Tractable Combinations of Global Constraints

CP 2013, pp 230—246

Authors: Cohen, Jeavons, Thornstensen, Zivny

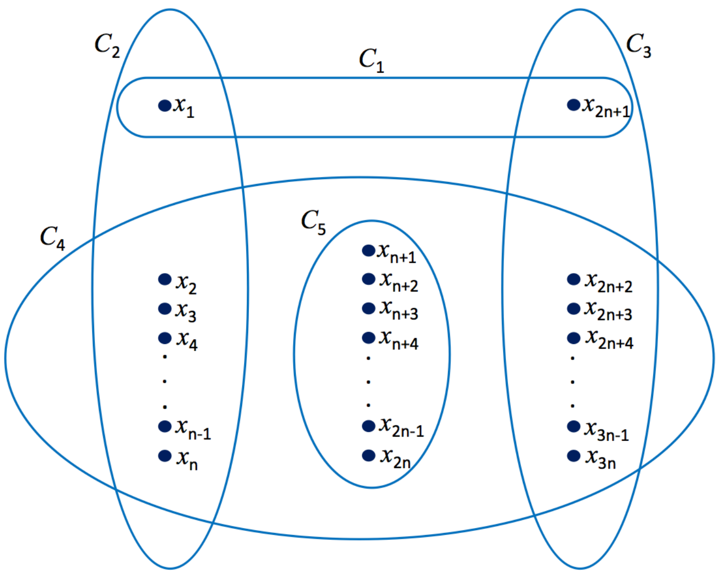
Presented by Robert Woodward

Disclaimer:

Some slides and images borrowed from the authors own slides at CP 2013

Main Contributions

- Addresses tractability (of intersection) of global constraints
- Identifies tractability conditions for arbitrary constraints
  1. Polynomial size of assignments of constraints intersections
  2. Bounded sizes of constraints
- Shows that property holds for constraints of
  1. Extended Global Cardinality (EGC) of bounded domains
  2. Positive Tables
  3. Negative Tables

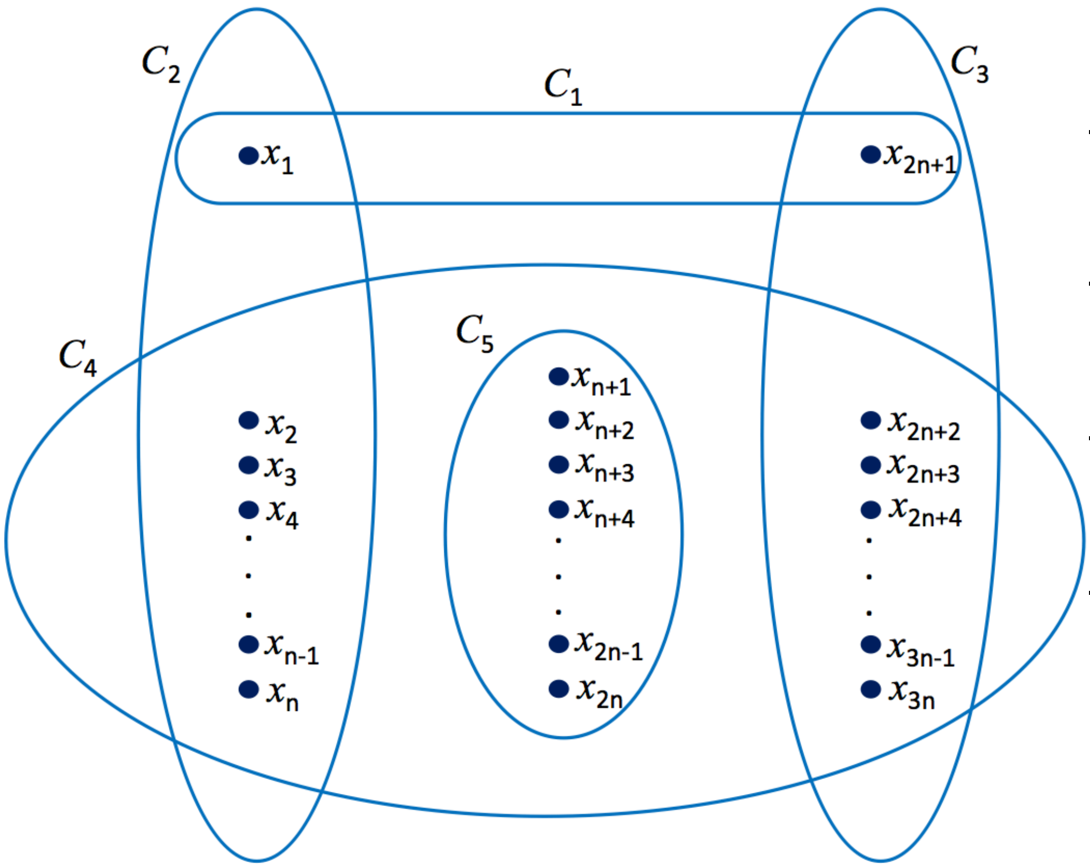


Overview

- Motivating Example
- Restricted Classes of CSPs
  - Acyclic hypergraph
  - Treewidth & constraint catalogue
- Further Constraint Restrictions
  - Extensional equivalence
  - Operations on sets of global constraints
  - Cooperating constraint catalogues
- Polynomial-Time Reductions
  - Take the dual of the dual
  - Tractability results
- Conclusion

Motivating Example

- Boolean vars: $\{x_1, x_2, \ldots, x_{3n}\}$

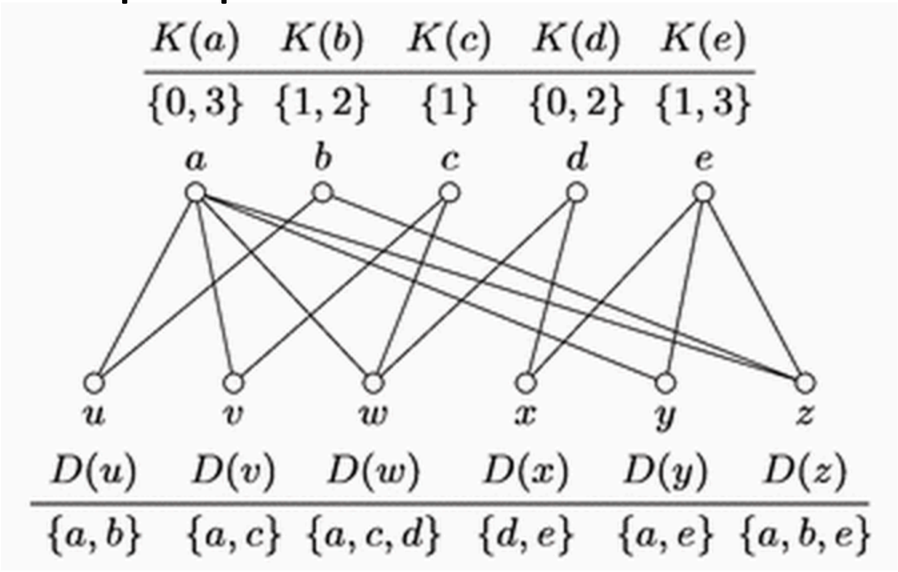


- 5 constraints:
  - C_1: $(x_1 \vee x_{2n+1})$
  - C_2: Exactly one literal is true
  - C_3: Exactly one literal is true
  - C_4: Exactly n+1 literals are true
  - C_5: $(\neg x_{n+1} \vee \neg x_{n+2} \vee \cdots \vee \neg x_{2n})$

Extended Global Cardinality Constraint

- For every domain element *a*
  - K(*a*) a finite set of natural numbers
  - Cardinality set of *a*
- Requires number of variables assigned to *a* to be in the set K(*a*)

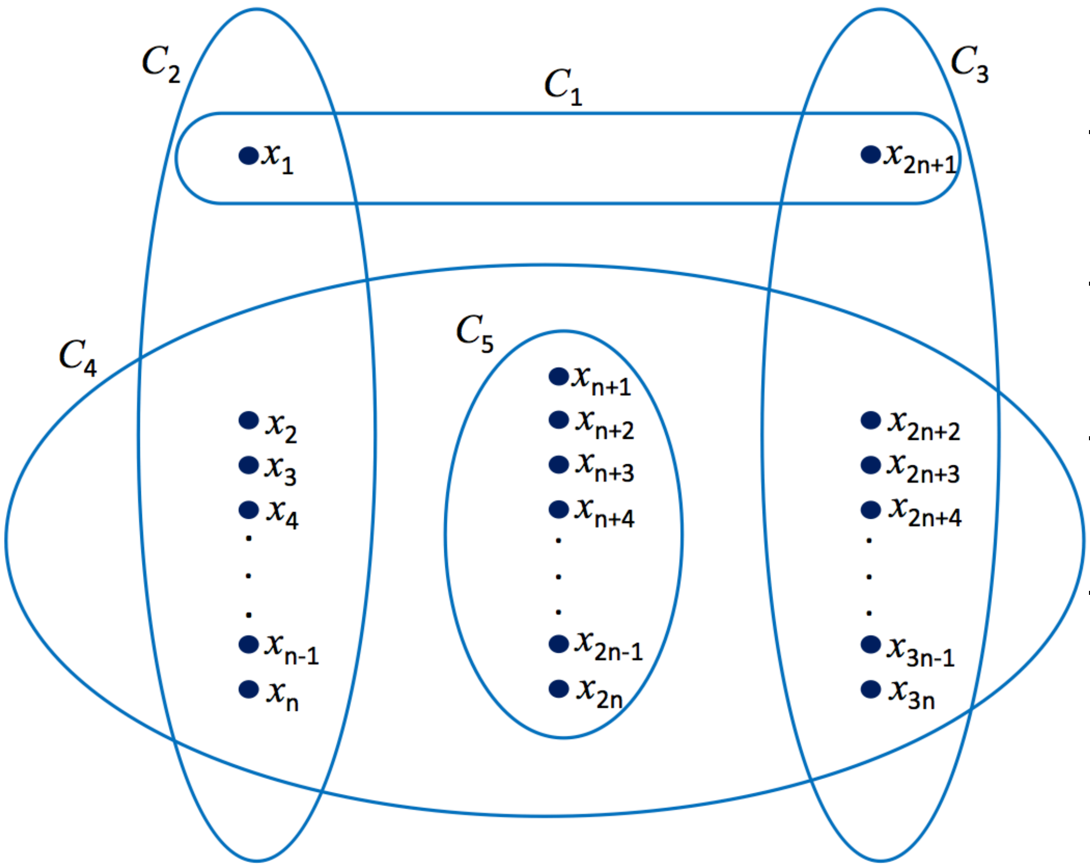
- Example: Timetabling
  - 6 workers {u,v,w,x,y,z}
  - 5 tasks {a,b,c,d,e}
  - Restrictions on how many people have to work on a task



Example from [Samer+ Constraints 11]

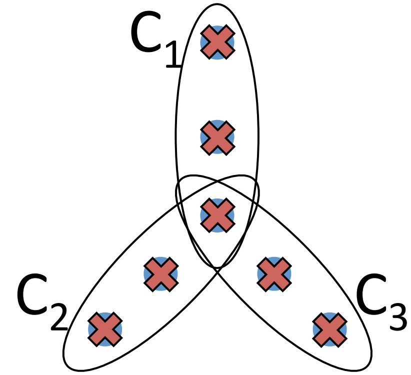
Motivating Example

All constraints are instances of Extended Global Cardinality (EGC) constraints

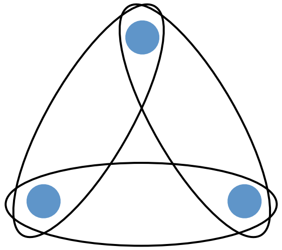


- C_1: $(x_1 \vee x_{2n+1})$
  $K(1)=\{1,2\}$, $K(0)=\{0,1\}$
- C_2: Exactly one literal is true
  $K(1)=\{1\}$, $K(0)=\{n-1\}$
- C_3: Exactly one literal is true
  $K(1)=\{1\}$, $K(0)=\{n-1\}$
- C_4: Exactly n+1 literals are true
  $K(1)=\{n+1\}$, $K(0)=\{2n-3\}$
- C_5: $(\neg x_{n+1} \vee \neg x_{n+2} \vee \cdots \vee \neg x_{2n})$
  $K(1)=\{0,1,\ldots,n-1\}$, $K(0)=\{1,2,\ldots,n\}$

Restricted Classes of CSPs



Acyclic



Non-acyclic

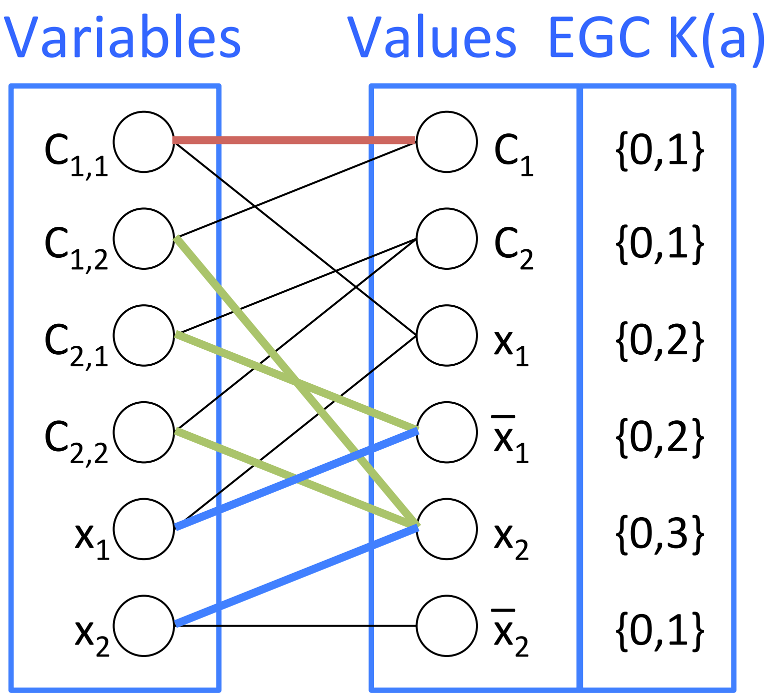
- Structural restrictions (e.g., treewidth)
- Hypergraph is acyclic when
  - Repeatedly removing
    - all hyperedges contained in other hyperedges, and
    - all vertices contained in only a single hyperedge
  - Eventually deletes all vertices
- Acyclic hypergraph
  - Tractable for table constraints
- Alert
  - Hypergraph of a global constraint has a single edge, is acyclic
  - However, not every global constraint is tractable
  - Two examples: An EGC constraint with unbounded & bounded domains

Example (I): EGC constraint with unbounded domain

- EGC constraint with unbounded domain is NP-complete
  - Reduction from SAT
  - Example: $(x_1 \vee x_2) \wedge (\overline{x}_1 \vee x_2)$

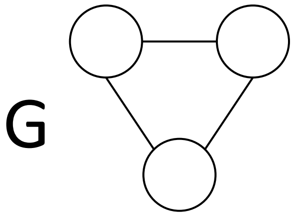
    $C_{1,1}$ $C_{1,2}$ $C_{2,1}$ $C_{2,2}$

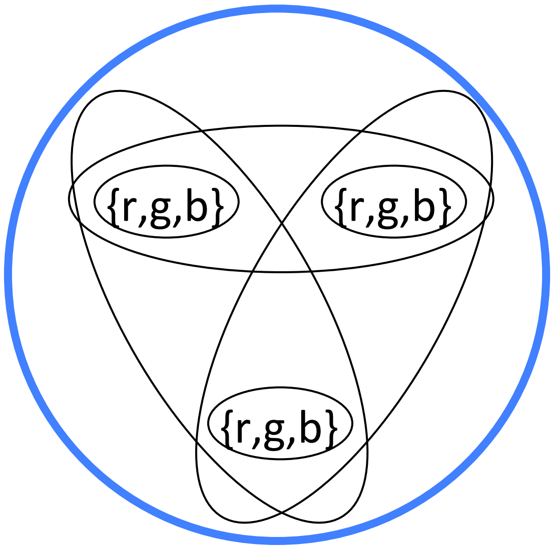
    Consider assignment:
    x_1 = false
    x_2 = true
  - Full proof in [Quimper+ CP04]



Example (II): EGC constraint with bounded domain

- EGC constraint with bounded domain is NP-complete
  - Reduction from 3-coloring G=(V,E)
- CSP:
  - V set of variables
  - Domains {r,g,b}
- For every edge create EGC constraint
  - K(r)=K(g)=K(b)={0,1}
- Make hypergraph acyclic
  - EGC constraint with scope V and
  - K'(r)=K'(g)=K'(b)={0,…,|V|}



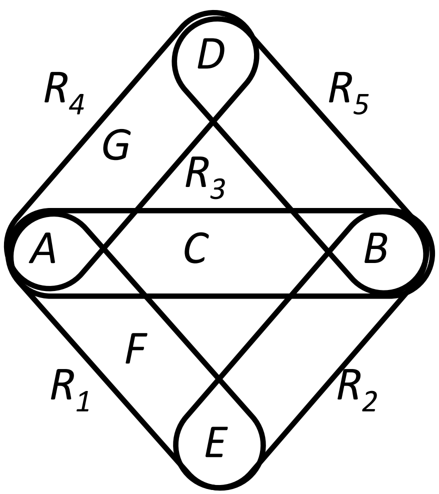


Review of Acyclic Hypergraphs

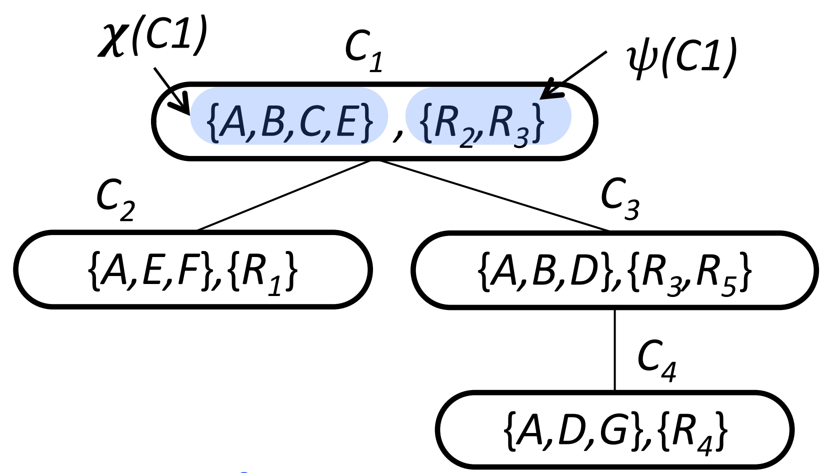
- Guarantee tractability for table constraints
- Do not guarantee tractability for global constraints
- Structural restrictions alone do not guarantee tractability in general!
- Need hybrid restrictions that restrict both
  – structure &
  – nature of the constraints

Tree Decomposition

- A tree decomposition: $\langle T, \chi, \psi \rangle$
  - T: a tree of clusters
  - χ: maps variables to clusters
  - ψ: maps constraints to clusters
- Conditions
  - Each constraint appears in at least one cluster with all the variables in the constraint's scope
  - For every variable, the clusters where the variable appears induce a connected subtree



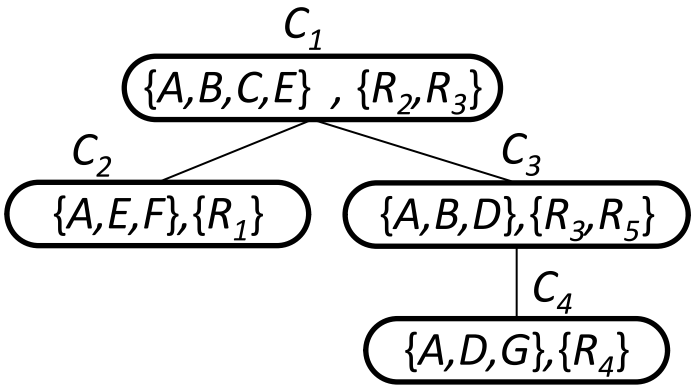
Hypergraph



Tree decomposition

Treewidth

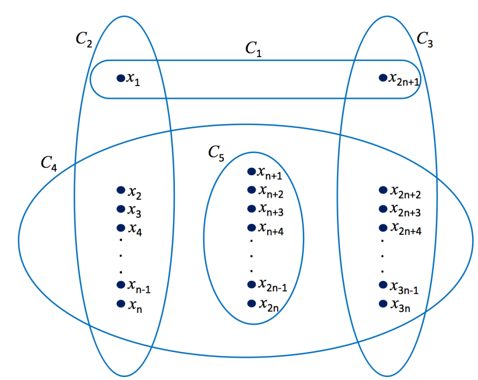
- Width of a tree decomposition
  - max({|χ(t)|-1 | t node of T})
- Treewidth tw(G) of a hypergraph G
  - minimum width over all its tree decompositions
- Great, but we are not interested in individual hypergraphs
  - $\mathcal{H}$ = class of hypergraphs
  - tw($\mathcal{H}$) = maximum treewidth over the hypergraphs in $\mathcal{H}$
- If tw($\mathcal{H}$) is unbounded, tw($\mathcal{H}$)=∞
- Otherwise tw($\mathcal{H}$)<∞
- Recall, I said we wanted to restrict both structure & constraints



Width of tree decomposition: 3

Constraint Catalogue

- Constraint catalogue $\mathcal{C}$ is a set of global constraints
- CSP instance is over a constraint catalog if every constraint in the instance is in the catalog
- Restricted CSP class
  - $\mathcal{C}$ a constraint catalog
  - $\mathcal{H}$ be a class of hypergraphs
  - CSP($\mathcal{H}$,$\mathcal{C}$) the class of CSP instances over $\mathcal{C}$ whose hypergraphs are in $\mathcal{H}$
- CSP($\mathcal{H}$,$\mathcal{C}$) is tractable if tw($\mathcal{H}$)<∞
- Does not help us with our example
  - tw($\mathcal{H}$)=∞



Overview

- Motivating Example
- Restricted Classes of CSPs
  - Acyclic hypergraph
  - Treewidth & constraint catalogue
- Further Constraint Restrictions
  - Extensional equivalence
  - Operations on sets of global constraints
  - Cooperating constraint catalogues
- Polynomial-Time Reductions
  - Take the dual of the dual
  - Tractability results
- Conclusion

Extension Equivalence

- Global constraint e[δ] to X⊆vars(δ)
- ext(μ ,e[δ])   [Selection & Projection]
  - Set of assignments of vars(δ)-X that extend μ to a satisfying assignment for e[δ]
- Two assignments θ_1, θ_2 to X
  - are extension equivalent on X w.r.t. e[δ]
  - if $ext(\theta_1,e[\delta])=ext(\theta_2,e[\delta])$
- Denoted equiv[e[δ],X]

A	B	C	
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
…	…	…	…

ext(A=0,B=1)

ext(A=1,B=0)

$$(\overset{X}{\boxed{A \vee B}} \vee C)$$

(A=0,B=1)&(A=1,B=0) are equivalent

Example: Extension Equivalence

- For any clause e[δ] & non-empty X⊆vars(δ)
- Any assignment to X will either
  - Satisfies at least one
    - Any extension will satisfy the clause
    - All such assignments are extension equivalent
  - Falsifies all of them
    - An extension will satisfy the clause iff it satisfies one of the other literals
- equiv[e[δ],X] has 2 equivalence classes

	A	B	C	
ext(A=0,B=0)	0	0	0	0
	0	0	1	1
ext(A=0,B=1)	0	1	0	1
	0	1	1	1
ext(A=1,B=0)	1	0	0	1
	1	0	1	1
	…	…	…	…

$$(\overset{X}{\boxed{A \vee B}} \vee C)$$

Operations on Sets of Global Constraints

- S a set of global constraints
- iv(S) = $\bigcap_{c \in S}$ vars(c)
  - Intersection of scopes of the constraints in S
- join(S) = a global constraint e'[δ']
  - Operates as you imagine a join should

Cooperating Constraint Catalogues

- Constraint catalogue $\mathcal{C}$ is cooperating if
  - For any finite set of global constraints $S \subseteq \mathcal{C}$
  - We can compute a set of assignments of the variables iv(S)
  - Containing at least one representative of each equivalence class of equiv[join(S),iv(S)]
  - In polynomial time in
    - the size of iv(S) and
    - the total size of the constraints in S

$$(A \vee B \vee C) \wedge (\bar{A} \vee B \vee D)$$

iv(S)

A	B	C	D	
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

ext(A=0,B=1)

ext(A=1,B=1)

Example: Cooperating Constraint Catalogue

- Constraint catalogue consisting entirely of clauses
- equiv[join(S),iv(S)] has at most |S|+1 classes
  - Similar argument to equiv[e[δ],X] has 2 equivalent classes
  - All other assignments that satisfy at least one literal in each clause (at most 1)
  - Single assignment of variables in iv(S) that falsify (at most |S|)
- Equivalence classes in equiv[join(S),iv(S)] increases linearly with S

$$(A \vee B \vee C) \wedge (\bar{A} \vee B \vee D)$$

iv(S)	A	B	C	D	
ext(A=0,B=0)	0	0	1	0	1
	0	0	1	1	1
	0	1	0	0	1
ext(A=0,B=1)	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	1
ext(A=1,B=0)	1	0	0	1	1
	1	0	1	1	1
	1	1	0	0	1
	1	1	0	1	1
ext(A=1,B=1)	1	1	1	0	1
	1	1	1	1	1

Are EGC Constraints Cooperating?

- In general, no
- Theorem

    Any constraint catalogue that contains only

    - counting constraints with bounded domain size,
    - table constraints, and
    - negative constraints,

    is a cooperating catalogue

- An EGC constraint is a counting constraint
    - Thus, it is tractable when it has bounded domain size
- Proof of theorem is not presented for lack of time

Overview

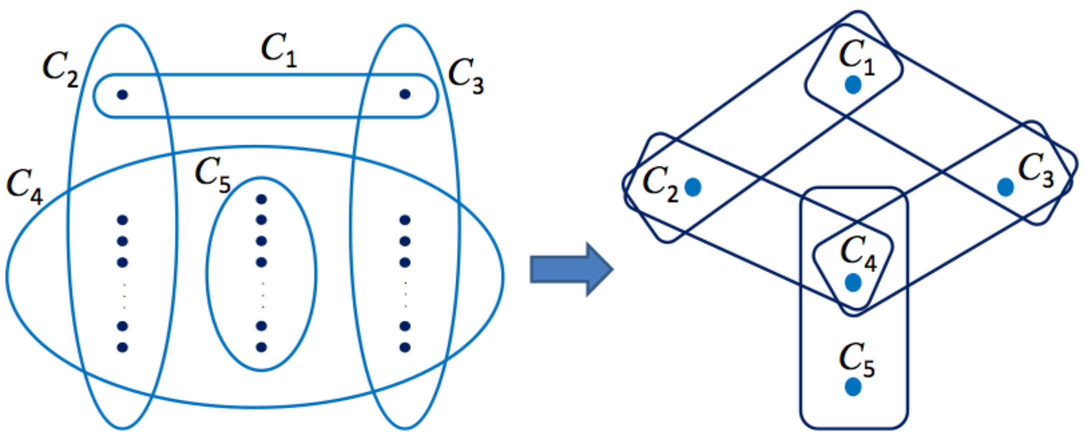
- Motivating Example
- Restricted Classes of CSPs
  - Acyclic hypergraph
  - Treewidth & constraint catalogue
- Further Constraint Restrictions
  - Extensional equivalence
  - Operations on sets of global constraints
  - Cooperating constraint catalogues
- Polynomial-Time Reductions
  - Take the dual of the dual
  - Tractability results
- Conclusion

Polynomial-Time Reductions

- Goal is to show for any constraint problem over a cooperating catalogue, give a polynomial-time reduction to a smaller problem
  - Consider a set of variables that all occur in exactly the same set of constraint scopes
  - Replace them by a single new variable with an appropriate domain
- How? Using the dual of a hypergraph

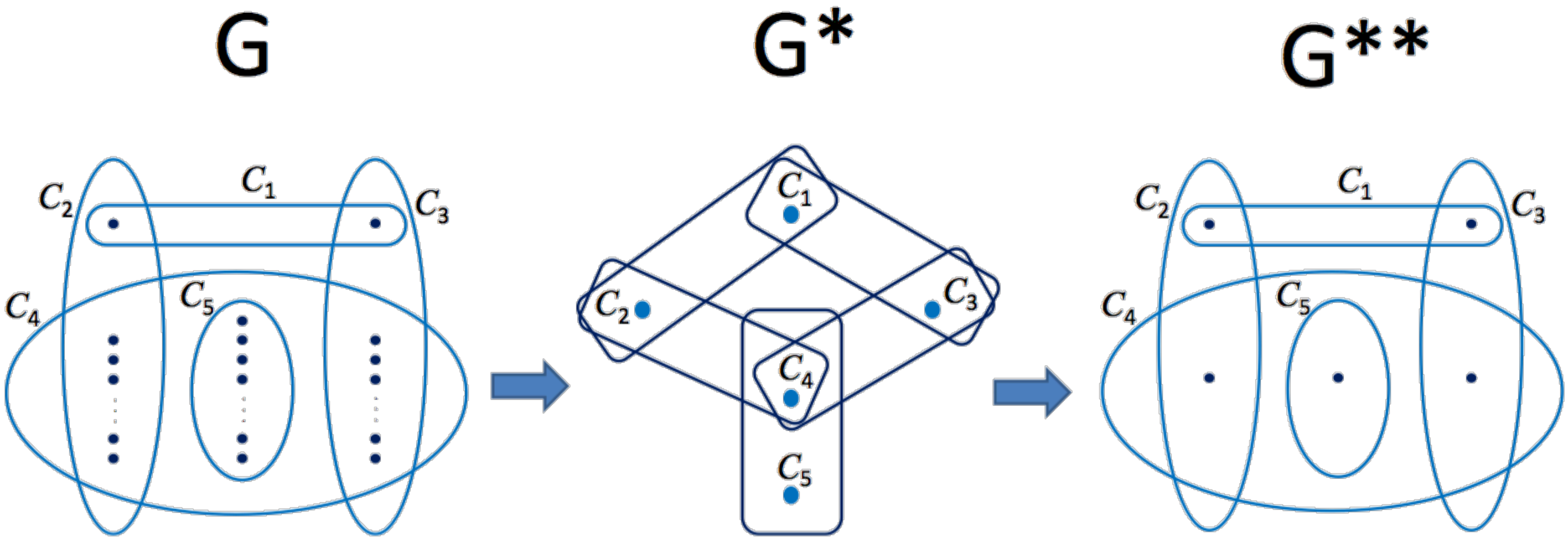
Dual of a Hypergraph

- G=(V,H) a hypergraph
- The dual of G, G* is a hypergraph with
  - Vertex set: H
  - For every v∈V, a hyperedge {h∈H|v∈h}



- For a class of hypergraphs $\mathscr{H}$, $\mathscr{H}$*={G*|G∈$\mathscr{H}$}

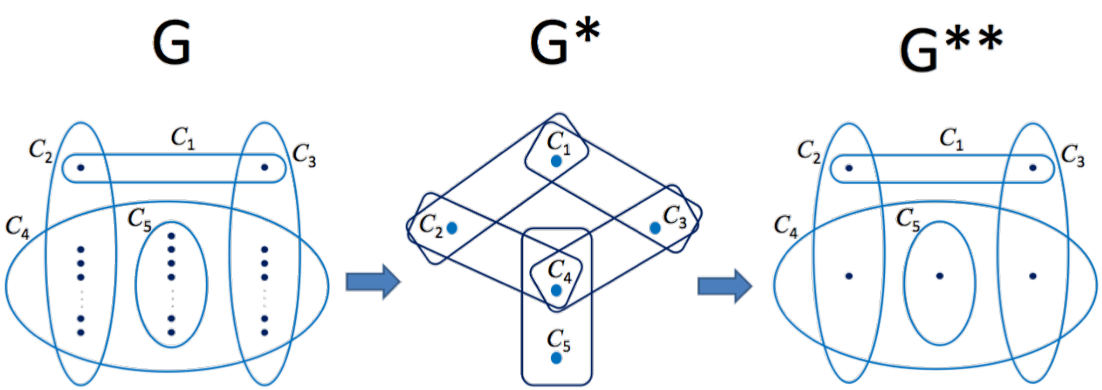
The Dual, in Pictures



Note: The dual of the dual of a hypergraph is not necessarily the original hypergraph

The Dual and Treewidth

- twDD: Treewidth of the dual of the dual of G
  - twDD(G)=tw(G**)
  - For class of hypergraphs $\mathscr{H}$, twDD($\mathscr{H}$)=tw($\mathscr{H}$**)
- For our example
  - twDD($\mathscr{H}$)=3



Tractability Result

- Constraint catalogue $\mathcal{C}$ and class of hypergraphs $\mathscr{H}$
- CSP($\mathscr{H}$,$\mathcal{C}$) is tractable if $\mathcal{C}$ is a cooperating catalogue and twDD($\mathscr{H}$)<∞
- I can sketch the definitions/ideas for the proof
  - The proof gives justification for why we can take the dual of the dual
  - See the paper for the full rigorous proof

Conclusions

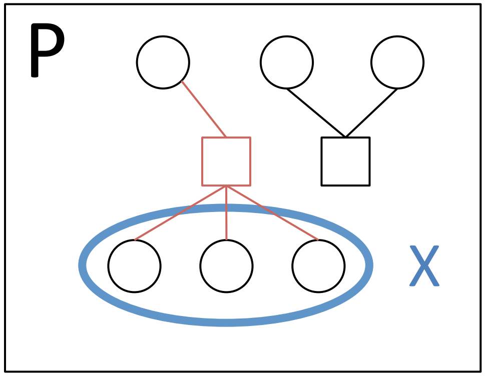
- Cannot achieve tractability by structural restrictions alone
- Introduce cooperating constraint catalogue
  - Sufficiently restricted to ensure that an individual constraint is always tractable
  - Not all structures are tractable even with cooperating constraint catalogue (twDD($\mathscr{H}$)=∞ NP-Complete)
- However, twDD($\mathscr{H}$)<∞ is tractable

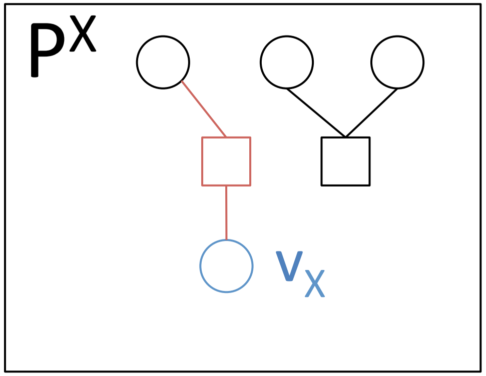
Thank You

- Any Questions?
  - ~~Don't ask me… I didn't write the paper~~
  - ~~Contact the authors ☺~~
  - Just kidding… I'll try to answer them!

Quotient of a CSP Instance

- Let P=(V,C) be a CSP instance
- $X \subseteq V$ non-empty subset of variables
  - all occur in the scope of the same set S of constraints
- The quotient of P w.r.t. X, P^X, defined:
  - Variables of P^X are given by $V^X=(V-X) \cup \{v_X\}$
    - v_X is a fresh variable
    - Domain of v_X is equiv[join(S),X]
  - Constraints of P^X are unchanged, except
    - each constraint $e[\delta] \in S$ is replaced by a new constraint $e^X[\delta^X]$
      - $vars(\delta^X)=(vars(\delta)-X) \cup \{v_X\}$
    - assignment θ true iff the equiv[join(S),X] is true





Using the Dual of the Dual

- CSP P can be converted to P'
  - With hyp(P')=hyp(P)**
  - Such that P' has a solution iff P does
  - If P is over a cooperating catalogue, this conversion can be done in polynomial time
- CSP($\mathscr{H}$,$\mathcal{C}$) is tractable if $\mathcal{C}$ is a cooperating catalogue and twDD($\mathscr{H}$)$<\infty$