Chapter 5: Global Constraints(Sudoku)

Helmut Simonis

Cork Constraint Computation Centre Computer Science Department University College Cork Ireland

ECLiPSe ELearning Overview





Helmut Simonis

Global Constraints

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Licence

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. To view a copy of this license, visit http:

//creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.





Helmut Simonis

Global Constraints

Outline

- Problem
- Program
- Initial Propagation (Forward Checking)
- Improved Reasoning
- Search



Helmut Simonis

Global Constraints

3

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

What we want to introduce

- Global Constraints
 - Powerful modelling abstractions
 - Non-trivial propagation
- Consistency Levels
 - Tradeoff between speed and propagation
 - Characterisation of reasoning power
- Example: Alldifferent
 - 3 variants shown



Methodology

- Evaluation on Sudoku puzzle
- Comparing
 - Initial setup
 - Search
 - Performance
- Explaining reasoning inside constraint
- Link to general classification of global constraints



Helmut Simonis

Global Constraints

F

Problem

Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

Problem Definition

Sudoku

Fill in numbers from 1 to 9 so that each row, column and block contain each number exactly once

	_	1	2	3				1	2	3	1	2	ω	1	2	3	1	2	3	1	2	3	1	2	3
/	1	4	5	6	Ι.	Q		4	5	6	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
-	т.	7	8	9	ļ '	U	'	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9
1 :	2 3	1	2	3	1	2	3	Г	-		Г			1	2	3	1	2	3	1	2	3	1	2	3
4 5	5 6	4	5	6	4	5	6	ı	1		ı	7	'	4	5	6	4	5	6	4	5	6	4	5	6
7 8	3 9	7	8	9	7	8	9		1			1		7	8	9	7	8	9	7	8	9	7	8	9
1 3	2 3	1	2	3	1	2	3	1	2	3	Г			1	2	3	1	2	3		Ξ		Г		
4 !	5 6	4	5	6	4	5	6	4	5	6	ı	Я		4	5	6	4	5	6		っ			つ	
7 8	3 9	7	8	9	7	8	9	7	8	9		U	,	7	8	9	7	8	9		U	'		<u>_</u>	
1.3	2 3	1	2	3				1	2	3	1	2	З	Г			Г				Ξ		1	2	3
4 !	5 6	4		6		റ		4	5	6	4	5	6	١.	Я			ソ	1		5		4	5	6
7 8		7	8	9		$\overline{}$		7	8	9	7	8	9		<u> </u>		-	_			<u> </u>	'	7	8	9
1 :	2 3	Г	$\overline{}$		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3		$\overline{}$		1	2	3 6
			q		4		6	4	5	6	4	5	6	4	5	6	4	5	6	١.	Я		4	5	
7 8	3 9		J		7	8	9	7	8	9	7	8	9	7	8	9	7	8	9		U	'	7	8	9
1 :	2 3	Г	$\overline{}$					Г			1	2	3	1	2	3	Г			1	2	3	1	2	3
4 !			'3			7		ı	h		4	5	6	4	5	6	1	Q		4	5	6	4		
7 8	3 9		<u> </u>			_			V		7	8	9	7	8	9		<u> </u>	_	7	8	9	7		9
		П			1	2	3	1	2	3	Г	Ξ		1	2	3	1	2	3	1	2	3	1	2	3
1.5)		7		4	5	6	4	5	6	ı	5		4 7	5	6	4	5	6	4	5	6	4		6
	_				7	8	9	7	8	9		V	'	7	8	9	7	8	9	7	8	9	7		9
1 :	2 3	1	2	3	1	2	3	1	2	3	Г	4			-		1	2	3	1	2	3	1	2	3
4 5	5 6	4	5	6	4	5	6	4	5	6	ı	1		١,	4	.	4	5	6	4	5	6	4	5	6
7 8	9	7	8	9	7	8	9	7	8	9	L			L	_		7	8	9	7		9	7	8	9
1.3	2 3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3		_		1	5	3		4	
4 !	5 6	4	5	6	4		6		5	6	4	5	6	4	5	6		h		4	5	6	١.	4	
7 8	3 9	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9		V		7	8	9		т	

4	2	8	5	6	3	1	7	9
3	5	တ	$ \leftarrow $	7	2	4	60	80
7	6	~	4	8	9	5	ന	2
1	4	6	ദ	9	8	2	5	7
5	9	2	7	4	1	3	8	6
8	ര	7	60	2	5	9	4	1
2	7	4	တ	5	60	8	$\overline{}$	3
6	8	3	2	1	4	7	9	5
9	1	5	8	3	7	6	2	4



Model

- A variable for each cell, ranging from 1 to 9
- A 9x9 matrix of variables describing the problem
- Preassigned integers for the given hints
- alldifferent constraints for each row, column and 3x3 block



Helmut Simonis

Global Constraints

_

Problem

Program Initial Propagation (Forward Checking) Improved Reasoning

Reminder: alldifferent

- Argument: list of variables
- Meaning: variables are pairwise different
- Reasoning: Forward Checking (FC)
 - When variable is assigned to value, remove the value from all other variables
 - If a variable has only one possible value, then it is assigned
 - If a variable has no possible values, then the constraint fails
 - Constraint is checked whenever one of its variables is assigned
 - Equivalent to decomposition into binary disequality constraints



Declarations

```
:-module(sudoku).
:-export(top/0).
:-lib(ic).

top:-
    problem(Matrix),
    model(Matrix),
    writeln(Matrix).
```



Helmut Simonis

Global Constraints

9

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

Data



Main Program

```
model(Matrix):-
    Matrix[1..9,1..9] :: 1..9,
    (for(I,1,9),
        param(Matrix) do
        alldifferent(Matrix[I,1..9]),
        alldifferent(Matrix[1..9,I])
),
    (multifor([I,J],[1,1],[7,7],[3,3]),
        param(Matrix) do
        alldifferent(flatten(Matrix[I..I+2,J..J+2]))
),
    flatten_array(Matrix,List),
    labeling(List).
```

Helmut Simonis

Global Constraints

11

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

Domain Visualizer

- Problem shown as matrix
- Each cell corresponds to a variable
- Instantiated: Shows integer value (large)
- Uninstantiated: Shows values in domain

	_	_	_			_		
	1 2 3	_	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
1 /1	4 5 6	2	4 5 6	4 5 6	4 5 6	4 5 6	4 5 6	4 5 6
	7 8 9	O	7 8 9	7 8 9	7 8 9	7 8 9	7 8 9	7 8 9
1 2 3	1 2 3	1 2 3			1 2 3	1 2 3	1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	1	7	4 5 6	4 5 6	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		l /	7 8 9	7 8 9	7 8 9	7 8 9
1 2 3			4 0 0	-			109	1 0 9
	1 2 3		1 2 3	0			0	0
4 5 6	4 5 6	4 5 6	4 5 6	1 X	4 5 6	4 5 6	.3	
7 8 9	7 8 9	7 8 9	7 8 9	0	7 8 9	7 8 9)	_
1 2 3	1 2 3		1 2 3	1 2 3			_	1 2 3
4 5 6	4 5 6	6	4 5 6	4 5 6	2	1	5	4 5 6
7 8 9	7 8 9	O	7 8 9	7 8 9	O	_	9	7 8 9
1 2 3		1 2 3	1 2 3	1 2 3	1 2 3	1 2 3		1 2 3
4 5 6	\circ	4 5 6	4 5 6	4 5 6	4 5 6	4 5 6	0	4 5 6
7 8 9	\exists	7 8 9	7 8 9	7 8 9	7 8 9	7 8 9	0	7 8 9
1 2 3	_	7 0 0	, , ,	1 2 3	1 2 3	100	1 2 3	1 2 3
	0	7	\sim			\sim	1 2 3	4 5 6
	.5	/	n		4 5 6	9	4 5 6	
7 8 9)	-)	7 8 9	7 8 9)	7 8 9	7 8 9
_	-	1 2 3	1 2 3	_	1 2 3	1 2 3	1 2 3	1 2 3
レン	/	4 5 6	4 5 6	5	4 5 6	4 5 6	4 5 6	4 5 6
_	- 1	7 8 9	7 8 9)	7 8 9	7 8 9	7 8 9	7 8 9
1 2 3	1 2 3	1 2 3	1 2 3			1 2 3	1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	4 5 6	I 1	1	4 5 6	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9	7 8 9		4	7 8 9	7 8 9	7 8 9
1 2 3	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3		1 2 3	
4 5 6	4 5 6	4 5 6	4 5 6	1 2 3		6	4 5 6	1
	14 3 0	14 3 6	- 3 0	1 3 0		Ö		4
7 8 9	789	789	789	789	7 8 9	_	7 8 9	



Constraint Visualizer

- Problem shown as matrix
- Currently active constraint highlighted
- Values removed at this step shown in blue
- Values assigned at this step shown in red

4	1 2 5 6	8	2 3 5 9	2 3 6 9	2 3 5 6 9	1 5 7	1 6 7 9	1 5 6 7 9
3 6 9	2 5 6	5 9	1	7	2 3 5 6 9	4 5 8	4 6 9	5 6 8 9
6 7 9	5 6	1 5 9	4 5 9	8	5 6 9	1 4 5 7	3	2
1	4	6	7 9	3	8	2	5	3 7 9
5	9	2	3 4 7	4	1 3	1 3 4 7	8	1 3 6 7
8	3	7	6	2	5	9	4	1
2	7	1 3 4 9	3 4 8 9	5	1 3 6 9	1 3 4 8	1 4 6 9	1 3 6 8 9
3 6 7 9	2 5 6 8	5 9	2 3 5 7 8 9	1	4	3 5 7 8	2 6 7 9	3 5 6 7 8 9



Helmut Simonis

Global Constraints

13

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Initial State (Forward Checking)

4	1 2 3	1 2 3 1 2 3	1 2 3 1 2 3	1 2 3 1 2 3
	4 5 6	4 5 6 4 5 6	4 5 6 4 5 6	4 5 6 4 5 6
	7 8 9	7 8 9 7 8 9	7 8 9 7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 7	1 2 3 1 2 3	1 2 3 1 2 3
4 5 6	4 5 6 4 5 6		4 5 6 4 5 6	4 5 6 4 5 6
7 8 9	7 8 9 7 8 9		7 8 9 7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3	1 2 3 1 2 3	3 2
4 5 6	4 5 6 4 5 6	4 5 6	4 5 6 4 5 6	
7 8 9	7 8 9 7 8 9	7 8 9	7 8 9 7 8 9	
1 2 3	1 2 3	1 2 3 1 2 3	8 2	5 1 2 3
4 5 6	4 5 6	4 5 6 4 5 6		4 5 6
7 8 9	7 8 9	7 8 9 7 8 9		7 8 9
1 2 3	9 1 2 3	1 2 3 1 2 3	1 2 3 1 2 3	8 1 2 3
4 5 6	4 5 6	4 5 6 4 5 6	4 5 6 4 5 6	4 5 6
7 8 9	7 8 9	7 8 9 7 8 9	7 8 9 7 8 9	7 8 9
1 2 3	3 7	6 1 2 3	1 2 3	1 2 3 1 2 3
4 5 6		4 5 6	4 5 6	4 5 6 4 5 6
7 8 9		7 8 9	7 8 9	7 8 9 7 8 9
2	7 1 2 3	1 2 3	1 2 3 1 2 3	1 2 3 1 2 3
	4 5 6	4 5 6	4 5 6 4 5 6	4 5 6 4 5 6
	7 8 9	7 8 9	7 8 9 7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3	4 1 2 3	1 2 3 1 2 3
4 5 6	4 5 6 4 5 6	4 5 6	4 5 6	4 5 6 4 5 6
7 8 9	7 8 9 7 8 9	7 8 9	7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3 1 2 3	1 2 3	1 2 3
4 5 6	4 5 6 4 5 6	4 5 6 4 5 6	4 5 6	4 5 6
7 8 9	7 8 9 7 8 9	7 8 9 7 8 9	7 8 9	7 8 9



Propagation Steps (Forward Checking)

4	1 2 5 6	8	2 3 5 9	2 3 6 9	2 3 5 6 9	1 5 7	1 6 7 9	1 5 6 7 9
3 6 9	2 5 6	3 5 9	1	7	2 3 5 6 9	4 5 8	4 6 9	5 6 8 9
6 7 9	1 5 6	1 5 9	4 5 9	8	5 6 9	1 4 5 7	3	2
1	4	6	3 7 9	3 9	8	2	5	379
5	9	2	3 4 7	4 6	1 3 6 7	1 3 4 7	8	1 3 6 7
8	3	7	6	2 4	1 2 5	9	1 2	5
2	7	1 3 4 9	3 4 8 9	5	1 3 6 9	1 3 4 8	1 4 6 9	1 3 6 8 9
3 6 7 9	2 5 6 8	3 5 9	2 3 5 7 8 9	1	4	5 7 8	2 6 7 9	3 5 6 7 8 9
3 7 9	1 2 5 8	1 3 5 9	2 3 5 7 8 9	2 3	1 2 3 5 7 9	6	 1 2 7 9 	4



Helmut Simonis

Global Constraints

15

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

After Setup (Forward Checking)

4	1 2 5 6	8	2 3 5 9	3 6 9	2 3 6 9	1 5 7	1 6 7 9	5 6 7 9
3 6 9	2 5 6	3 5 9	1	7	2 3 6 9	4 5 8	6 9	5 6 8 9
6 7 9	1 5 6	1 5 9	4 5 9	8	6 9	1 4 5 7	3	2
1	4	6	379	3	8	2	5	7
5	9	2	3 4 7	3	1 3	3 7	8	3 6 7
8	3	7	6	2	5	9	4	1
2	7	1 3 4 9	3 8 9	5	3 6 9	1 3	1 9	8 9
3 6 9	5 6 8	3 5 9	2 3 7 8 9	1	4	5 7 8	2 7 9	5 7 8 9
3	1 5 8	1 3 5 9	2 37 8 9	3	2 37 9	6	 1 2 7 9 	4



Can we do better?

- The alldifferent constraint is missing propagation
 - How can we do more propagation?
 - Do we know when we derive all possible information from the constraint?
- Constraints only interact by changing domains of variables



Helmut Simonis

Global Constraints

17

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

A Simpler Example

```
:-lib(ic).

top:-
    X :: 1..2,
    Y :: 1..2,
    Z :: 1..3,
    alldifferent([X,Y,Z]),
    writeln([X,Y,Z]).
```



Using Forward Checking

- No variable is assigned
- No reduction of domains
- But, values 1 and 2 can be removed from Z
- This means that Z is assigned to 3



Helmut Simonis

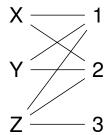
Global Constraints

19

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

Bounds Consistency Domain Consistency Comparison

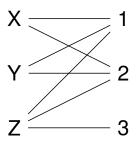
Visualization of alldifferent as Graph



- Show problem as graph with two types of nodes
 - Variables on the left
 - Values on the right
- If value is in domain of variable, show link between them
- This is called a bipartite graph



A Simpler Example



Value Graph for

X :: 1..2,

Y :: 1..2,

Z :: 1..3



Helmut Simonis

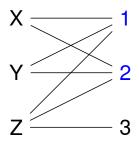
Global Constraints

2.

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

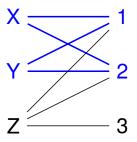
A Simpler Example



Check interval [1,2]



A Simpler Example



- Find variables completely contained in interval
- There are two: X and Y
- This uses up the capacity of the interval



Helmut Simonis

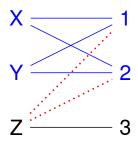
Global Constraints

23

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

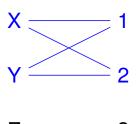
A Simpler Example



No other variable can use that interval



A Simpler Example



Only one value left in domain of Z, this can be assigned



Helmut Simonis

Global Constraints

25

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Idea (Hall Intervals)

- Take each interval of possible values, say size N
- Find all K variables whose domain is completely contained in interval
- If K > N then the constraint is infeasible
- If K = N then no other variable can use that interval
- Remove values from such variables if their bounds change
- If K < N do nothing
- Re-check whenever domain bounds change



Implementation

- Problem: Too many intervals $(O(n^2))$ to consider
- Solution:
 - Check only those intervals which update bounds
 - Enumerate intervals incrementally
 - Starting from lowest(highest) value
 - Using sorted list of variables
- Complexity: $O(n \log(n))$ in standard implementations
- Important: Only looks at min/max bounds of variables



Helmut Simonis

Global Constraints

27

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Bounds Consistency

Definition

A constraint achieves *bounds consistency*, if for the lower and upper bound of every variable, it is possible to find values for all other variables between their lower and upper bounds which satisfy the constraint.



Can we do better?

- Bounds consistency only considers min/max bounds
- Ignores "holes" in domain
- Sometimes we can improve propagation looking at those holes



Helmut Simonis

Global Constraints

29

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

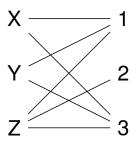
Bounds Consistency Domain Consistency Comparison

Another Simple Example

```
:-lib(ic).

top:-
    X :: [1,3],
    Y :: [1,3],
    Z :: 1..3,
    alldifferent([X,Y,Z]),
    writeln([X,Y,Z]).
```





Value Graph for

X :: [1,3],

Y :: [1,3],

Z :: 1..3



Helmut Simonis

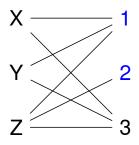
Global Constraints

31

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

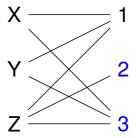
Bounds Consistency Domain Consistency Comparison

Another Simple Example



- Check interval [1,2]
- No domain of a variable completely contained in interval
- No propagation





- Check interval [2,3]
- No domain of a variable completely contained in interval
- No propagation



Helmut Simonis

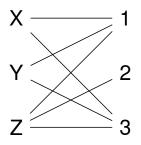
Global Constraints

33

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

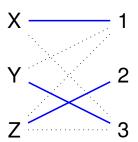
Bounds Consistency Domain Consistency Comparison

Another Simple Example



But, more propagation is possible, there are only two solutions





Solution 1: assignment in blue



Helmut Simonis

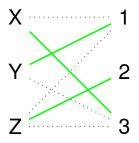
Global Constraints

35

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

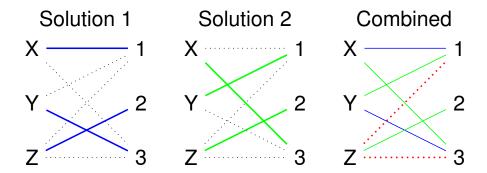
Bounds Consistency Domain Consistency Comparison

Another Simple Example



Solution 2: assignment in green





Combining solutions shows that Z=1 and Z=3 are not possible. Can we deduce this without enumerating solutions?



Helmut Simonis

Global Constraints

37

Problem
Program
Initial Propagation (Forward Checking)
Improved Reasoning
Search

Bounds Consistency Domain Consistency Comparison

Solutions and maximal matchings

- A Matching is subset of edges which do not coincide in any node
- No matching can have more edges than number of variables
- Every solution corresponds to a maximal matching and vice versa
- If a link does not belong to some maximal matching, then it can be removed



Implementation

- Possible to compute all links which belong to some matching
 - Without enumerating all of them!
- Enough to compute one maximal matching
- Requires algorithm for strongly connected components
- Extra work required if more values than variables
- All links (values in domains) which are not supported can be removed
- Complexity: O(n^{1.5}d)



Helmut Simonis

Global Constraints

39

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Domain Consistency

Definition

A constraint achieves *domain consistency*, if for every variable and for every value in its domain, it is possible to find values in the domains of all other variables which satisfy the constraint.

- Also called generalized arc consistency (GAC)
- or hyper arc consistency



Can we still do better?

- NO! This extracts all information from this one constraint
- We could perhaps improve speed, but not propagation
- But possible to use different model
- Or model interaction of multiple constraints



Helmut Simonis

Global Constraints

41

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Should all constraints achieve domain consistency?

- Domain consistency is usually more expensive than bounds consistency
 - Overkill for simple problems
 - Nice to have choices
- For some constraints achieving domain consistency is NP-hard
 - We have to live with more restricted propagation



Improved Propagation in ECLiPSe

- ic_global library bounds consistent version
- ic_global_gac library domain consistent version
- Choose which version to use by using module annotation
- Choice can be passed as parameter



Helmut Simonis

Global Constraints

43

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Declarations

```
:-module(sudoku).
:-export(top/0).
:-lib(ic).
:-lib(ic_global).
:-lib(ic_global_gac).

top:-
    problem(Matrix),
    model(ic_global,Matrix),
    writeln(Matrix).
```



Main Program

```
model(Method, Matrix):-
    Matrix[1..9,1..9] :: 1..9,
     (for(I, 1, 9),
     param (Method, Matrix) do
         Method:alldifferent(Matrix[I,1..9]),
         Method:alldifferent(Matrix[1..9,I])
     ),
     (multifor([I,J],[1,1],[7,7],[3,3]),
     param (Method, Matrix) do
         Method: alldifferent (flatten (Matrix [I..I+2,
                                              J..J+21))
     ),
                                                      onstraint
                                                     omputation
    flatten array (Matrix, List), labeling (List).
                                                        entre
```

Helmut Simonis

Global Constraints

45

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Initial State (Bounds Consistency)

1 2 3	1 2 3 4 5 6 7 8 9 1 2 3 1 2 3	1 2 3 1 2 3 4 5 6 4 5 6 7 8 9 7 8 9	1 2 3 1 2 3 4 5 6 4 5 6 7 8 9 7 8 9 1 2 3 1 2 3	1 2 3 1 2 3 4 5 6 4 5 6 7 8 9 7 8 9 1 2 3 1 2 3
4 5 6	4 5 6 4 5 6	1 7	4 5 6 4 5 6	4 5 6 4 5 6
7 8 9	7 8 9 7 8 9		7 8 9 7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3	1 2 3 1 2 3	3 2
4 5 6	4 5 6 4 5 6	4 5 6	4 5 6 4 5 6	
7 8 9	7 8 9 7 8 9	7 8 9	7 8 9 7 8 9	
1 2 3	1 2 3	1 2 3 1 2 3	8 2	5 1 2 3
4 5 6	4 5 6	4 5 6 4 5 6		4 5 6
7 8 9	7 8 9	7 8 9 7 8 9		7 8 9
1 2 3	9 1 2 3	1 2 3 1 2 3	1 2 3 1 2 3	8 1 2 3
4 5 6	4 5 6	4 5 6 4 5 6	4 5 6 4 5 6	4 5 6
7 8 9	7 8 9	7 8 9 7 8 9	7 8 9 7 8 9	7 8 9
1 2 3	3 7	6 1 2 3	1 2 3	1 2 3 1 2 3
4 5 6		4 5 6	4 5 6	4 5 6 4 5 6
7 8 9		7 8 9	7 8 9	7 8 9 7 8 9
2	7 1 2 3	1 2 3	1 2 3 1 2 3	1 2 3 1 2 3
	4 5 6	4 5 6	4 5 6 4 5 6	4 5 6 4 5 6
	7 8 9	7 8 9	7 8 9 7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3	4 1 2 3	1 2 3 1 2 3
4 5 6	4 5 6 4 5 6	4 5 6	4 5 6	4 5 6 4 5 6
7 8 9	7 8 9 7 8 9	7 8 9	7 8 9	7 8 9 7 8 9
1 2 3	1 2 3 1 2 3	1 2 3 1 2 3	1 2 3	1 2 3
4 5 6	4 5 6 4 5 6	4 5 6 4 5 6	4 5 6	4 5 6
7 8 9	7 8 9 7 8 9	7 8 9 7 8 9	7 8 9	7 8 9



Propagation Steps (Bounds Consistency)

4	1 2 5 6	8	5	6	2 3 5 6 9	1 5 7	1 6 7 9	1 5 6 7 9
3 6 9	2 5 6	3 5 9	~	7	2 3 5 6 9	4 5 8	4 6 9	5 6 8 9
7	6	1 5 9	4	8	9	1 4 5	ന	2
1	4	6	3 7 9	3 9	8	2	15	379
5	9	2	3 4 7	4 6	1	1 3 4 7	80	6
8	3	7	6	2	5	9	4	1
2	7	4	3 4 8 9	5	6	1 3 4 8	1 4 6 9	1 3 6 8 9
6	2 5 6 8	3 5 9	2 3 5 7 8 9	1	4	5 7 8	2 6 7 9	5 6 7 8 9
3	1 2 5 8	1 3 5 9	2 3 5 7 8 9	2 3	7	6	1 2 7 9	4



Helmut Simonis

Global Constraints

47

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

After Setup (Bounds Consistency)

4	1 2	8	5	6	2 3	7	179	7 9
3	2 5	3	1	7	2 3	4 5	6 9	5 8 9
7	6	1 5	4	8	9	1 5	3	2
1	4	6	379	3	8	2	5	7
5	9	2	7	4	1	7	8	6
8	3	7	6	2	5	9	4	7
2	7	4	8 9	5	6	1 3	1 9	8 9
6	5 8	3	2 3 8 9	_	4	5 7 8	2 7 9	5 7 8 9
3	1 5 8	1 5	2	3 9	7	6	1 2	4



Initial State (Domain Consistency)

_			_									_		_	_		_			_				_		
			1	2	3				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Ι.	Δ		4	5	6		X		4	5	6	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
Ι '		•	7	8	9		U		7	8	9	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9
1	2	3	1	2	3	1	2	3							1	2	3	1	2	3	1	2	3	1	2	3
4	5	6	4	5	6	4	5	6		1			7	'	4	5	6	4	5	6	4	5	6	4	5	6
7	8	9	7	8	9	7	8	9					1		7	8	9	7	8	9	7	8	9	7	8	9
1	2	3	1	2	3	1	2	3	1	2	3	Н		_	1	2	3	1	2	3	Ė	Ť	Ť	r-	Ť	Ť
4	5	6	4	5	6	4	5	6	4	5	6		0		4	5	6	ľ	5	6		0)
		- 1	7		- 1	7	8						Ö)	7	8	9	7	8			J)		Z	
7	8	9	÷	8	9	<u> </u>	<u> </u>	9	7	8	9	H	Ť	_	-	<u> </u>	9	_	0	9	_	_	_	H	_	_
1	2	3	1	2	3				1	2	3	1	2	3								_		1	2	3
4	5	6	4	5	6		h		4	5	6	4	5	6		X			\mathcal{I}	'		<u>'</u>)	4	5	6
7	8	9	7	8	9		<u> </u>		7	8	9	7	8	9		$oldsymbol{ol}}}}}}}}}}}}}}}$	_		_			U		7	8	9
1	2	3				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3				1	2	3
4	5	6		u		4	5	6	4	5	6	4	5	6	4	5	6	4	5	6		Я		4	5	6
7	8	9		U	'	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9		U	'	7	8	9
1	2	3										1	2	3	1	2	3				1	2	3	1	2	3
4	5	6		2			7	1		C		4	5	6	4	5	6		വ	١	4	5	6	4	5	6
7	8	9		U)					U	1	7	8	9	7	8	9		J	,	7	8	9	7	8	9
Ė	Ť	Ť			$\overline{}$	1	2	3	1	2	3	Ė	Ť	Ť	Ť	2	3	1	2	3	1	2	3	1	2	3
)		7	•	4	5	6	4	5	6			'	4	5	6	4	5	6	4	5	6	4	5	6
Ι.	Z			/		7	8	9	7	8	9		J)	7	8	9	7	8	9	7	8	9	7	8	9
4	2	3	1	2	3	1	2	3	4	2	3	⊢	-		H	0	3	4	2	3	4	2	3	4	2	3
Ľ	-	- 1	Ľ	_	- 1				Ľ				1			A		Ľ			Ľ		- 1	Ľ		
4	5	6	4	5	6	4	5	6	4	5	6					4		4	5	6	4	5	6	4	5	6
7	8	9	7	8	9	7	8	9	7	8	9	L	-		L	-		7	8	9	7	8	9	7	8	9
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3				1	2	3	l	4	
4	5	6	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6		h		4	5	6	١,	Δ	
7	8	9	7	8	9	7	8	9	7	8	9	7	8	9	7	8	9		U	,	7	8	9			



Helmut Simonis

Global Constraints

49

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Bounds Consistency Domain Consistency Comparison

Propagation Steps (Domain Consistency)

4	2	8	5	6	3	1	1 6 7 9	1 5 6 7 9
3 6 9	5	3 5 9	~	7	2	4	6	8
7	6	1	4	8	9	5	3	2
1	4	6	3 7 9	3 9	8	2	5	379
5	9	2	3 7	4	1	1 3 4 7	8	6
8	ര	7	6	2	5	9	4	1
2	7	4	3 8 9	5	6	8	1	1 3 6 8 9
6	8	5 9	2	1	4	5 7 8	2 6 7 9	5
3	1	5	8	2 3	7	6	2	4



After Setup (Domain Consistency)

4	2	8	5	6	3	1	7 9	7 9
3	5	3	1	7	2	4	6	8
7	6	~	4	8	9	5	3	2
1	4	6	379	3	8	2	5	7
5	9	2	7	4	1	7	8	6
8	ര	7	6	2	5	9	4	~
2	7	4	3	5	60	8	~	3 9
6	8	3	2	1	4	7	7 9	5
3	1	5	8	3	7	6	2	4



Helmut Simonis

Global Constraints

51

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Comparison

Comparison

Forward Checking

4	1	5 6	8	2 3 5 9	3 6 9	2 3 6 9	1 5 7	1 6 7 9	5 6 7 9
	3 6 9	5 6	5 9	$\overline{}$	7	2 3 6 9	4 5 8	6 9	5 8
7	6 9	5 6	1 5 9	4 5 9	8	6	1 4 5 7	3	2
1		4	6	7 9	3	8	2	5	7
5		9	2	3 4 7	3 4	1 3	7	8	7
8		3	7	6	2	5	9	4	1
2		7	1 3 4 9	8 9	5	3 6 9	1 3	1 9	8 9
	3 6 9	5 6 8	5 9	2 3 7 8 9	1	4	5 7 8	2 7 9	5 7 8 9
	3 1 9	5	1 3 5 9	2 3 7 8 9	3 9	2 3 7 9	6	1 2 7 9	4

Bounds Consistency Domain Consistency

4	1 2	8	5	6	2 3	1 7	1 7 9	7 9
3	2 5	3 9	1	7	2 3	4 5 8	6 9	5 8 9
7	6	1 5	4	8	9	5	ര	2
1	4	6	7 9	3	8	2	5	7
5	9	2	7	3	$ \leftarrow $	7	8	6
8	3	7	6	2	5	9	4	1
2	7	4	8 9	5	6	1 3	1 9	8 9
6	5 8	3	2 3	1	4	5 7 8	2 7 9	5 7 8 9
3	1 5 8	1 5	2	3	7	6	1 2	4

4	2	8	5	6	3	1	7 9	7 9
3	5	3 9	1	7	2	4	60	8
7	6	1	4	8	9	5	ദ	2
1	4	6	7 9	3	8	2	5	7
5	9	2	7	4	$ \leftarrow $	7	8	6
8	3	7	6	2	5	9	4	1
2	7	4	3 9	5	6	8	1	3
6	8	3	2	$ \leftarrow $	4	7	7 9	5
3	1	5	8	3	7	6	2	4



Typical?

- This does not always happen
- Sometimes, two methods produce same amount of propagation
- Possible to predict in certain special cases
- In general, tradeoff between speed and propagation
- Not always fastest to remove inconsistent values early
- But often required to find a solution at all



Helmut Simonis Global Constraints 53

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

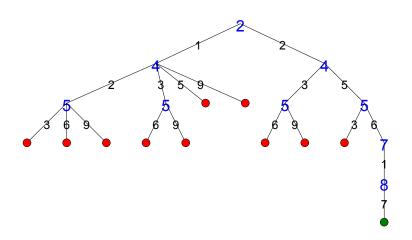
Solution

Simple search routine

- Enumerate variables in given order
- Try values starting from smallest one in domain
- Complete, chronological backtracking



Search Tree (Forward Checking)





Helmut Simonis

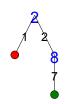
Global Constraints

55

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Solution

Search Tree (Bounds Consistency)





Search Tree (Domain Consistency)





Helmut Simonis

Global Constraints

57

Problem Program Initial Propagation (Forward Checking) Improved Reasoning Search

Solution

Observations

- Search tree much smaller for bounds/domain consistency
- Does not always happen like this
- Smaller tree = Less execution time
- Less reasoning = Less execution time
- Problem: Finding best balance
- For Sudoku: not good enough, should not require any search!



Solution

4	2	8	5	6	3	1	7	9
3	5	9	~	7	2	4	6	8
7	6	~	4	80	9	5	ന	2
1	4	6	3	ග	80	2	5	7
5	9	2	7	4	~	3	8	6
8	3	7	6	2	5	9	4	1
2	7	4	9	5	6	8	~	3
6	8	3	2	~	4	7	9	5
9	1	5	8	3	7	6	2	4



Helmut Simonis

Global Constraints

50

Conclusions

Complete Example: Domain Consistent Alldifferent Generic Model Exercises

Global Constraints

- Powerful modelling abstractions
- Efficient reasoning



Consistency Levels

- Defined levels of propagation
- Tradeoff speed/reasoning
- Characterisation of power of constraint



Helmut Simonis

Global Constraints

61

Conclusions

Complete Example: Domain Consistent Alldifferent Generic Model Exercises

Alldifferent Variants

- Forward Checking
 - Only reacts when variables are assigned
 - Equivalent to decomposition into binary constraints
- Bounds Consistency
 - Typical best compomise speed/reasoning
 - Works well if no holes in domain
- Domain Consistency
 - Extracts all information from single constraint
 - Cost only justified for very hard problems



Bigger Example



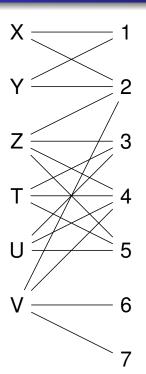
Helmut Simonis

Global Constraints

63

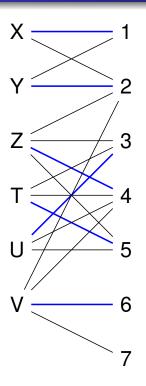
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Making constraint domain consistent



Problem shown as bipartite graph





Find maximal matching (in blue)



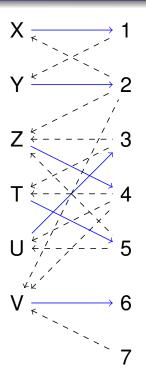
Helmut Simonis

Global Constraints

65

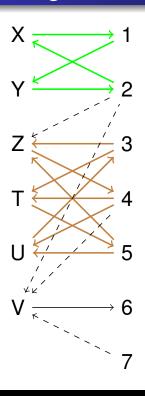
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Making constraint domain consistent



Orient graph (edges in matching from variables to values, all others from values to variables), mark edges in matching





Find strongly connected components (green and brown), mark their edges



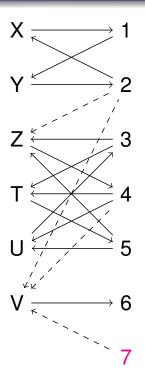
Helmut Simonis

Global Constraints

67

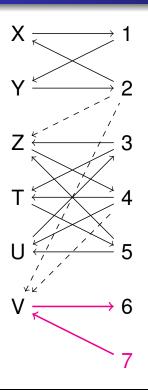
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Making constraint domain consistent



Find unmatched value nodes (here node 7, magenta)





Find alternating paths from such nodes (in magenta), mark their edges



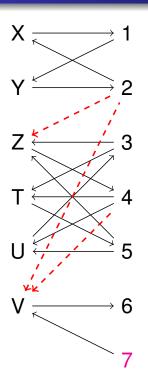
Helmut Simonis

Global Constraints

69

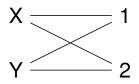
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Making constraint domain consistent

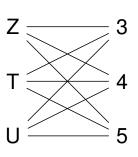


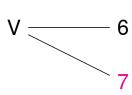
All unmarked edges can be removed





Resulting graph, constraint is domain consistent







Helmut Simonis

Global Constraints

71

Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Extended Example

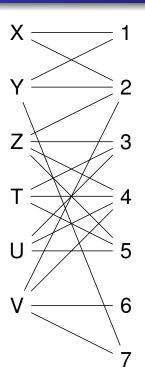
```
:-lib(ic).
:-lib(ic_global_gac).

top:-
    X :: 1..2,
    Y :: [1,2,7],
    Z :: 2..5,
    [T,U] :: 3..5,
    V :: [2,4,6,7],
    ic_global_gac:alldifferent([X,Y,Z,T,U,V]).
```

Helmut Simonis

Global Constraints

72



Problem shown as bipartite graph



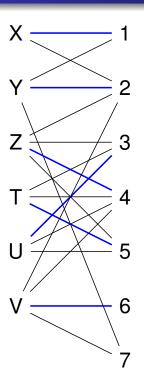
Helmut Simonis

Global Constraints

73

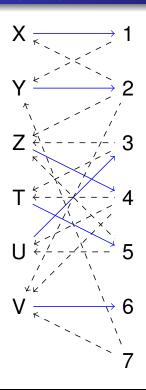
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

No propagation in expanded example



Find maximal matching (in blue)





Orient graph (edges in matching from variables to values, all others from values to variables), mark edges in matching



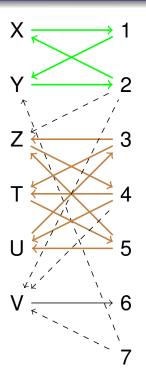
Helmut Simonis

Global Constraints

75

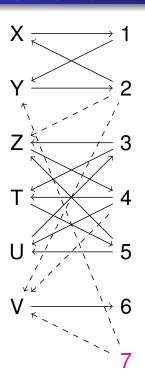
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

No propagation in expanded example



Find strongly connected components (green and brown), mark their edges





Find unmatched value nodes (here node 7, magenta)



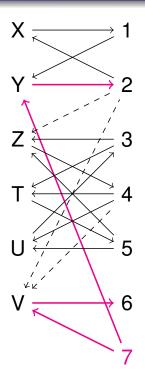
Helmut Simonis

Global Constraints

77

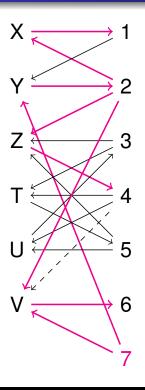
Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

No propagation in expanded example



Find alternating paths from such nodes (in magenta), mark their edges





Continue with alternating paths



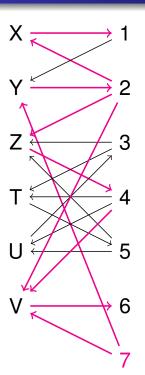
Helmut Simonis

Global Constraints

79

Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

No propagation in expanded example



Continue with alternating paths, all edges marked, no propagation, constraint is domain consistent



Observation

- A lot of effort for no propagation
- Problem: Slows down search without any upside
- Constraint is woken every time any domain is changed
- How often does the constraint do actual pruning?



Helmut Simonis

Global Constraints

81

Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

Generalize Program for different sizes

- How to generalize program for different sizes (4,9,16,25,36...)
- Add parameter R (Order, number of blocks in a row/column)
- Size N is square of R
- Remove explicit integer bounds by expressions
- Useful to do this change as rewriting of working program



Main Program

```
model(R,M,Matrix):-
    N is R*R, Matrix[1...N, 1...N] :: 1...N,
     (for(I,1,N),
     param(N, M, Matrix) do
         M:alldifferent(Matrix[I,1..N]),
         M:alldifferent(Matrix[1..N,I])
    ),
     (multifor([I,J],[1,1],[N-R+1,N-R+1],[R,R]),
     param(R, M, Matrix) do
         M:alldifferent(flatten(Matrix[I..I+R-1,
                                       J..J+R-11)
    ),
                                                      onstraint
                                                      omputation
    flatten array (Matrix, List), labeling (List).
                                                        entre
```

Helmut Simonis

Global Constraints

83

Conclusions
Complete Example: Domain Consistent Alldifferent
Generic Model
Exercises

More Information

- Krzysztof R. Apt and Mark Wallace.

 Constraint Logic Programming using ECLiPSe.

 Cambridge University Press, New York, NY, USA, 2007.
- Willem Jan van Hoeve.
 The alldifferent constraint: A survey.

 CoRR, cs.PL/0105015, 2001.
- Claude-Guy Quimper.

 Efficient Propagators for Global Constraints.

 PhD thesis, University of Waterloo, 2006.



More Information



H. Simonis.

Sudoku as a constraint problem.

In B. Hnich, P. Prosser, and B. Smith, editors, *Proceedings* of the 4th International Workshop on Modelling and Reformulating Constraint Satisfaction Problems, pages 13-27, September 2005.



I. Lynce and J. Ouaknine.

Sudoku as a SAT problem.

In 9th International Symposium on Artificial Intelligence and Mathematics, January 2006.

Helmut Simonis

Global Constraints

85

onstraint omputation **e**ntre

Conclusions Complete Example: Domain Consistent Alldifferent Generic Model Exercises

More Information



H. Simonis.

Kakuro as a constraint problem.

In P. Flener, H. Simonis, editors, Proceedings of the 8th International Workshop on Modelling and Reformulating Constraint Satisfaction Problems, September 2008.



Exercises





Helmut Simonis

Global Constraints

87