms using **realpaver** [Granvilliers, 2002].

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *“the more the better”*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$\left\{\begin{array}{lllllllllll} lc_1: & x & +y & & & +u_1 & +u_2 & +u_3 & & & =0 \\ lc_2: & x & & & +t & +u_1 & +u_2 & & +u_4 & & =0 \\ lc_3: & & y & +z & & +u_1 & & & & +u_5 & =0 \\ lc_4: & x & & +z & & +u_1 & +u_2 & +u_3 & & +u_5 & =0 \end{array}\right.$$

and the abstracted system: $\left\{\begin{array}{lll} u_1 & = & x^2 \\ u_2 & = & xy \\ u_3 & = & y^2 \\ u_4 & = & t^2 \\ u_5 & = & z^2 \end{array}\right.$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *“the more the better”*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$
\boxed{\left\{\begin{array}{lcccccccccc}
lc_1: & u_1 & +y & & & & & +x & +u_2 & +\ u_3 & =0 \\
lc'_2: & & y & & & -t & -u_4 & & & +\ u_3 & =0 \\
lc'_3: & & & -z & -u_5 & & & +x & +u_2 & +\ u_3 & =0 \\
lc'_4: & & & & & -t & -u_4 & +x & +u_2 & +2u_3 & =0
\end{array}\right.}
$$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$
\boxed{\left\{\begin{array}{lcccccccccl}
lc_1: & u_1 & +y & & & & & +x & +u_2 & +\ u_3 & =0 \\
lc'_2: & & y & & & -t & -u_4 & & & +\ u_3 & =0 \\
lc'_3: & & & -z & -u_5 & & & +x & +u_2 & +\ u_3 & =0 \\
lc'_4: & & & & & -t & -u_4 & +x & +u_2 & +2u_3 & =0
\end{array}\right.}
$$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$\boxed{\left\{\begin{array}{lcccccccccc}
lc_1 : & x^2 & +y & & & & & +x & +xy & +\ y^2 & =0 \\
lc'_2 : & & y & & & -t & -t^2 & & & +\ y^2 & =0 \\
lc'_3 : & & & -z & -z^2 & & & +x & +xy & +\ y^2 & =0 \\
lc'_4 : & & & & & -t & -t^2 & +x & +xy & +2y^2 & =0
\end{array}\right.}$$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

$$\boxed{\left\{\begin{array}{llllllllllll} lc_1: & x^2 & +y & & & & & +x & +xy & + & y^2 & =0 \\ lc'_2: & & y & & & -t & -t^2 & & & + & y^2 & =0 \\ lc'_3: & & & -z & -z^2 & & & +x & +xy & + & y^2 & =0 \\ lc'_4: & & & & & -t & -t^2 & +x & +xy & & +2y^2 & =0 \end{array}\right.}$$

The new system is solved in 240*ms!!*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Let us consider again the same problem. We begin with the linearized system:

$$\boxed{\left\{\begin{array}{llllllllll} lc_1: & x & +y & & & +u_1 & +u_2 & +u_3 & & = 0 \\ lc_2: & x & & & +t & +u_1 & +u_2 & & +u_4 & = 0 \\ lc_3: & & y & +z & & +u_1 & & & & +u_5 = 0 \\ lc_4: & x & & +z & & +u_1 & +u_2 & +u_3 & & +u_5 = 0 \end{array}\right.}$$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *“the more the better”*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

First step of elimination

$$\boxed{\left\{\begin{array}{llllllllllll} \boldsymbol{lc_3}: & & y & +z & & +u_1 & & & & +\boldsymbol{u_5} & = & 0 \\ lc_1: & x & +y & & & +u_1 & +u_2 & +u_3 & & & = & 0 \\ lc_2: & x & & & +t & +u_1 & +u_2 & & +u_4 & & = & 0 \\ lc_4': & -x & +y & & & & -u_2 & -u_3 & & & = & 0 \end{array}\right.}$$

Control criterion: *controls the densification of the “linear” system*

User linear part: 0

Abstracted linear part: -2

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Second step of elimination

$$\boxed{\left\{\begin{array}{lccccccccccc} lc_3: & & y & +z & & +u_1 & & & & +u_5 & = & 0 \\ \boldsymbol{lc'_4}: & -x & +y & & & & -u_2 & \boldsymbol{-u_3} & & & = & 0 \\ lc'_1: & & 2y & & & +u_1 & & & & & = & 0 \\ lc_2: & x & & & +t & +u_1 & +u_2 & & +u_4 & & = & 0 \end{array}\right.}$$

Control criterion: *controls the densification of the "linear" system*

User linear part: 0

Abstracted linear part: -1

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Third step of elimination

$$\begin{cases} lc_3: & & y & +z & & +u_1 & & & & +u_5 & = & 0 \\ lc'_4: & -x & +y & & & & -u_2 & -u_3 & & & = & 0 \\ \boldsymbol{lc'_1}: & & 2y & & & +\boldsymbol{u_1} & & & & & = & 0 \\ lc'_2: & -x & +2y & & -t & & -u_2 & & -u_4 & & = & 0 \end{cases}$$

Control criterion: *controls the densification of the "linear" system*

User linear part: $+1$

Abstracted linear part: -1

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Triangularized system

$$\boxed{\left\{\begin{array}{llllllllllll} lc_2' : & -t & -x & -u_2 & & -u_4 & & & & +2y & = & 0 \\ lc_4' : & & -x & -u_2 & -u_3 & & & & & +y & = & 0 \\ lc_3 : & & & & & & +u_5 & +z & +u_1 & +\ y & = & 0 \\ lc_1' : & & & & & & & & +u_1 & +2y & = & 0 \end{array}\right.}$$

Concretization: *nonlinear terms are restored*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Concretization phase

$$\boxed{\left\{\begin{array}{lcccccccccccc}
c'_1: & -t & -x & -xy & & -t^2 & & & & +2y & = & 0\\
c'_2: & & -x & -xy & -y^2 & & & & & +y & = & 0\\
c'_3: & & & & & & z^2 & +z & +x^2 & +\,y & = & 0\\
c'_4: & & & & & & & & x^2 & +2y & = & 0
\end{array}\right.}$$

Post-processing: *simplification of the system using specific constraints*

$$x_i = f(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

Post-processing: $x^2 = -2y$

$$
C_T : \left\{ \begin{array}{lcccccccccc}
c_1^T : & -t & -x & -xy & & -t^2 & & & & +2y & = & 0 \\
c_2^T : & & -x & -xy & -y^2 & & & & & +y & = & 0 \\
c_3^{T'} : & & & & & & z^2 & +z & & -\ y & = & 0 \\
c_4^T : & & & & & & & & +x^2 & +2y & = & 0
\end{array} \right.
$$

Solving stage: 4 *solutions reached in* 10*ms!*

Approaches (2)

Efficiency of domain contraction:

- Redundant constraints: *"the more the better"*
- Linearization: *as the solving process goes, eliminate non-linear terms*
- Triangularization: *Gaussian-like elimination*

- Common Sub-Expressions technique

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Applications

- Chemistry
- Aeronautics
- Software verification
- Bio-medical engineering
- Geosciences
- Car industry
- Security
- Grid computing
- ...

Applications

- Chemistry
- Aeronautics
- Software verification
- Bio-medical engineering
- Geosciences
- Car industry
- Security
- Grid computing
- ...

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

- How to model this problem as a constraint problem?

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

- How to model this problem as a constraint problem?
  1. *Transform both the specification S and program P as constraint problems*

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

- How to model this problem as a constraint problem?
  1. *Transform both the specification S and program P as constraint problems*
  2. *Solve* $S \wedge \neg P$

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

- How to model this problem as a constraint problem?
  1. *Transform both the specification S and program P as constraint problems*
  2. *Solve* $S \wedge \neg P$
  3. *If there is no solution: the program is verified*

Software verification

- What is the general problem?
  Verifying programs: check that they are consistent with their specifications
  and/or generating test cases

- How to model this problem as a constraint problem?
  1. *Transform both the specification S and program P as constraint problems*
  2. *Solve* $S \wedge \neg P$
  3. *If there is no solution: the program is verified*
  4. *Otherwise we have test cases for counter-examples*

Bio-medical engineering: Gait Analysis

- What is the general problem?

  Diagnosing patients' gait

  Long-term objective: automating and guiding the therapy

  = an intelligent system for gait therapy

Bio-medical engineering: Gait Analysis

- What is the general problem?
  Diagnosing patients' gait
  Long-term objective: automating and guiding the therapy
  = an intelligent system for gait therapy

- What do we know about the gait?
  Gait measurements: through markers placed on patients' joints

Bio-medical engineering: Gait Analysis

- What is the general problem?

  Diagnosing patients' gait

  Long-term objective: automating and guiding the therapy

  = an intelligent system for gait therapy

- What do we know about the gait?

  Gait measurements: through markers placed on patients' joints

- How to model this problem as a constraint problem?

Bio-medical engineering: Gait Analysis

- What is the general problem?
  Diagnosing patients' gait
  Long-term objective: automating and guiding the therapy
  = an intelligent system for gait therapy

- What do we know about the gait?
  Gait measurements: through markers placed on patients' joints

- How to model this problem as a constraint problem?

  1. *It can be a* <u>*parameter estimation problem*</u>*: but need a single model*

Bio-medical engineering: Gait Analysis

- What is the general problem?
  Diagnosing patients' gait
  Long-term objective: automating and guiding the therapy
  = an intelligent system for gait therapy

- What do we know about the gait?
  Gait measurements: through markers placed on patients' joints

- How to model this problem as a constraint problem?
  1. *It can be a <u>parameter estimation problem</u>: but need a single model*
  2. *It can be translated into a <u>pattern constraint problem</u>: qualitative approach*

Geosciences

- What is the problem?
  Finding the structure of the earth in a given region

Geosciences

- What is the problem?
  Finding the structure of the earth in a given region

- What data is at our disposal to determine it?
  Measurements of a signal travel time
  + a model whose parameters describe the structure of the earth

Geosciences

- What is the problem?
  Finding the structure of the earth in a given region

- What data is at our disposal to determine it?
  Measurements of a signal travel time
  + a model whose parameters describe the structure of the earth

- How to model it as constraints?
  Parameter estimation

Security in Computer Networks

- What is the problem?
  - *Determining whether there is a risk of information leakage in a MLS computer network*
  - *If necessary, make the network safe*

Security in Computer Networks

- What is the problem?
  - *Determining whether there is a risk of information leakage in a MLS computer network*
  - *If necessary, make the network safe*
- How to model it as constraints?
  - *Variables = network connections*
  - *Constraints = no invalid connection + no risk of leakage*

Security in Computer Networks

- What is the problem?
  - *Determining whether there is a risk of information leakage in a MLS computer network*
  - *If necessary, make the network safe*
- How to model it as constraints?
  - *Variables = network connections*
  - *Constraints = no invalid connection + no risk of leakage*
- Limitations...
  - *The odds are that there will be no “satisfactory” solution*
  - *The constraints are too strong: modeling a “too secure” network*

Security in Computer Networks

- What is the problem?
  - *Determining whether there is a risk of information leakage in a MLS computer network*
  - *If necessary, make the network safe*
- How to model it as constraints?
  - *Variables = network connections*
  - *Constraints = no invalid connection + no risk of leakage*
- Soft constraints?
  - *Keep the validity constraints: they model the MLS policy*
  - *Relax the risk constraints: aim at the minimum risk*
  - *Add the objective that the cuts have to be “minimum”: to avoid ending up with a network without connections*

Security in Computer Networks

- What is the problem?
  - *Determining whether there is a risk of information leakage in a MLS computer network*
  - *If necessary, make the network safe*
- How to model it as constraints?
  - *Variables = network connections*
  - *Constraints = no invalid connection + no risk of leakage*
- Soft constraints?
  - *Keep the validity constraints: they model the MLS policy*
  - *Relax the risk constraints: aim at the minimum risk*
  - *Add the objective that the cuts have to be "minimum": to avoid ending up with a network without connections*

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Outline of the presentation

- Overview of Symbolic-Numeric Algorithms
- Challenges
- Applications
- Future directions?

Future directions of work? (1)

Existing challenges:

- Extension of global constraints
- Combining the current work on DAGs with triangularization approaches
- “New" kinds of interval arithmetic

  e.g., circular arithmetic of Siegfried Rump

Future directions of work? (2)

New challenges?

- Robustness of solutions
- Use of numeric tensors, n-d arrays to better represent the dependence between variables' “values"

  (discussions from NSF-sponsored workshop, CoProD'08)

  Similar to shaving to some extent (forward looking)

  Can bring improvement thanks to the availability of information (n-d)

Conclusion

- Symbolic-Numeric algorithms for constraint solving:
  - *Necessary due to the nature of the problem (use of intervals)*
  - *The combination occurs at different stages: e.g., pre-processing, completely integrated*

Conclusion

- Symbolic-Numeric algorithms for constraint solving:
  - *Necessary due to the nature of the problem (use of intervals)*
  - *The combination occurs at different stages: e.g., pre-processing, completely integrated*
- Many challenges still to be addressed, including:
  - *Pursuing existing research directions*
  - *Exploring new representations: such as tensors, circular arithmetic*

The end

Thank you for your attention

ANY QUESTIONS?

Martine Ceberio

mceberio@utep.edu

www.constraintsolving.com

www.martineceberio.fr

University of Texas at El Paso

How to solve continuous constraints?

- Enumeration is not an option...

How to solve continuous constraints?

- Enumeration is not an option...
- Algorithms based on intervals

  Intervals are enumerated and the whole search space is covered

How to solve continuous constraints?

- Enumeration is not an option...
- Algorithms based on intervals

Intervals are enumerated and the whole search space is covered

⋆ *Branch and Bound (B&B):*

http://www-sop.inria.fr/coprin/logiciels/ALIAS/Movie/movie_undergraduate.mpg

⋆ *More sophisticated consistency algorithms:*

Box / Hull-consistencies and their combinations

Result in Branch and Prune algorithms (B&P)

How is Branch and Prune different?

- Discarding boxes in no longer based on the constraints' evaluations only
- Consistency techniques are used to filter out / prune elements that do not satisfy the constraints

B& B vs. B& P

Consider $f(x,y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0,2]$.

B& B vs. B& P

Consider $f(x, y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0, 2]$.

- B& B: Evaluating f on $[0, 2]^2$ gives $[-1, 7]$ which contains 0

B& B vs. B& P

Consider $f(x, y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0, 2]$.

- B& B: Evaluating f on $[0, 2]^2$ gives $[-1, 7]$ which contains 0 $\rightsquigarrow$ we have to keep the whole domain and bisect it for further information

B& B vs. B& P

Consider $f(x, y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0, 2]$.

- B& B: Evaluating f on $[0, 2]^2$ gives $[-1, 7]$ which contains 0 $\rightsquigarrow$ we have to keep the whole domain and bisect it for further information
- B& P: Using a consistency technique can lead to:

B& B vs. B& P

Consider $f(x,y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0,2]$.

- B& B: Evaluating f on $[0,2]^2$ gives $[-1,7]$ which contains 0 $\rightsquigarrow$ we have to keep the whole domain and bisect it for further information
- B& P: Using a consistency technique can lead to:
  - $y = (1 - x^2)^{1/2} = (1 - [0,4])^{1/2} = [-1,1]$

B& B vs. B& P

Consider $f(x,y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0,2]$.

- B& B: Evaluating f on $[0,2]^2$ gives $[-1,7]$ which contains 0 $\rightsquigarrow$ we have to keep the whole domain and bisect it for further information
- B& P: Using a consistency technique can lead to:
  - $y = (1 - x^2)^{1/2} = (1 - [0,4])^{1/2} = [-1,1]$ $\rightsquigarrow$ since we know that $y \in [0,2]$, now we know that the consistent part of this domain is $[0,1]$

    We can do the same with x and narrow it scope to $[0,1]$

B& B vs. B& P

Consider $f(x, y) = x^2 + y^2 - 1 = 0$ where x and $y \in [0, 2]$.

- B& B: Evaluating f on $[0, 2]^2$ gives $[-1, 7]$ which contains 0 $\rightsquigarrow$ we have to keep the whole domain and bisect it for further information
- B& P: Using a consistency technique can lead to:
  - $y = (1 - x^2)^{1/2} = (1 - [0, 4])^{1/2} = [-1, 1]$ $\rightsquigarrow$ since we know that $y \in [0, 2]$, now we know that the consistent part of this domain is $[0, 1]$

    We can do the same with x and narrow it scope to $[0, 1]$
  - Then, we have to bisect if we want more information

    but the domain to bisect is much smaller (25%) than with B& B

What are the solutions like?

- Example from chemistry:

$$\begin{aligned}
14 * z_1^2 + 6 * z_1 * z_2 + 5 * z_1 - 72 * z_2^2 - 18 * z_2 &= 850 * z_3 - 2.0e - 9, \\
0.5 * z_1 * z_2^2 + 0.01 * z_1 * z_2 + 0.13 * z_2^2 + 0.04 * z_2 &= 4.0e4, \\
0.03 * z_1 * z_3 + 0.04 * z_3 &= 850
\end{aligned}$$

$$z_i \in [-1000; +1000]$$

What are the solutions like?

- Example from chemistry:

$$\begin{aligned} 14 * z_1^2 + 6 * z_1 * z_2 + 5 * z_1 - 72 * z_2^2 - 18 * z_2 &= 850 * z_3 - 2.0e - 9, \\ 0.5 * z_1 * z_2^2 + 0.01 * z_1 * z_2 + 0.13 * z_2^2 + 0.04 * z_2 &= 4.0e4, \\ 0.03 * z_1 * z_3 + 0.04 * z_3 &= 850 \end{aligned}$$

$$z_i \in [-1000; +1000]$$

- Solutions:

OUTER BOX 1 $z_1 \in [124.7643488370932, 124.7643488370934]$

$z_2 \in [25.28546066708894, 25.28546066708898]$

$z_3 \in [224.6935300130224, 224.6935300130227]$

OUTER BOX 2 $z_1 \in [131.7475644308769, 131.7475644403561]$

$z_2 \in [-24.62787829918984, -24.62787829860991]$

$z_3 \in [212.9030823228049, 212.9030823233716]$

What are the solutions like?

- Solutions:

OUTER BOX 1	$z_1 \in [124.7643488370932, 124.7643488370934]$
	$z_2 \in [25.28546066708894, 25.28546066708898]$
	$z_3 \in [224.6935300130224, 224.6935300130227]$
OUTER BOX 2	$z_1 \in [131.7475644308769, 131.7475644403561]$
	$z_2 \in [-24.62787829918984, -24.62787829860991]$
	$z_3 \in [212.9030823228049, 212.9030823233716]$

- Comments?
  - Not accurate...

What are the solutions like?

- Solutions:

OUTER BOX 1 $z_1 \in [124.7643488370932, 124.7643488370934]$

$z_2 \in [25.28546066708894, 25.28546066708898]$

$z_3 \in [224.6935300130224, 224.6935300130227]$

OUTER BOX 2 $z_1 \in [131.7475644308769, 131.7475644403561]$

$z_2 \in [-24.62787829918984, -24.62787829860991]$

$z_3 \in [212.9030823228049, 212.9030823233716]$

- Comments?
  - Not accurate... Really?

What are the solutions like?

- Solutions:

OUTER BOX 1 $z_1 \in [124.7643488370932, 124.7643488370934]$

$z_2 \in [25.28546066708894, 25.28546066708898]$

$z_3 \in [224.6935300130224, 224.6935300130227]$

OUTER BOX 2 $z_1 \in [131.7475644308769, 131.7475644403561]$

$z_2 \in [-24.62787829918984, -24.62787829860991]$

$z_3 \in [212.9030823228049, 212.9030823233716]$

- Comments?
  - Not accurate... Really?
  - Noise

What are the solutions like?

- Solutions:

OUTER BOX 1
$$z_1 \in [124.7643488370932, 124.7643488370934]$$
$$z_2 \in [25.28546066708894, 25.28546066708898]$$
$$z_3 \in [224.6935300130224, 224.6935300130227]$$

OUTER BOX 2
$$z_1 \in [131.7475644308769, 131.7475644403561]$$
$$z_2 \in [-24.62787829918984, -24.62787829860991]$$
$$z_3 \in [212.9030823228049, 212.9030823233716]$$

- Comments?
  - Not accurate... Really?
  - Noise : part of it can be filtered as a post-processing step
  - Duplicates...

What are the solutions like?

- Solutions:

  OUTER BOX 1 $z_1 \in [124.7643488370932, 124.7643488370934]$

  $z_2 \in [25.28546066708894, 25.28546066708898]$

  $z_3 \in [224.6935300130224, 224.6935300130227]$

  OUTER BOX 2 $z_1 \in [131.7475644308769, 131.7475644403561]$

  $z_2 \in [-24.62787829918984, -24.62787829860991]$

  $z_3 \in [212.9030823228049, 212.9030823233716]$

- Comments?
  - Not accurate... Really?
  - Noise : part of it can be filtered as a post-processing step
  - Duplicates... This is a real problem

Constraint consistency techniques

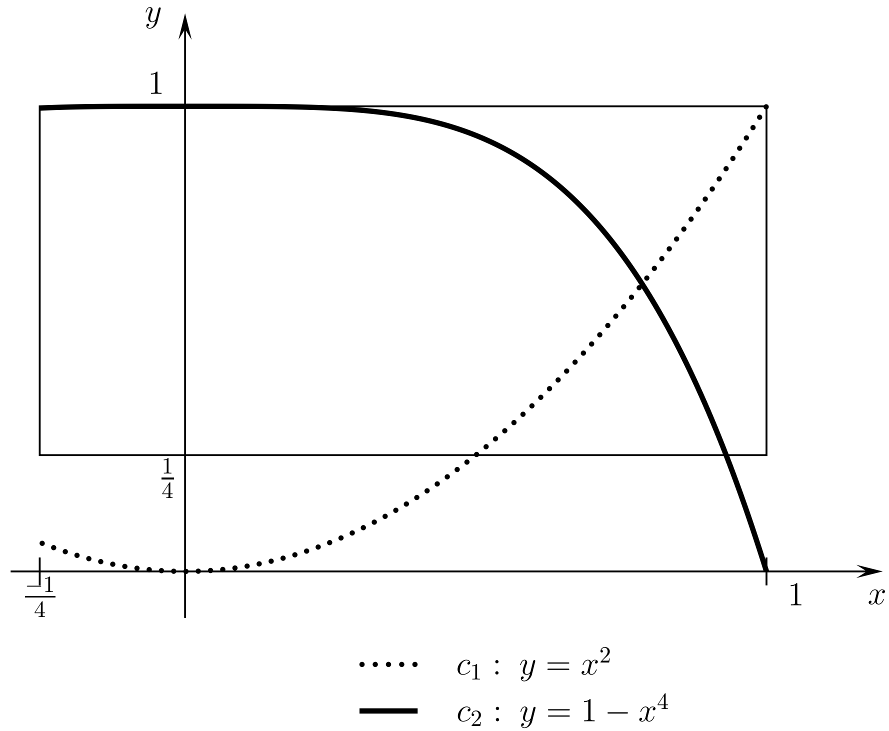
Most of them are local...

Constraint consistency techniques

Most of them are local... **Locality of Reasonings**

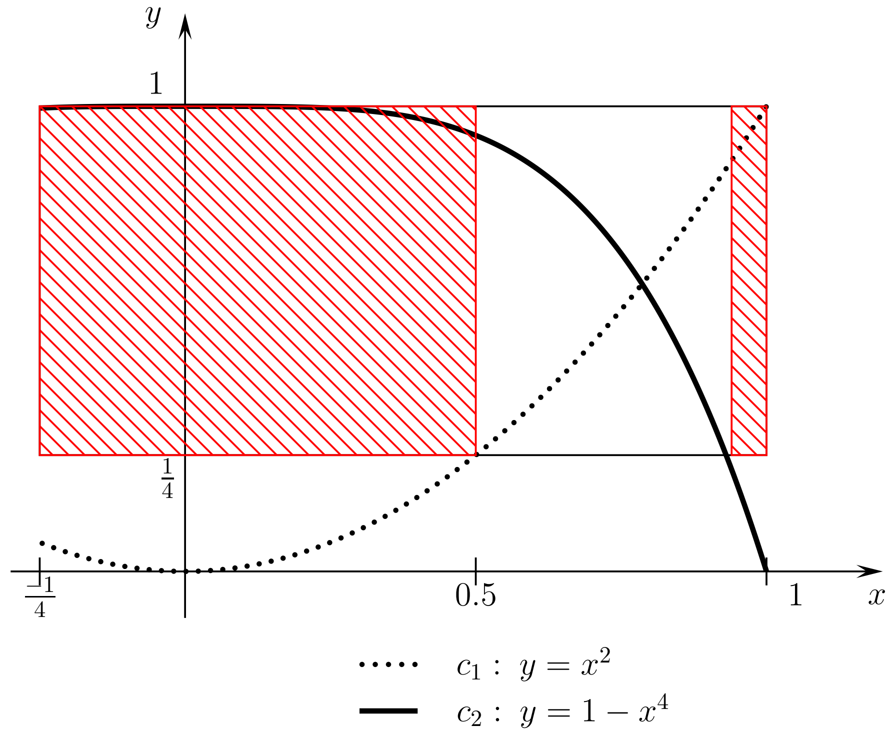
Constraint consistency techniques

Most of them are local... **Locality of Reasoning**



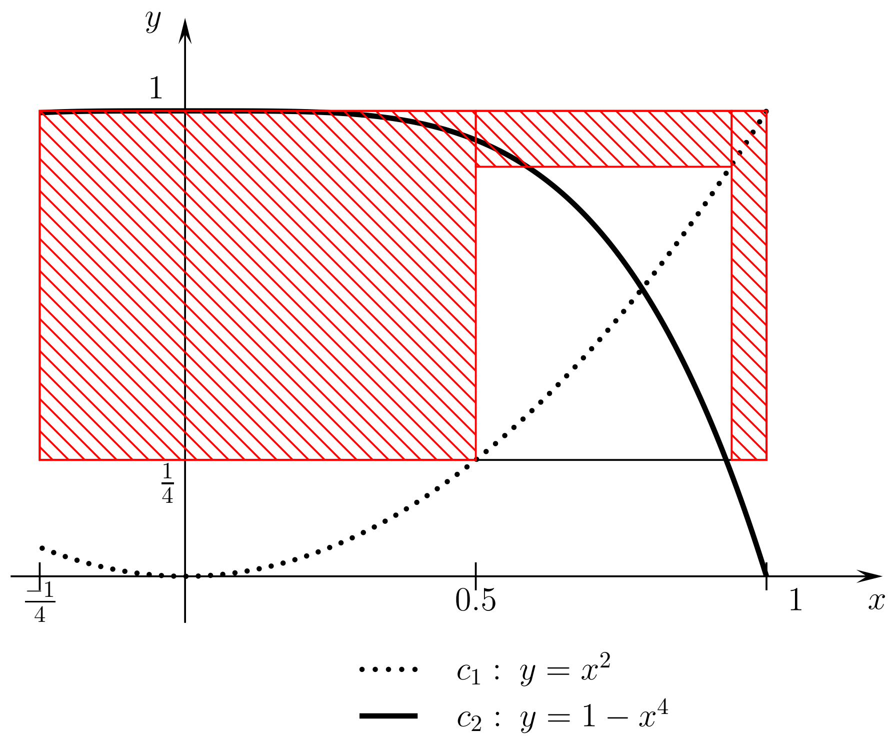
Constraint consistency techniques

Most of them are local... **Locality of Reasonings**



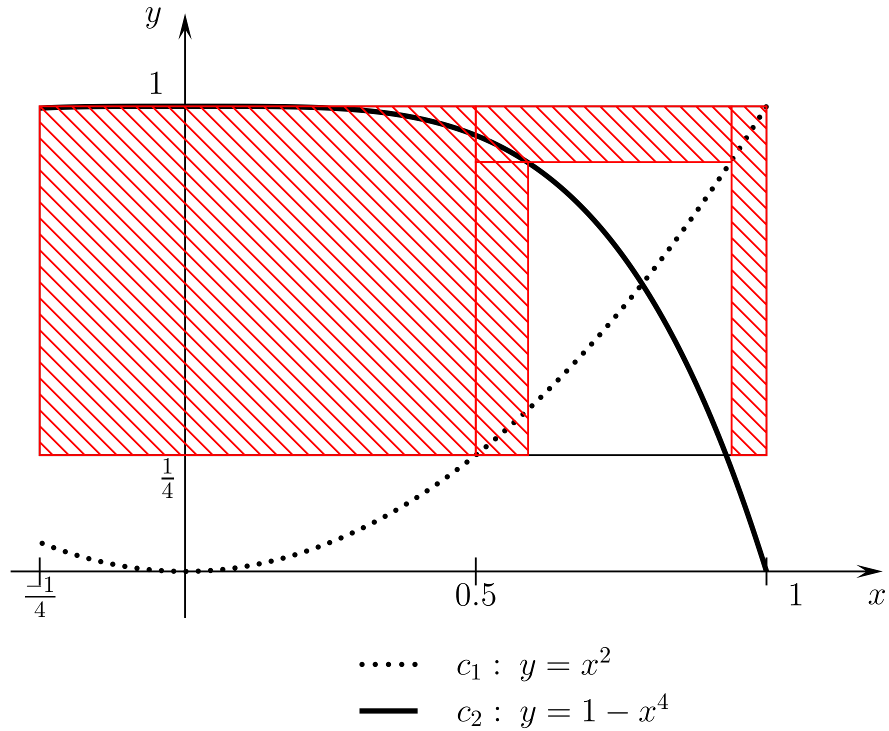
Constraint consistency techniques

Most of them are local... **Locality of Reasonings**



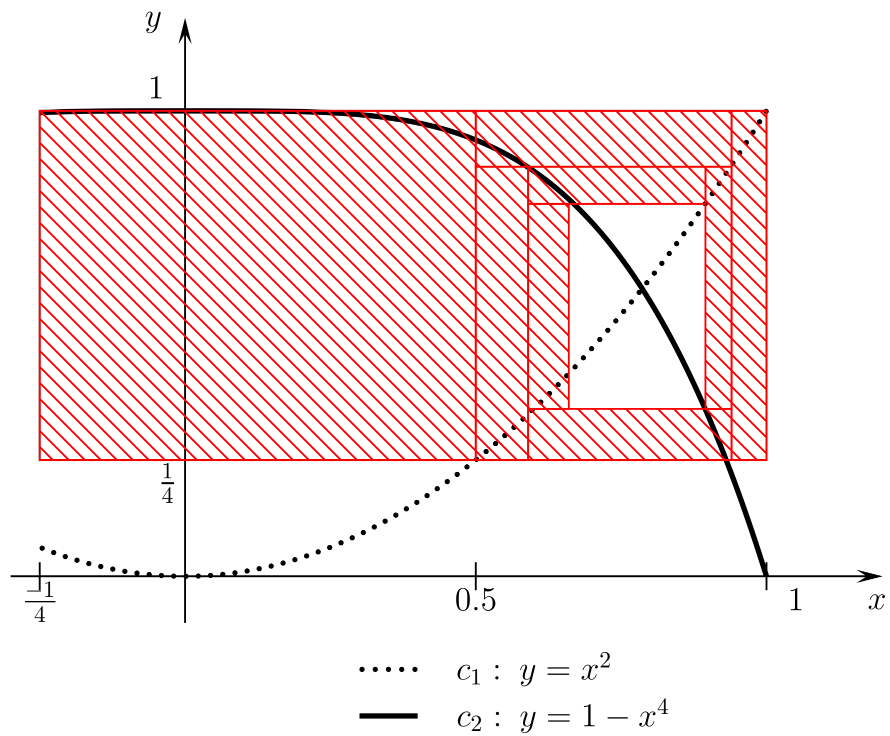
Constraint consistency techniques

Most of them are local... **Locality of Reasonings**



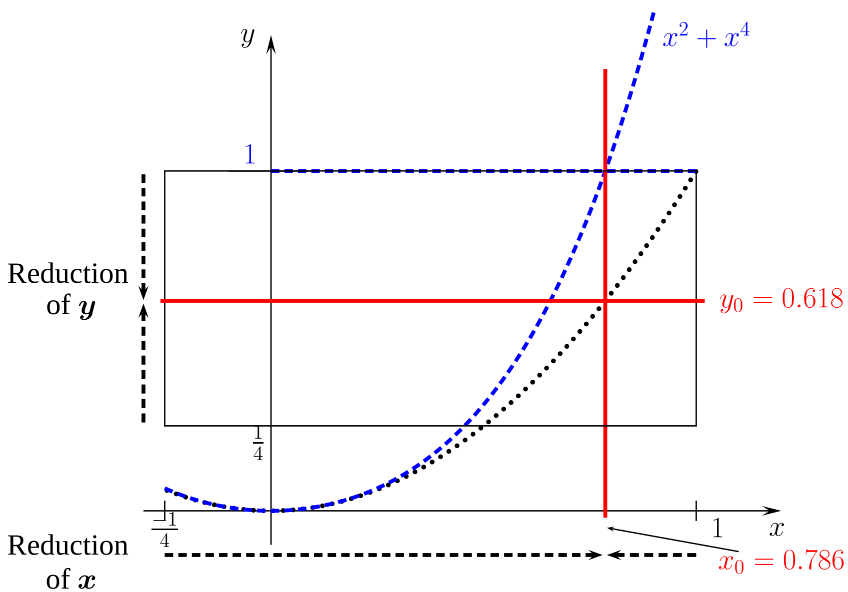
Constraint consistency techniques

Most of them are local... **Locality of Reasonings**



Constraint consistency techniques

Most of them are local... **Locality of Reasonings**



$$\boxed{\begin{array}{ll} c_1: & y = x^2 \\ c_2: & y = 1 - x^4 \end{array}}$$

$$\Downarrow$$

$$\boxed{\begin{array}{ll} c_1: & y = x^2 \\ c_2': & x^2 = 1 - x^4 \end{array}}$$

Some results

Problem	v	Initial Pb.		Triangul. Pb.	
		Time	Sol.	Time	Sol.
Bratu	7	1.10	3	0.60	4
	8	0.70	2	0.10	2
	10	2.30	2	0.10	2
	13	20.50	6	0.10	2
	14	46.40	11	0.20	2
	15	94.40	12	0.20	2

Some results

Problem	v	Initial Pb.		Triangul. Pb.	
		Time	Sol.	Time	Sol.
Bratu	7	1.10	3	0.60	4
	8	0.70	2	0.10	2
	10	2.30	2	0.10	2
	13	20.50	6	0.10	2
	14	46.40	11	0.20	2
	15	94.40	12	0.20	2

+ Recent work of Gilles Trombettoni et al. on combining these ideas within the consistency techniques.