# Global constraints

# Global constraints

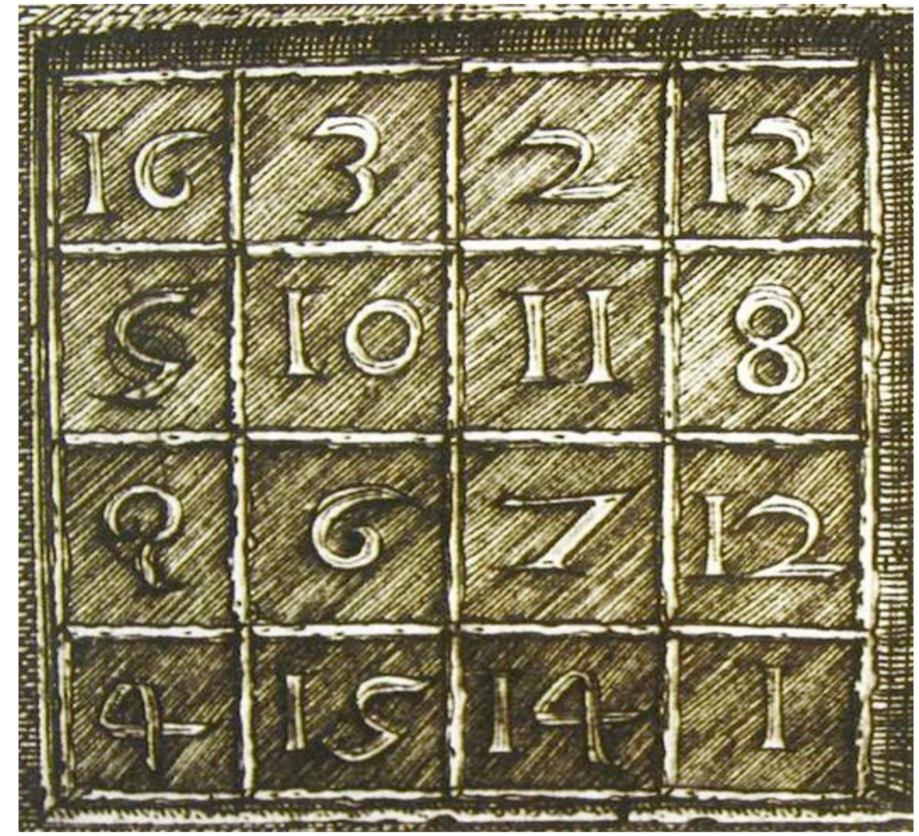
**Definition 13 (Global constraint)** *A global constraint $G$ is a* class *of constraints defined by a Boolean function $f_G$ that takes as input an unbounded number of variables. That is, given $n_0$, there exists an integer $n > n_0$ such that $\mathbb{Z}^n \cap f_G \neq \emptyset$.*

**`Alldifferent`($X_1,\ldots,X_n$):**
all variables take different values ($X_i \neq X_j$ for all $i,j$)

**`Sum`($X_1,\ldots,X_n,N$):**
$N$ is equal to the sum of the $X_i$ 's ($\sum X_i = N$)

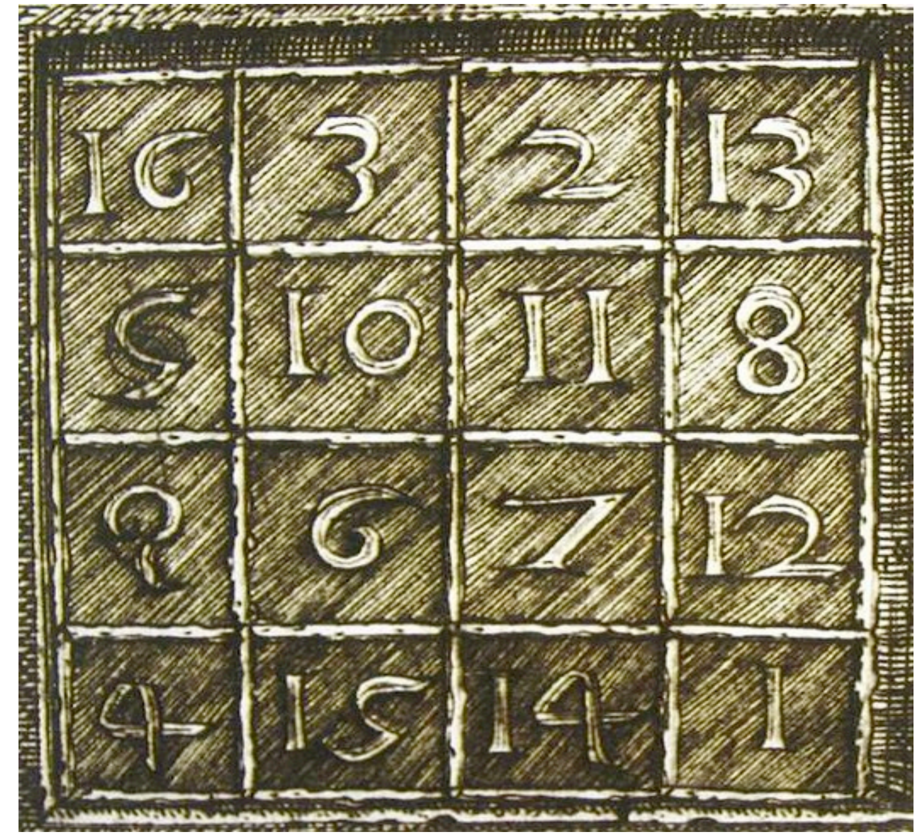
**`AtLeast-p-v`($X_1,\ldots,X_n$):**
at least $p$ variables take value $v$ ($|\{i \in 1..n \mid X_i = \mathrm{v}\}| \geq p$)

# Magic Square

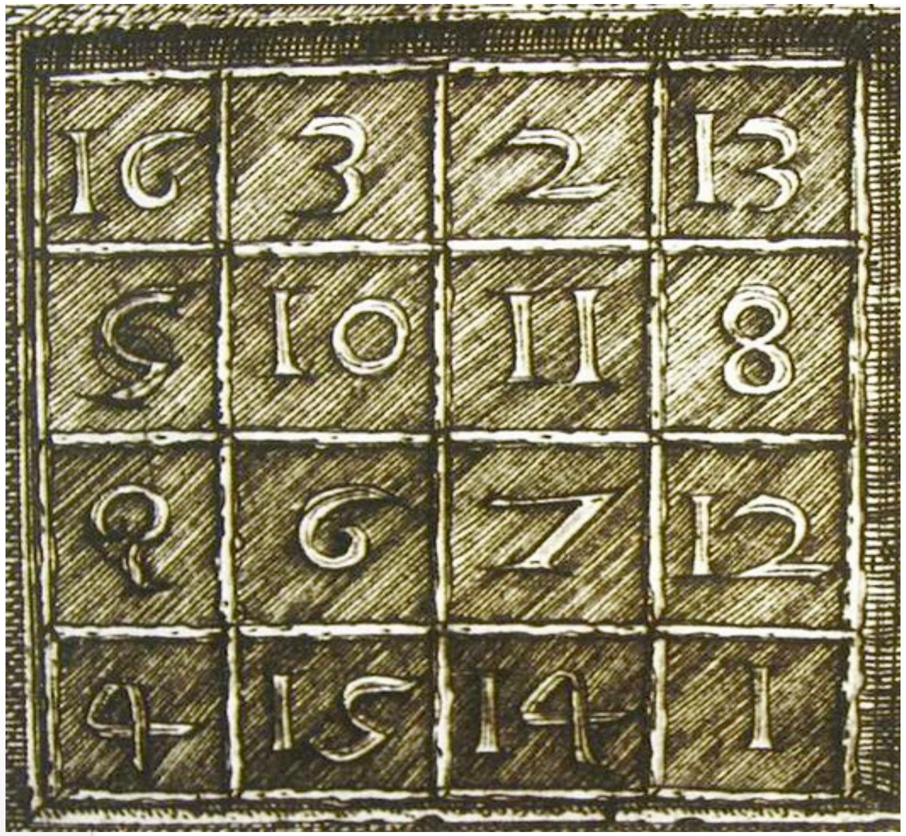


# Magic Square

- SAT: > 600.000 clauses



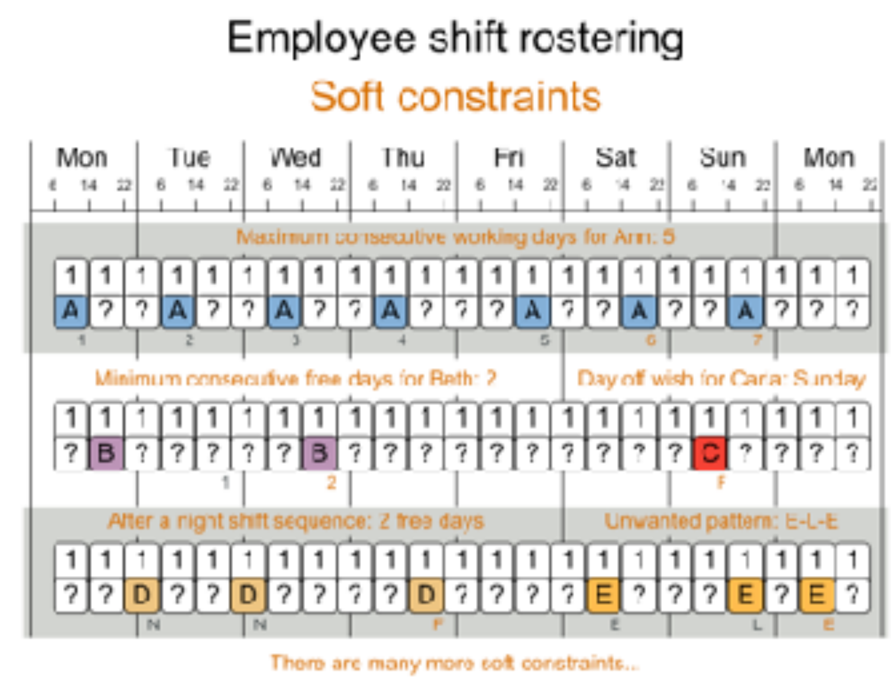
# Magic Square



- SAT: > 600.000 clauses
- CP:

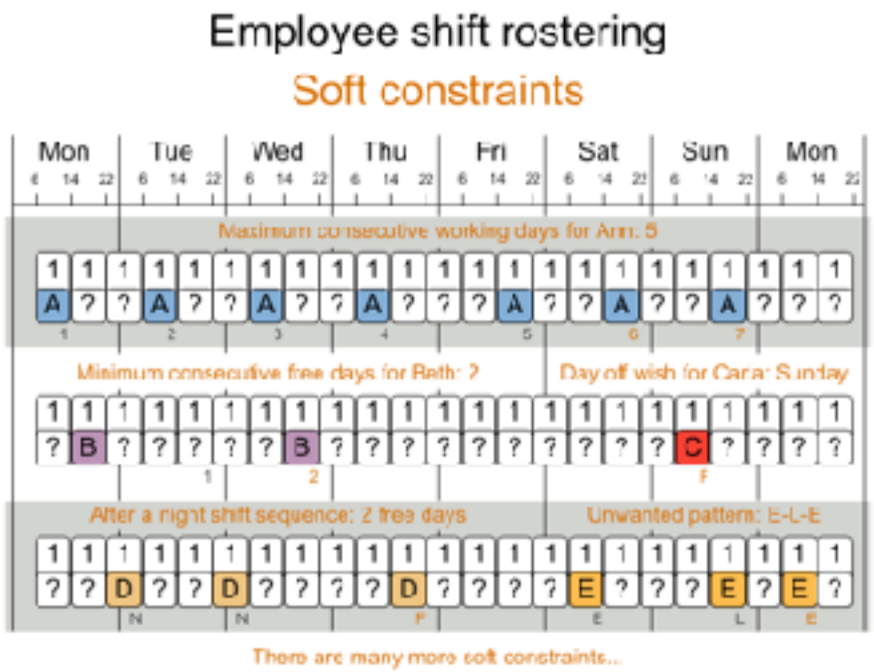
```
% All different on cells
constraint
alldifferent(i,j in 1..4)(magic[i,j]);

% sum in rows.
constraint forall (i in 1..4) (
  sum(j in 1..4)(magic[i,j]) = 34 );

% sum in columns.
constraint forall (j in 1..4) (
  sum(i in 1..4)(magic[i,j]) = 34 );

% sum in diagonals.
constraint
  sum(i in 1..4)(magic[i,i]) = 34;
  sum(i in 1..4)(magic[i,4-i+1]) = 34;

solve satisfy;
```

# Nurse Rostering

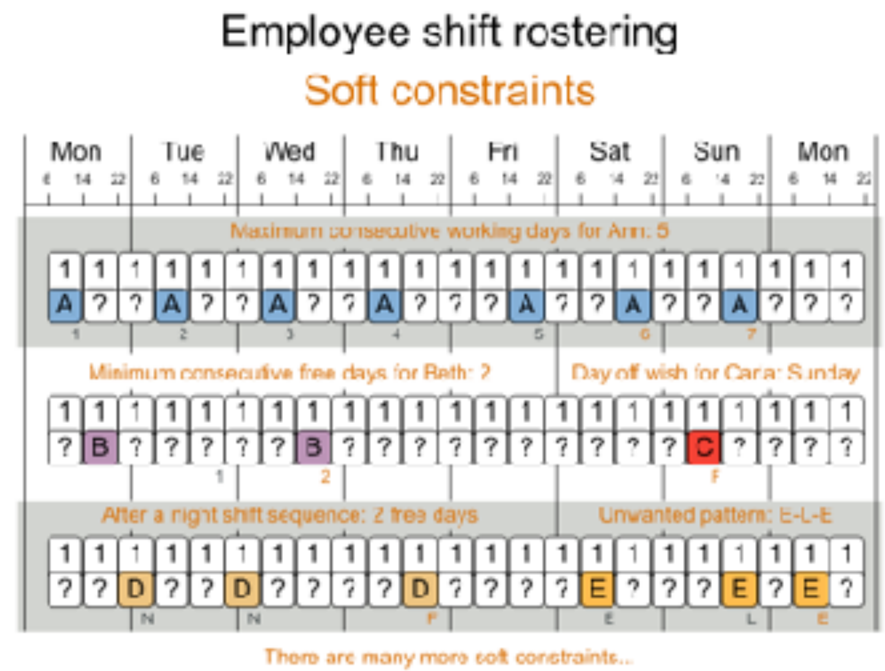


# Nurse Rostering

- Linear Programming: more than 10.000 lines



# Nurse Rostering




- Linear Programming: more than 10.000 lines
- CP:

```
int: Q = 6; int: q0 = 1; set of int: STATES = 1..Q;
array[STATES,SHIFTS] of int: t =
     [| 2, 3, 1    % state 1
      | 4, 4, 1    % state 2
      | 4, 5, 1    % state 3
      | 6, 6, 1    % state 4
      | 6, 0, 1    % state 5
      | 0, 0, 1|]; % state 6

array[NURSES,DAYS] of var SHIFTS: roster;

constraint forall(j in DAYS)(
             sum(i in NURSES)(bool2int(roster[i,j] == d)) == req_day /\
             sum(i in NURSES)(bool2int(roster[i,j] == n)) == req_night
           );
constraint forall(i in NURSES)(
             regular([roster[i,j] | j in DAYS], Q, S, t, q0, STATES) /\
             sum(j in DAYS)(bool2int(roster[i,j] == n)) >= min_night
           );

solve satisfy;
```

# Global constraints and propagation

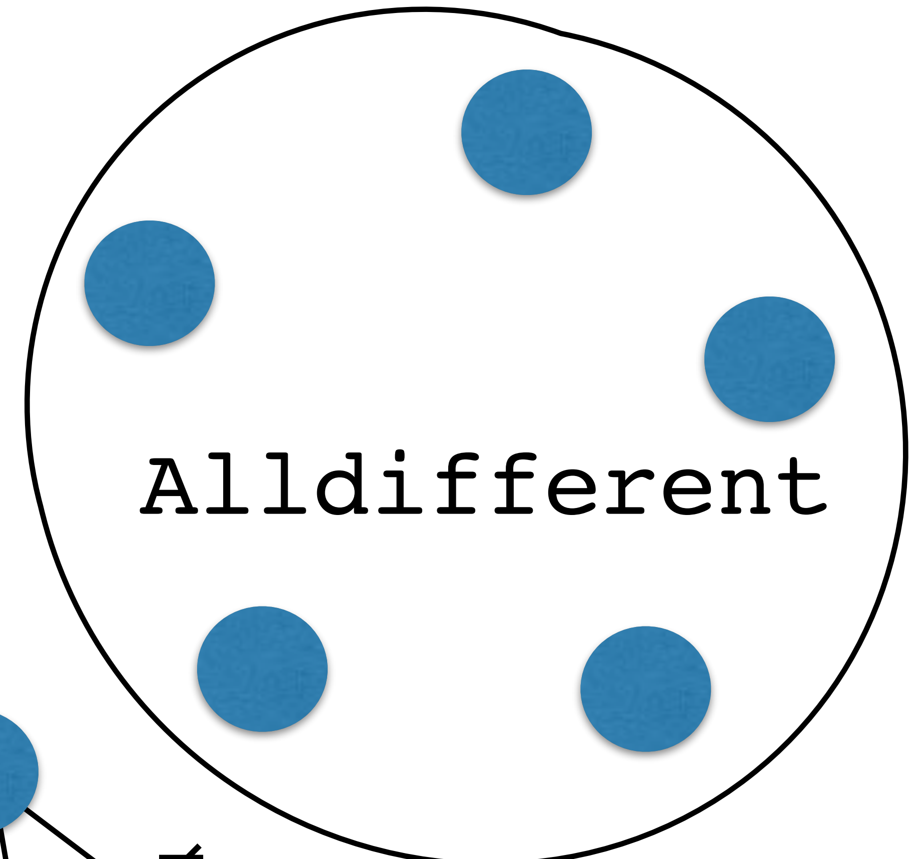
- Global constraints compactly express properties of a problem
- Arc consistency allows strong propagation (because they preserve properties of the problem)
- **BUT**: Using a generic AC algorithm on $C(X_1 \ldots X_n)$ is in $O(d^n)$
  - ➡ We need ad hoc propagators
- **BUT**: there are more than 400 global constraints in the catalog!
  - ➡ Decompose!

# Simple Decompositions for Global Constraints

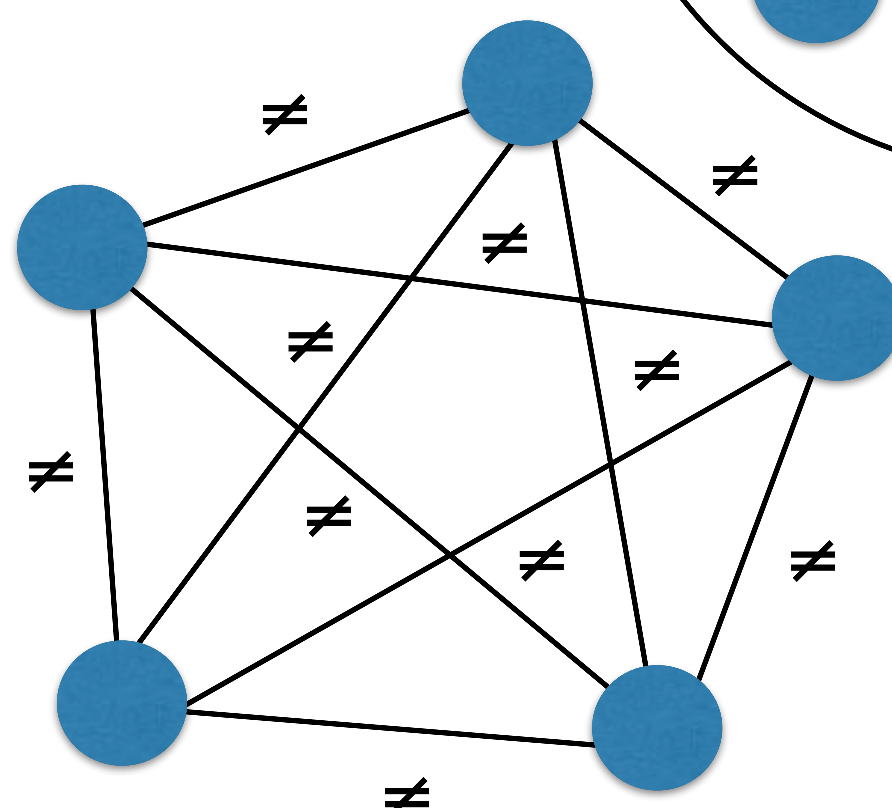
- A simple decomposition $\boldsymbol{\delta}_G$ of a global constraint G is a function that given an instance $c(X)=(X,D_X,\{c\})$ of $G$ returns a network $\boldsymbol{\delta}_G(c)=(X,D_X,C)$ such that:
    - for all $c_i \in C$, $c_i$ involves a ***bounded*** number of variables
    - $sol(c) = sol(\boldsymbol{\delta}_G(c))$

# Example 1: Alldifferent

- `Alldifferent`($X_1 \ldots X_n$)




- Decomposition:
    - for all $i,j,i \neq j$: $X_i \neq X_j$



# Decompositions for Global Constraints

- A decomposition $\boldsymbol{\delta}_G$ of a global constraint G is a function that given an instance $c(X)=(X,D_X,\{c\})$ of $G$ returns a network $\boldsymbol{\delta}_G(c) =(X \cup Y,D_X+D_Y,C)$ such that:
    - for all $c_i \in C$, $c_i$ involves a ***bounded*** number of variables
    - $sol(c) = sol(\boldsymbol{\delta}_G(c))[X]$
    - $|Y|+|D_Y|$ is polynomial in $|X|+|D_X|$

# Example 2: AtLeast

- **`AtLeast-p-v`**($X_1 \ldots X_n$): $\boxed{X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n}$

# Example 2: AtLeast

- **`AtLeast-p-v`**($X_1 \ldots X_n$): $\boxed{X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n}$

- Decomposition:

$$
\begin{array}{cccccc}
 & X_1 & \ldots. & X_i & \ldots. & X_n \\
B_0 & B_1 & \ldots B_{i-1} & B_i & \ldots. & B_n
\end{array}
$$

$D(B_i)=\{0,..,n\}$

# Example 2: AtLeast

- **`AtLeast-p-v`**$(X_1 \ldots X_n)$: $\boxed{X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n}$

- Decomposition:

$$X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n$$

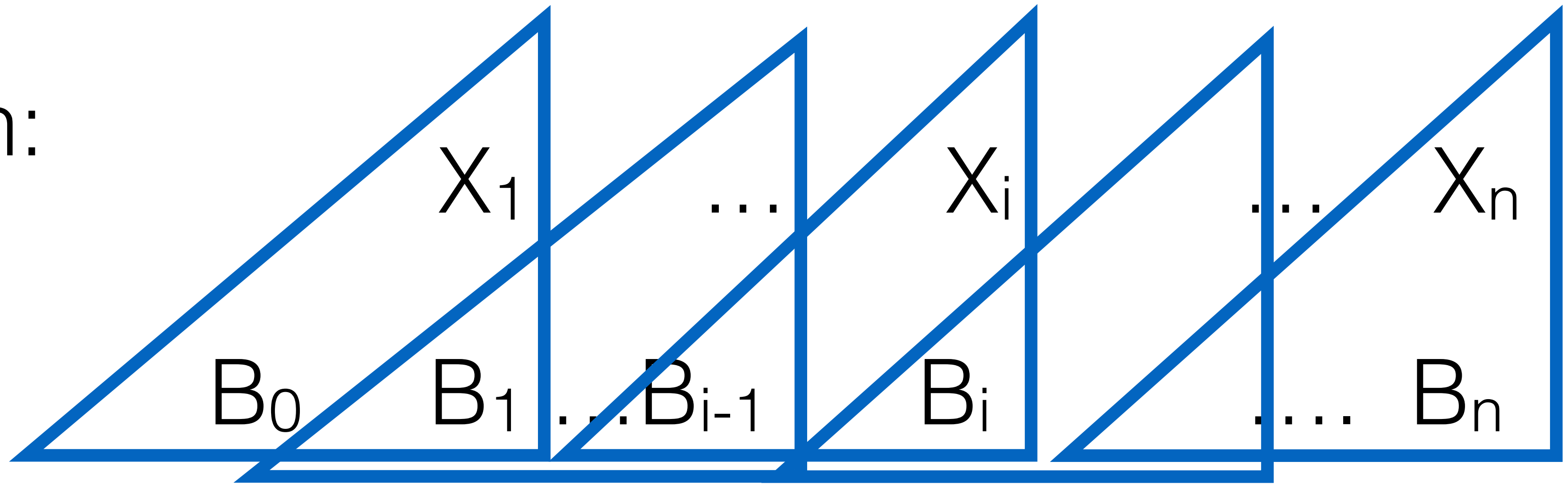
$$B_0 \quad B_1 \quad \ldots B_{i-1} \quad B_i \quad \ldots. \quad B_n$$

$D(B_i)=\{0,..,n\}$

$(X_i = v \;\&\; B_i=B_{i-1}+1) \vee (X_i \neq v \;\&\; B_i=B_{i-1})$

# Example 2: AtLeast

- **`AtLeast-p-v`**($X_1 \ldots X_n$): | $X_1$ …. $X_i$ …. $X_n$ |
- Decomposition:




$D(B_i)=\{0,..,n\}$

$(X_i = v \ \& \ B_i=B_{i-1}+1) \vee (X_i \neq v \ \& \ B_i=B_{i-1})$

$B_0=0, \ B_n \geq p$

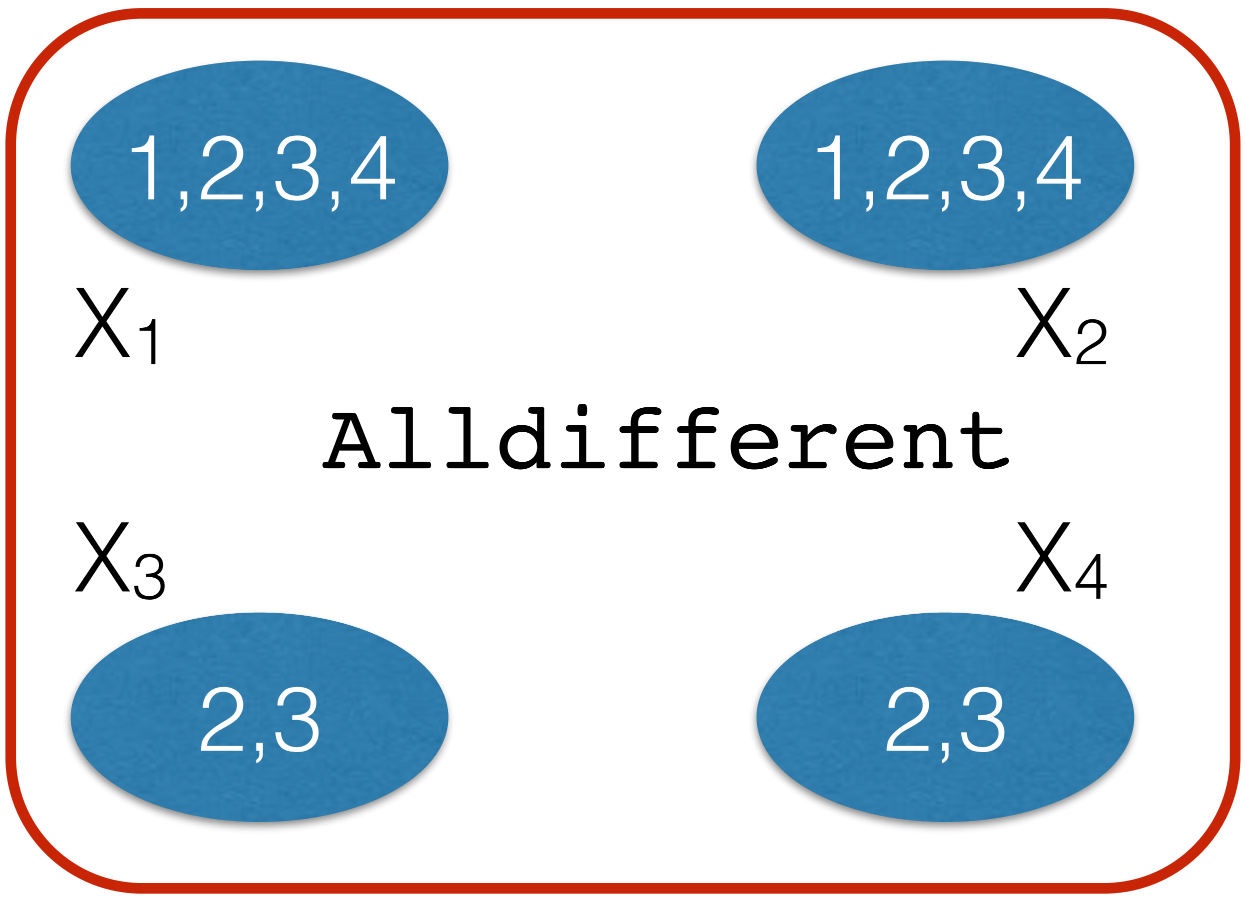
# Do Decompositions Preserve Propagation?

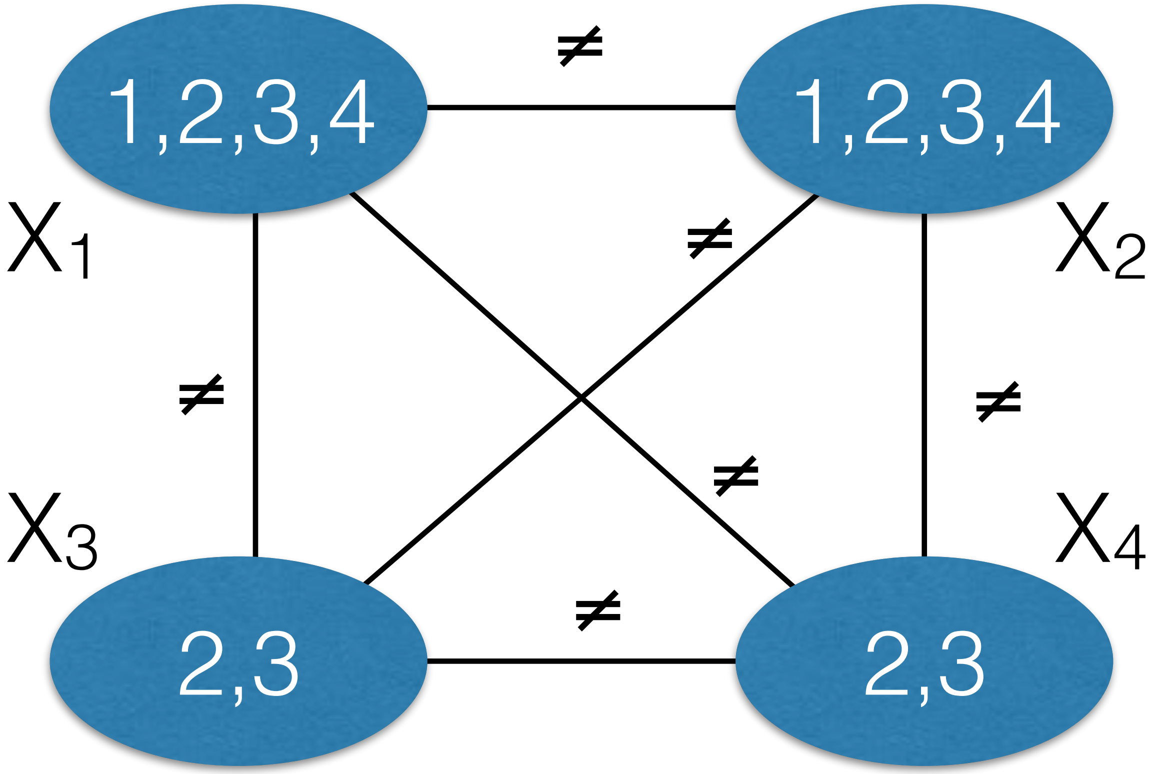
- A decomposition $\boldsymbol{\delta}_G$ of a global constraint $G$ ***preserves arc consistency*** iff for any instance $c(X)=(X,D_X,\{c\})$ of $G$ and any domain $D'_X \subseteq D_X$,

$$D'_{AC}(c) = D'_{AC}(\boldsymbol{\delta}_G(c))[X]$$

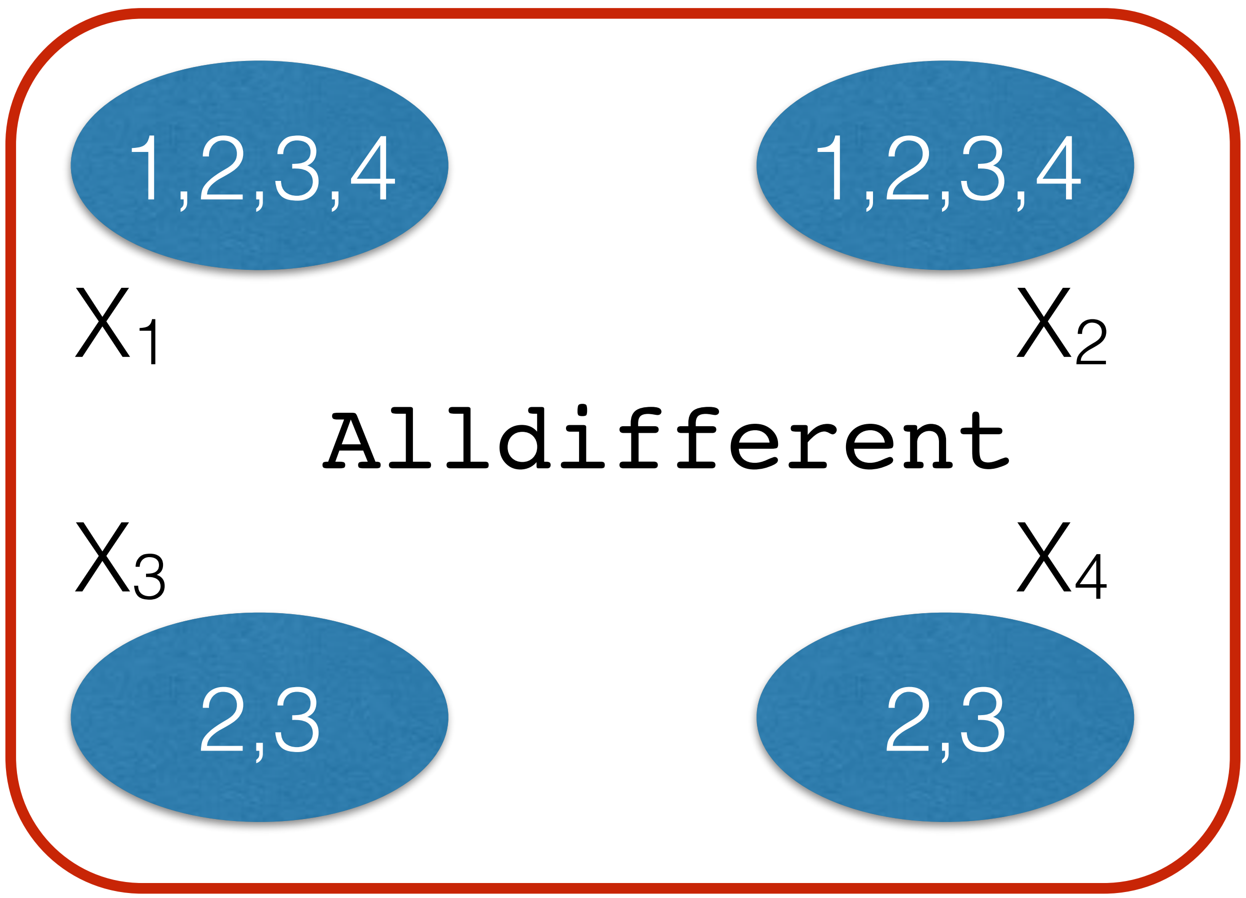
- Constraint $G$ is said ***AC-decomposable***

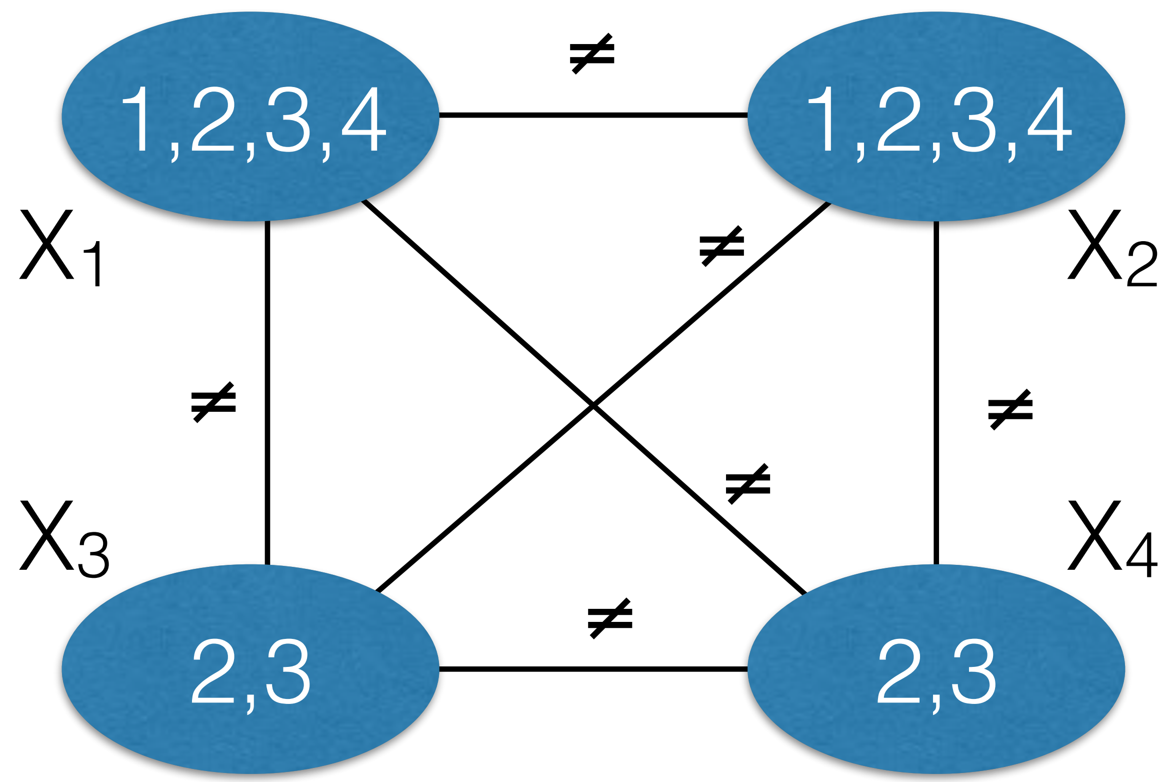
# Alldifferent: No!



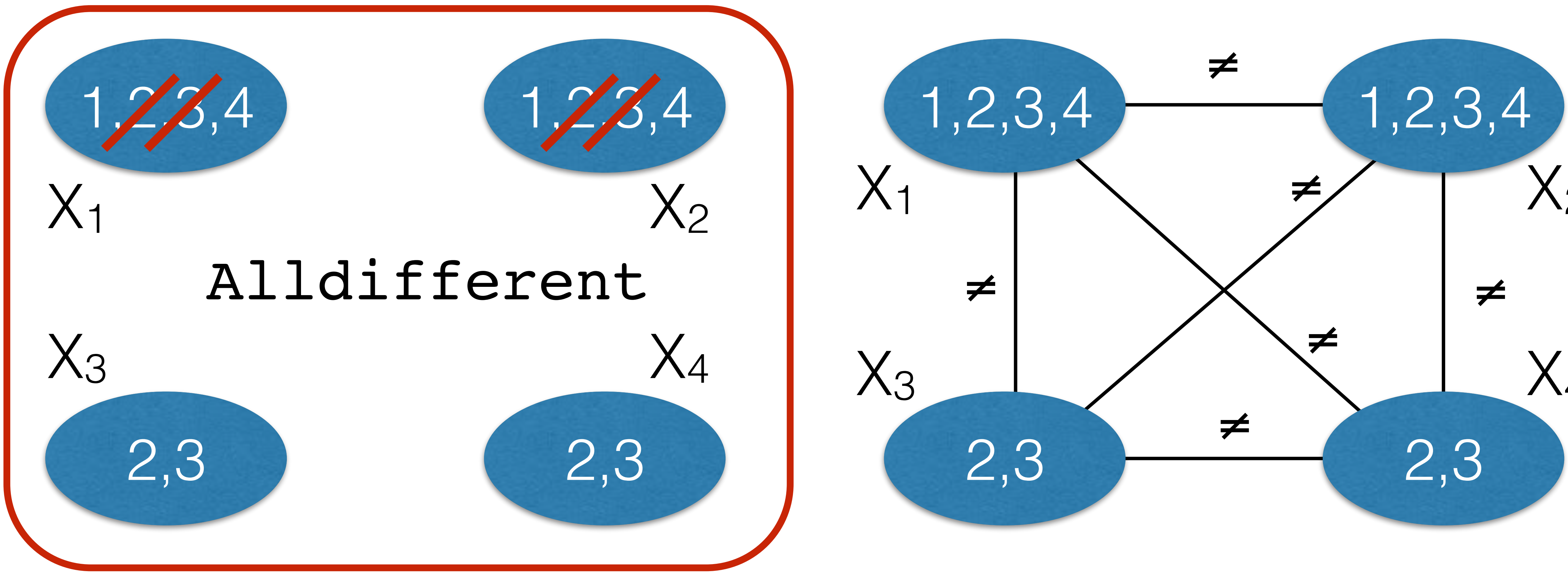
# Alldifferent: No!




already arc consistent

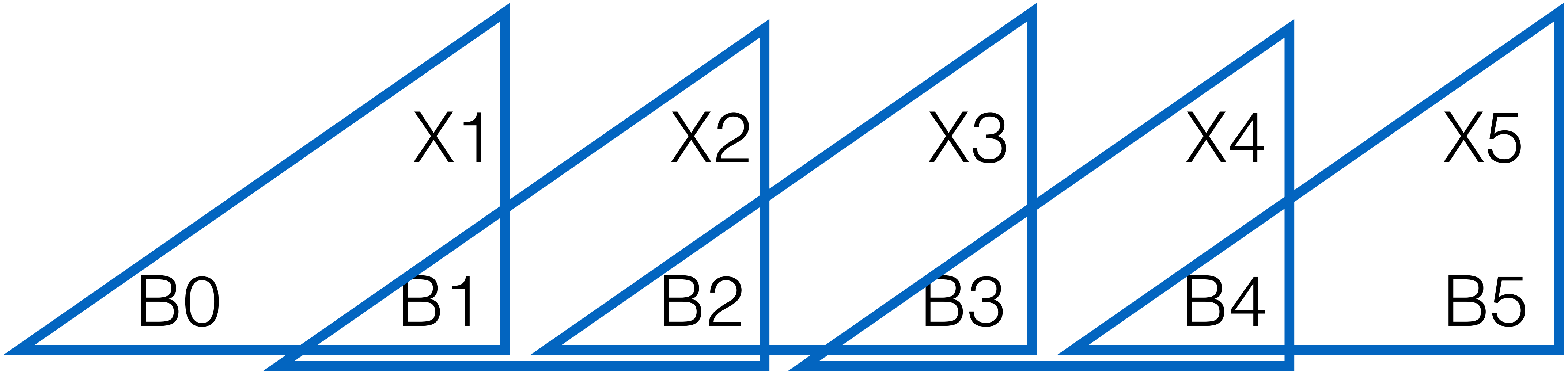
# Alldifferent: No!




already arc consistent

# AtLeast: Yes!

- **`AtLeast-p-v`**($X_1 \ldots X_n$)



# Why?

- Why some decompositions preserve AC and some don't?
- Why some constraints can be AC-decomposed and some not?

# Case 1: NP-hard constraints

**Definition 17** *Given a global constraint $G$, the problem $checker_G$ is defined by :*

- *Instance : $(X(c), D, \{c\})$, where $c$ is an instance of $G$*
- *Question : Is there a tuple $\tau \in D^{X(c)}$ such that $c(\tau)$ ?*

- **Lemma** Given a constraint *G*, if *$checker_G$* is NP-complete on *G*, then arc consistency is NP-hard on *G*

**Theorem 2** *If a global constraint $G$ is such that $checker_G$ is NP-complete, then there doesn't exist any AC-decomposition of $G$.*

- Examples: `NValue`($X_1 \ldots X_n$,$N$), `Sum`($X_1 \ldots X_n$,$N$), etc

# Case 1: Apply weaker consistency —> Bound consistency

- We relax the notion of support
- **Bound support:**
  - it is a tuple $t \in c(X)$ and for all $X_i \in X$, $\min(X_i) \leq t[X_i] \leq \max(X_i)$
- We ensure bound support only for the min and max of each domain

# Bound consistency

- ***$X_1 + X_2 = X_3$***
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

bound supports:

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| 1 | 1 | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | 5 | 5 |
| 6 | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

| | | |
|---|---|---|
| $\boxed{1}$ | 1 | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | 5 | 5 |
| 6 | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

bound supports:

1 4 5

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains:

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| $\boxed{1}$ | 1 | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | 5 | 5 |
| $\boxed{6}$ | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

bound supports:

1 4 5

6 4 10

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains:

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| $\boxed{1}$ | $\boxed{1}$ | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | 5 | 5 |
| $\boxed{6}$ | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

| $X_1$ | $X_2$ | $X_3$ |
| --- | --- | --- |
| $\boxed{1}$ | $\boxed{1}$ | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | $\boxed{5}$ | 5 |
| $\boxed{6}$ | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

bound supports:

1 4 5

6 4 10

5 1 6

5 5 10

# Bound consistency

- ***$X_1 + X_2 = X_3$***
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |
| 5 | 5 | 10 |

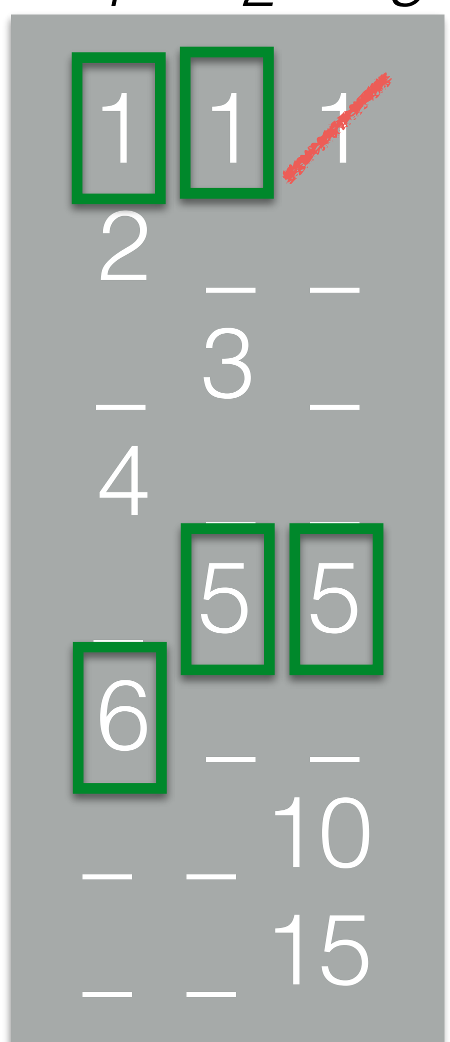
domains:

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| $\boxed{1}$ | $\boxed{1}$ | ~~1~~ |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | $\boxed{5}$ | 5 |
| $\boxed{6}$ | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$




bound supports:

1 4 5

6 4 10

5 1 6

5 5 10

# Bound consistency

- ***$X_1 + X_2 = X_3$***
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| $\boxed{1}$ | $\boxed{1}$ | $\cancel{1}$ |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | | |
| _ | $\boxed{5}$ | $\boxed{5}$ |
| $\boxed{6}$ | _ | _ |
| _ | _ | 10 |
| _ | _ | $\cancel{15}$ |

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |
| 5 | 5 | 10 |

# Bound consistency

- ***$X_1 + X_2 = X_3$***
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| $\boxed{1}$ | $\boxed{1}$ | $\cancel{1}$ |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | $\boxed{5}$ | $\boxed{5}$ |
| $\boxed{6}$ | _ | |
| _ | _ | $\boxed{10}$ |
| _ | _ | $\cancel{15}$ |

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |
| 5 | 5 | 10 |

# Bound consistency

- $X_1 + X_2 = X_3$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains:

| $X_1$ | $X_2$ | $X_3$ |
|---|---|---|
| **1** | **1** | ~~1~~ |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | **5** | **5** |
| **6** | _ | _ |
| _ | _ | **10** |
| _ | _ | ~~15~~ |

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |
| 5 | 5 | 10 |

supports:

| | | |
|---|---|---|
| 2 | 3 | 5 |
| 4 | 1 | 5 |

# Bound consistency

- ***$X_1 + X_2 = X_3$***
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$

domains: $X_1$ $X_2$ $X_3$

| | | |
|---|---|---|
| $\boxed{1}$ | $\boxed{1}$ | ~~1~~ |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | $\boxed{5}$ | $\boxed{5}$ |
| $\boxed{6}$ | _ | _ |
| _ | _ | $\boxed{10}$ |
| _ | _ | ~~15~~ |

bound supports:

| | | |
|---|---|---|
| 1 | 4 | 5 |
| 6 | 4 | 10 |
| 5 | 1 | 6 |
| 5 | 5 | 10 |

supports:

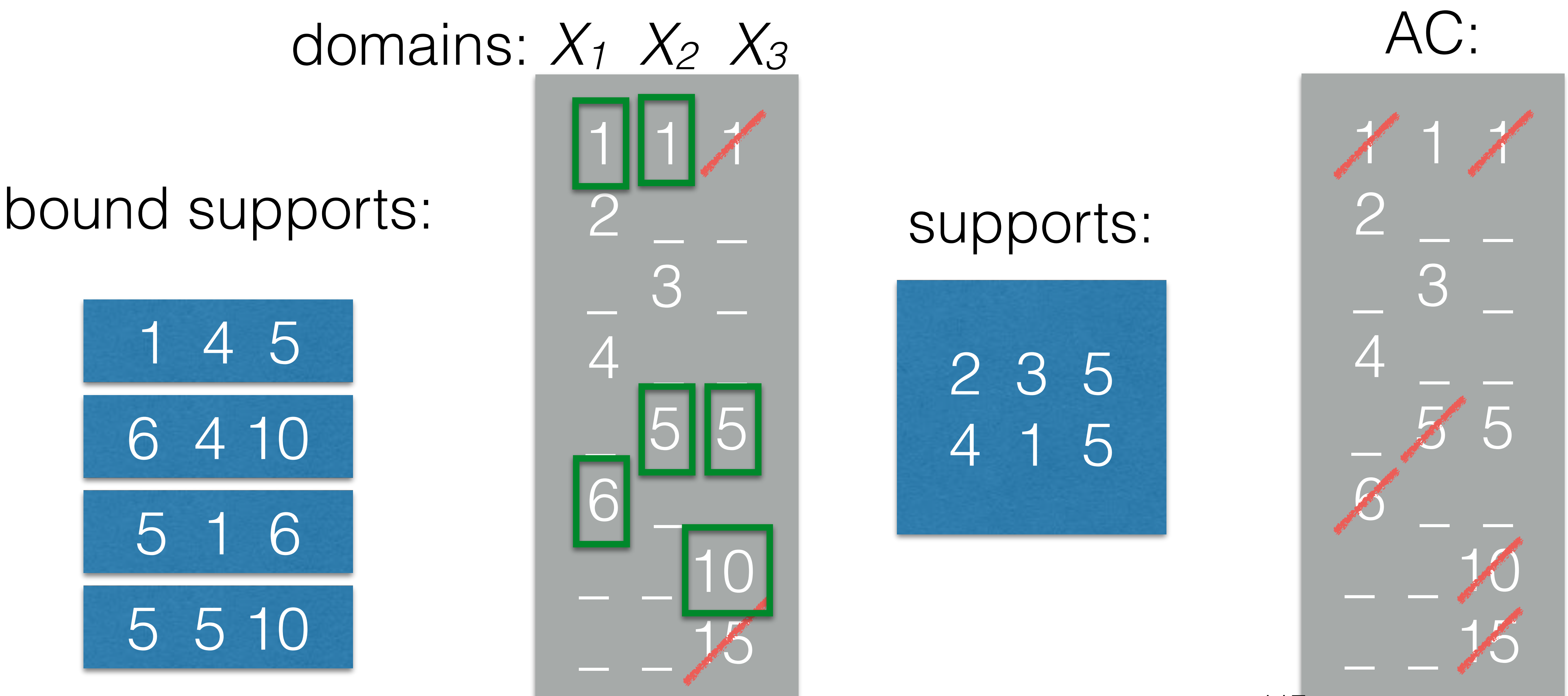
| | | |
|---|---|---|
| 2 | 3 | 5 |
| 4 | 1 | 5 |

AC:

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 2 | _ | _ |
| _ | 3 | _ |
| 4 | _ | _ |
| _ | 5 | 5 |
| 6 | _ | _ |
| _ | _ | 10 |
| _ | _ | 15 |

# Bound consistency

- $\boldsymbol{X_1 + X_2 = X_3}$
- $D(X_1)=\{1,2,4,6\}$, $D(X_2)=\{1,3,5\}$, $D(X_3)=\{1,5,10,15\}$



# BC on Case 1

- BC is polynomial on
    - `NValue`($X_1 \ldots X_n, N$),
    - `Sum`($X_1 \ldots X_n, N$),
    - etc.

# Supports and Bound Supports

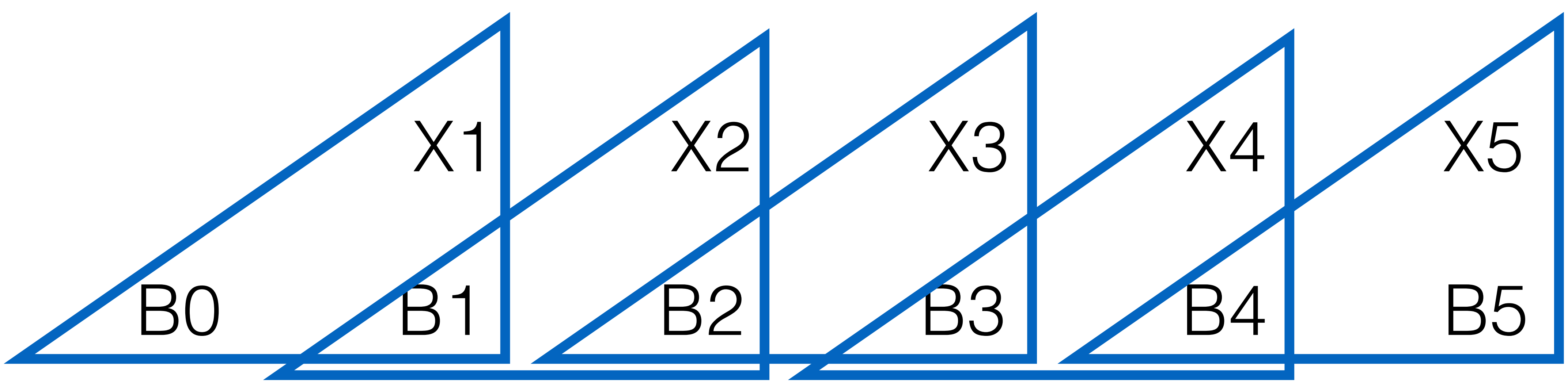
| | support | bound support |
|---|---|---|
| **all values** | AC | Range Consistency |
| **min and max** | BC(D) | BC |

# Case 2: AC-decomposable

**Definition 18** *The* incidence graph *of a network $N = (X, D, C)$ is the graph $G_N^I = (X \cup C, E)$ with $E = \{(x, c) \mid c \in C, x \in X(c)\}$. The network $N$ is Berge-acyclic iff $G_N^I$ is acyclic.*

**Theorem 3** *If a network $N$ is Berge-acyclic,* all *values in $AC(N)$ belong to a solution.*

**Corollary 2** *If a global constraint has a decomposition that is Berge-acyclic, then this is an AC-decomposition.*



# Warning!!

- `Sum`($X_1 \ldots X_n$,N):

| | | | | |
|---|---|---|---|---|
| $X_1$ | …. | $X_i$ | …. | $X_n$ |

# Warning!!

- `Sum`($X_1 \ldots X_n$,N):

$$\boxed{X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n}$$

- Decomposition:

$$X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n$$

$$B_0 \quad B_1 \ldots B_{i-1} \quad B_i \quad \ldots. \quad B_n$$

# Warning!!

- $\texttt{Sum}(X_1 \ldots X_n, N)$:

$$\boxed{X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n}$$

- Decomposition:

$X_1 \quad \ldots. \quad X_i \quad \ldots. \quad X_n$

$B_0 \quad B_1 \ \ldots. B_{i-1} \quad B_i \quad \ldots. \ B_n$

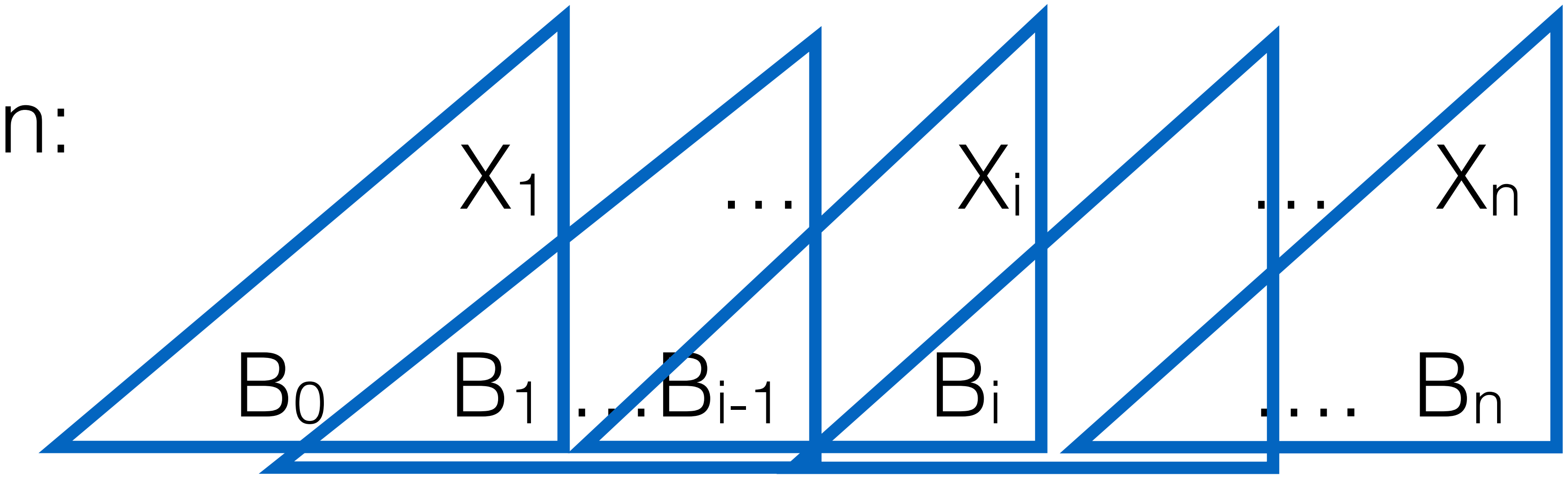
$$B_i = B_{i-1} + X_i$$

# Warning!!

- `Sum`($X_1 \ldots X_n$,N):

| | | | | |
|---|---|---|---|---|
| $X_1$ | …. | $X_i$ | …. | $X_n$ |

- Decomposition:

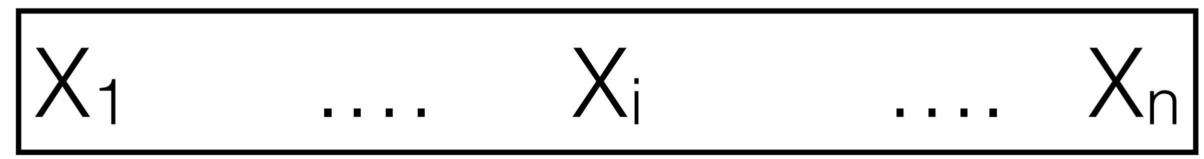



$B_i = B_{i-1} + X_i$

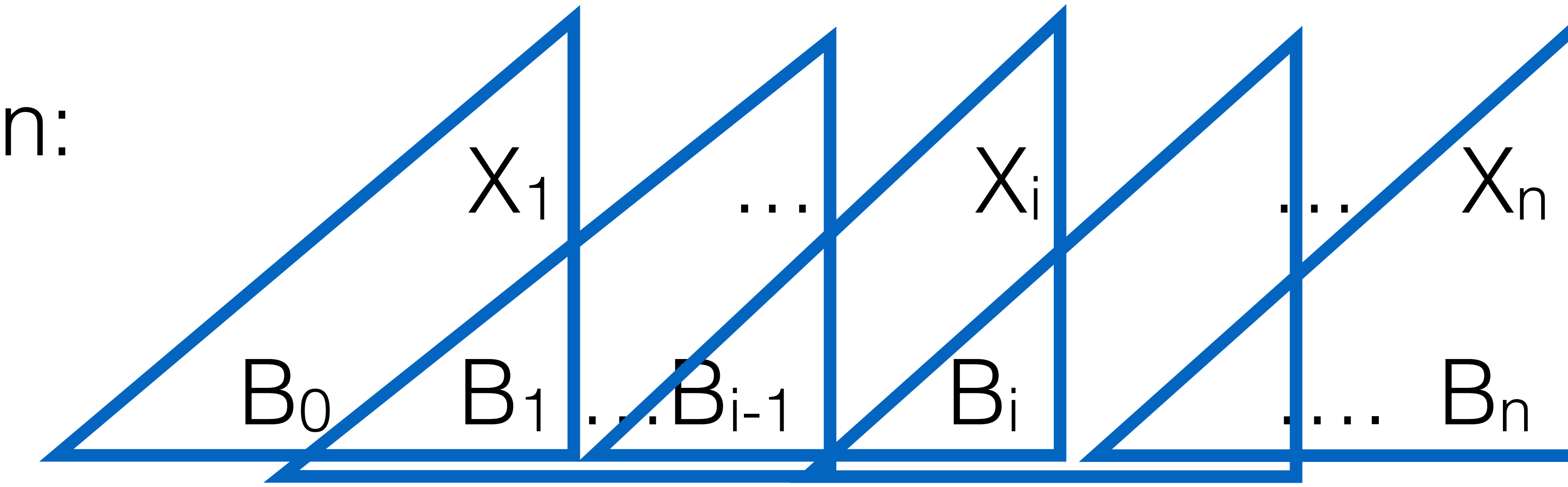
$B_0 = 0,\ B_n = N$

# Warning!!

- `Sum`($X_1 \ldots X_n$,N):




- Decomposition:




$B_i = B_{i-1} + X_i$

$B_0 = 0,\ B_n = N$

D($B_i$)=???? —> exponential size!

# The Picture

- Global constraints can be:
  - Type 1: We proved NP-hardness of arc consistency
  - Type 2: We found a decomposition preserving arc consistency
  - Is there a third type? ………….YES!

# Circuit Complexity

- If there exists a decomposition that preserves arc consistency on a global constraint *c*

  ⇒ ∃ a poly-size CNF formula that decides existence of support on *c*

  ⇒ ∃ a poly-size monotone Boolean circuit that decides existence of support on *c*

**Theorem 4** *If there does not exist a poly-size monotone circuit for a Boolean function $G$ then there does not exist any CNF decomposition computing $AC$ on $G$ and there does not exist any $AC$-decomposition for $G$.*

# Case 3: polynomial and non AC-decomposable

- Some Boolean functions (such as ‘’perfect matching’’) cannot be represented as a monotone Boolean circuit [Razborov 1985]

- `Alldifferent` subsumes perfect matching

  ➜ **no decomposition preserves arc consistency on `Alldifferent`**

  ➜ **no SAT formula for arc consistency on `Alldifferent`**

- Other popular constraints: `global-cardinality, same`

# Sports League Scheduling

- In 1995, ILP was not able to solve the NHL problem with more than 12 teams (NHL involves 30 teams)
- CP solved it up to 60 teams

# Pigeon Hole

- In 2018, SAT was not able to solve it with more than 20 pigeons (no polynomial-size proof)
- CP solves it in milliseconds

# Summary on global constraints

- NP-hard to enforce AC ➜ **weaker consistency**
- AC-decomposable ➜ **decompose!**
- polynomial to enforce AC but no AC-decomposition ➜ **build an ad hoc algorithm**

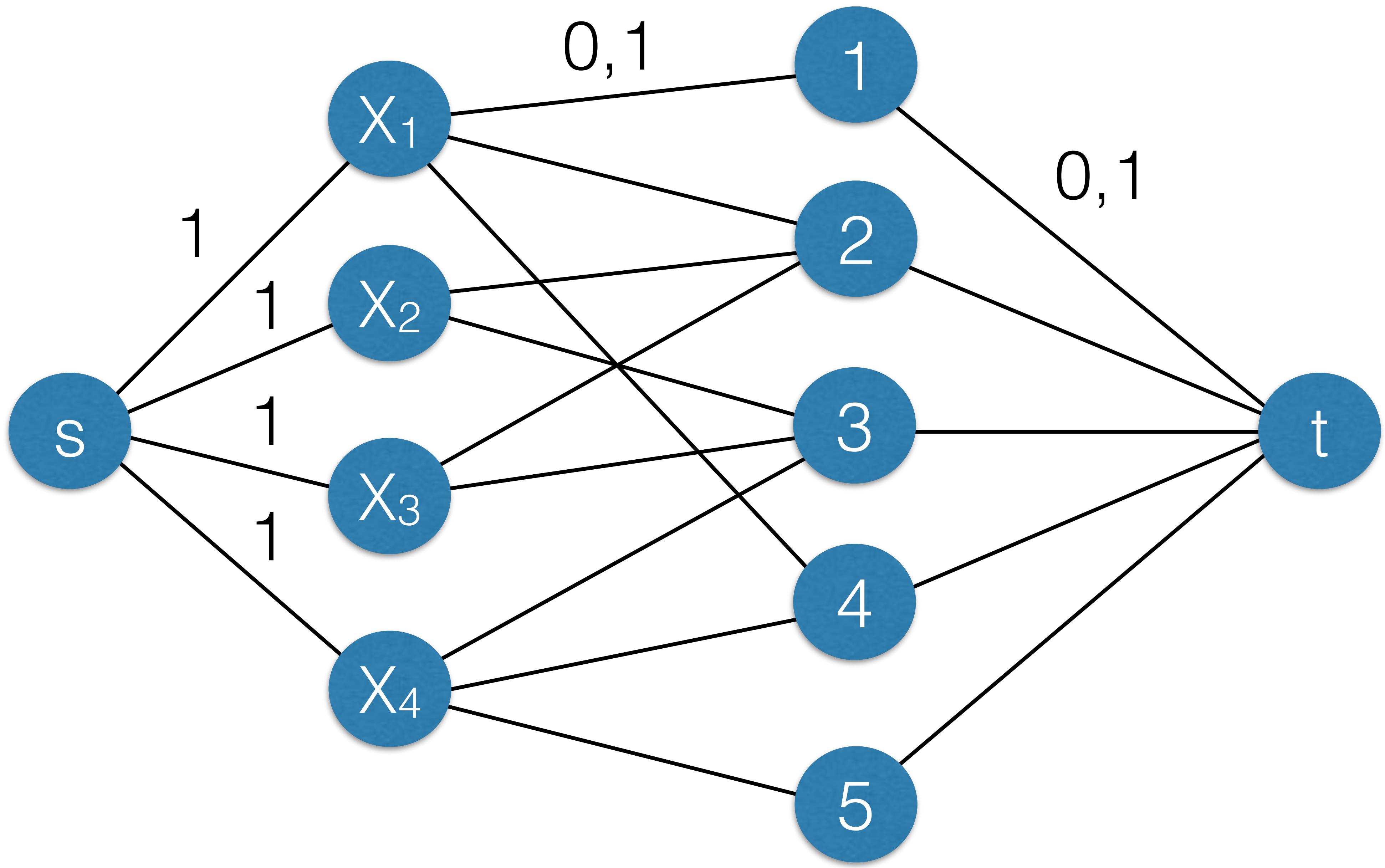
# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

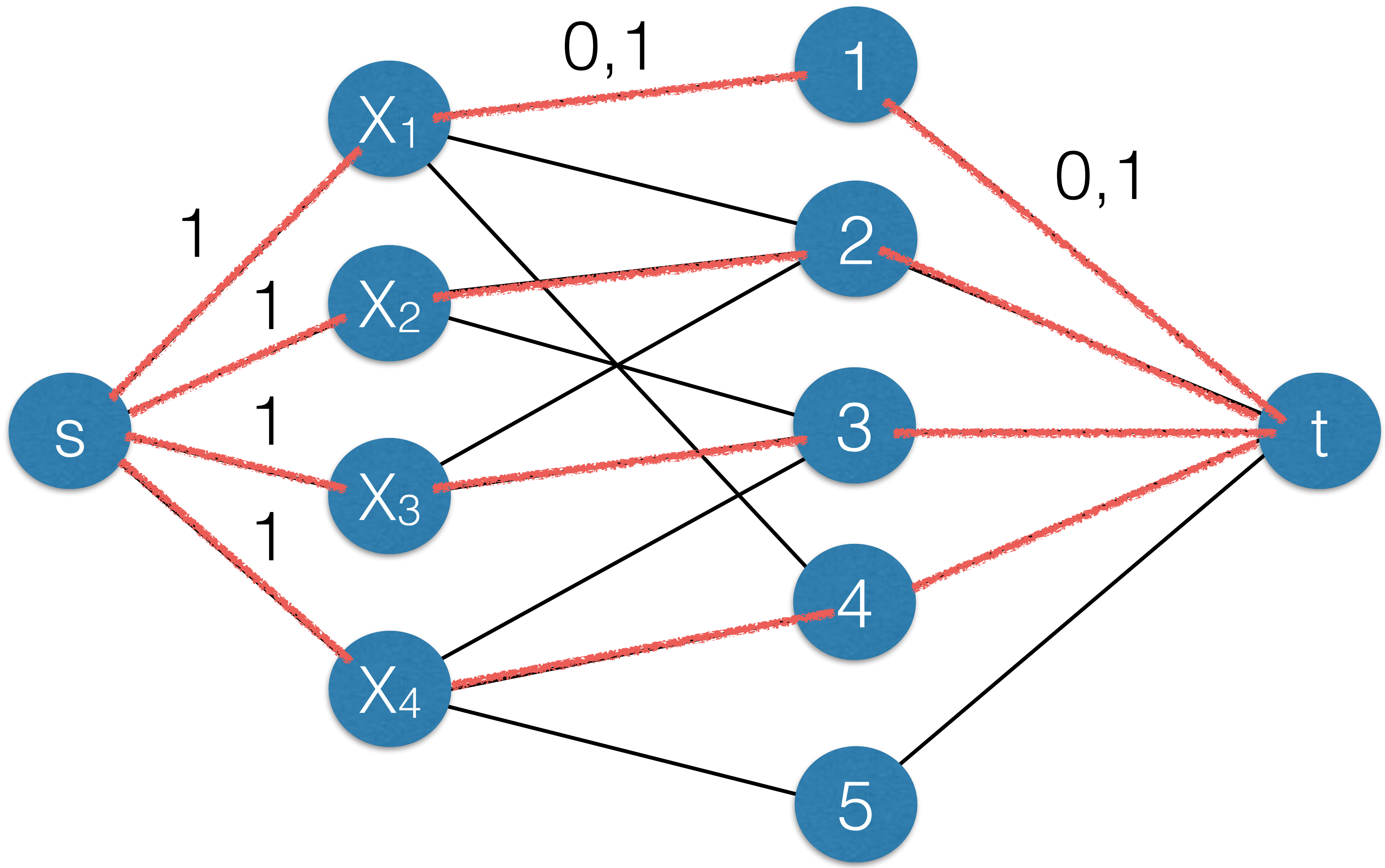
value graph



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

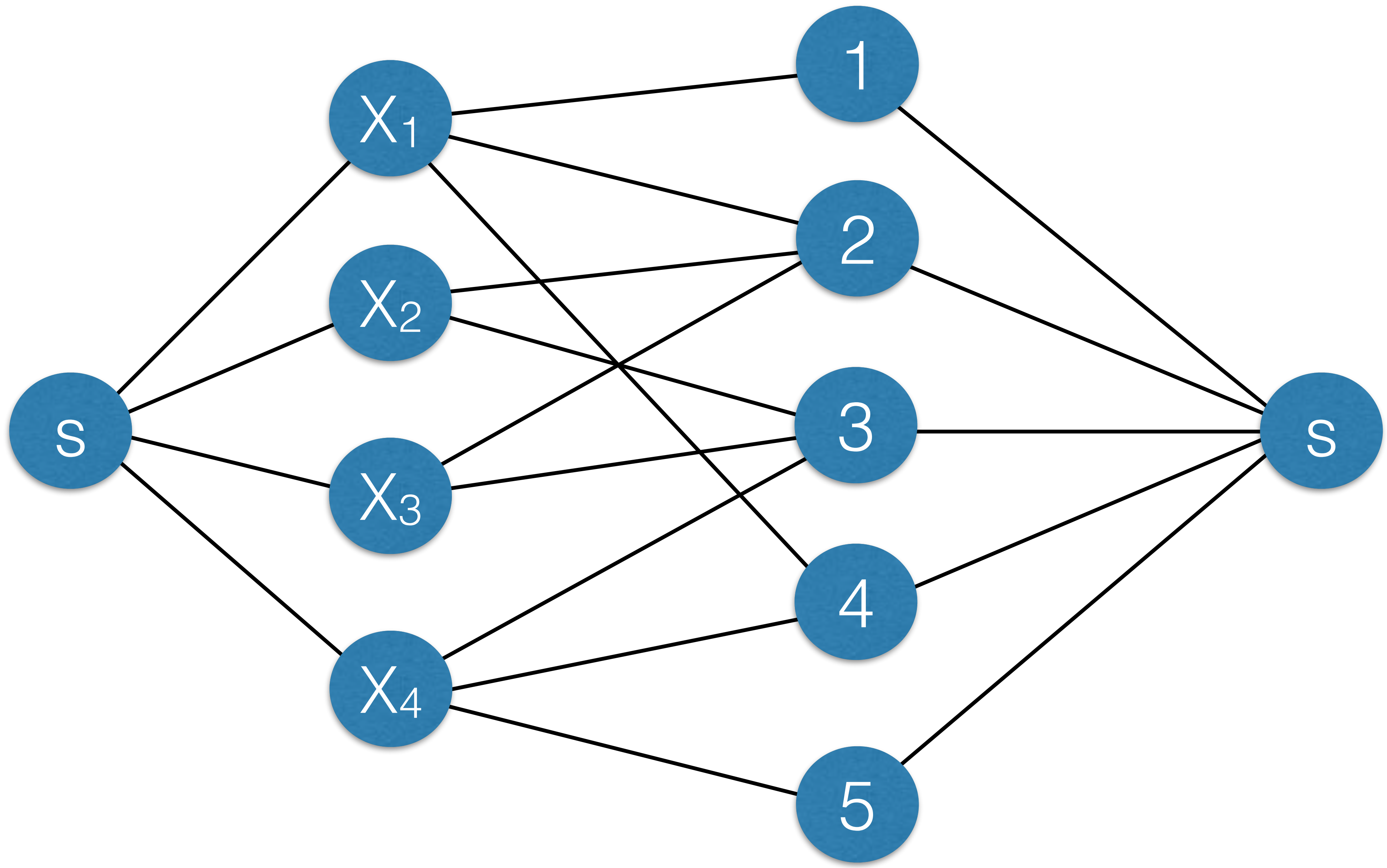
max flow



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

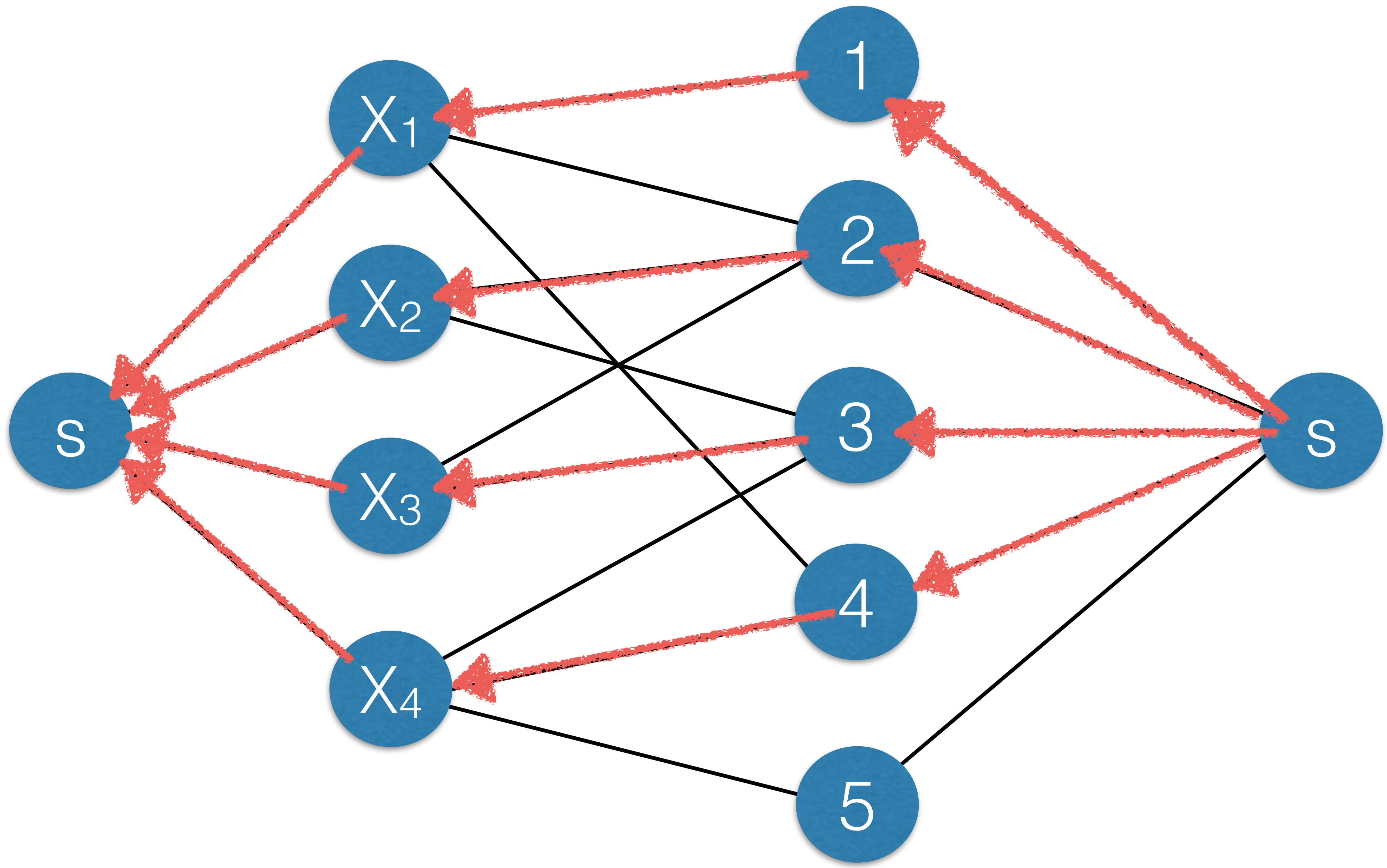
residual graph



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

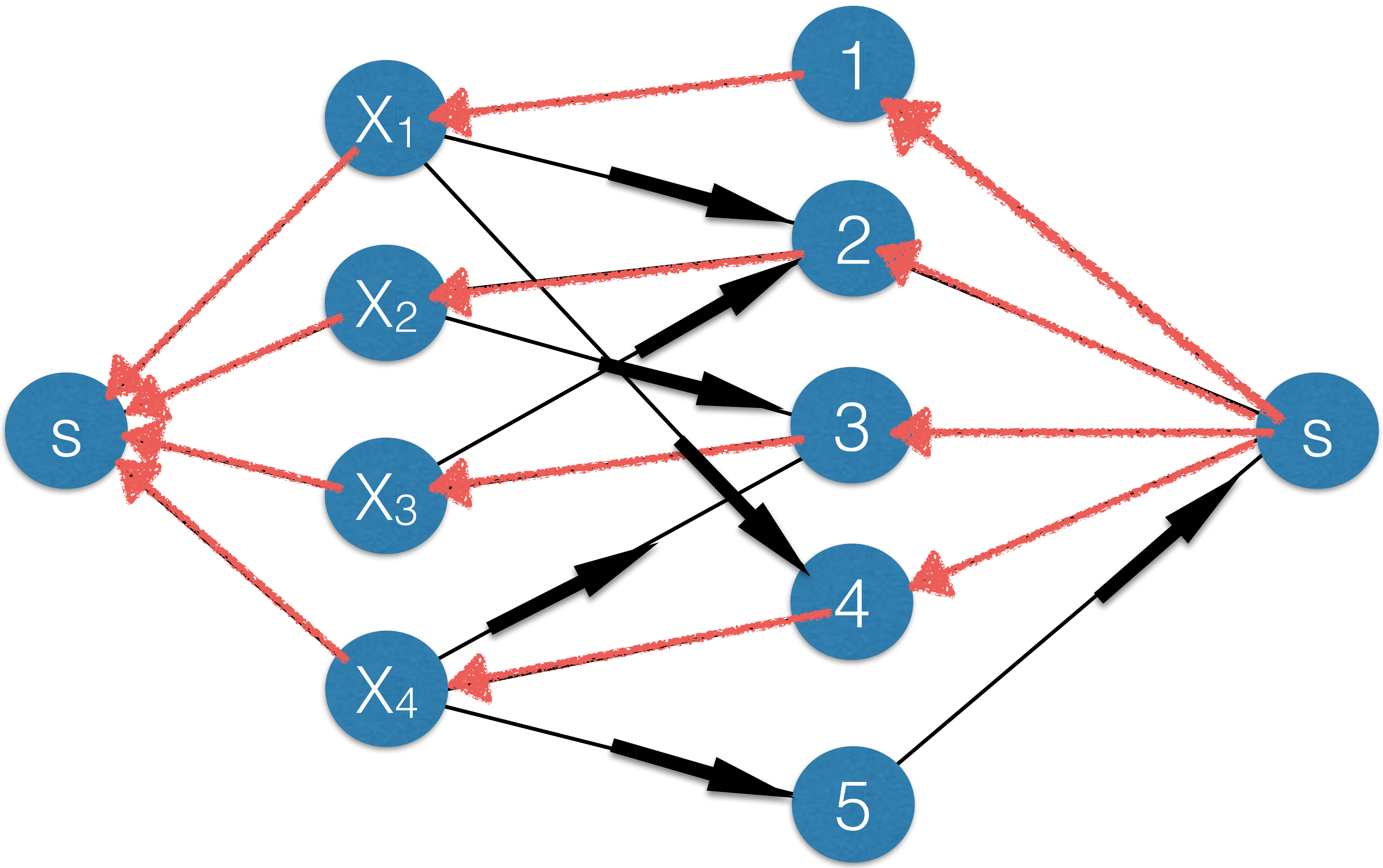
residual graph



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

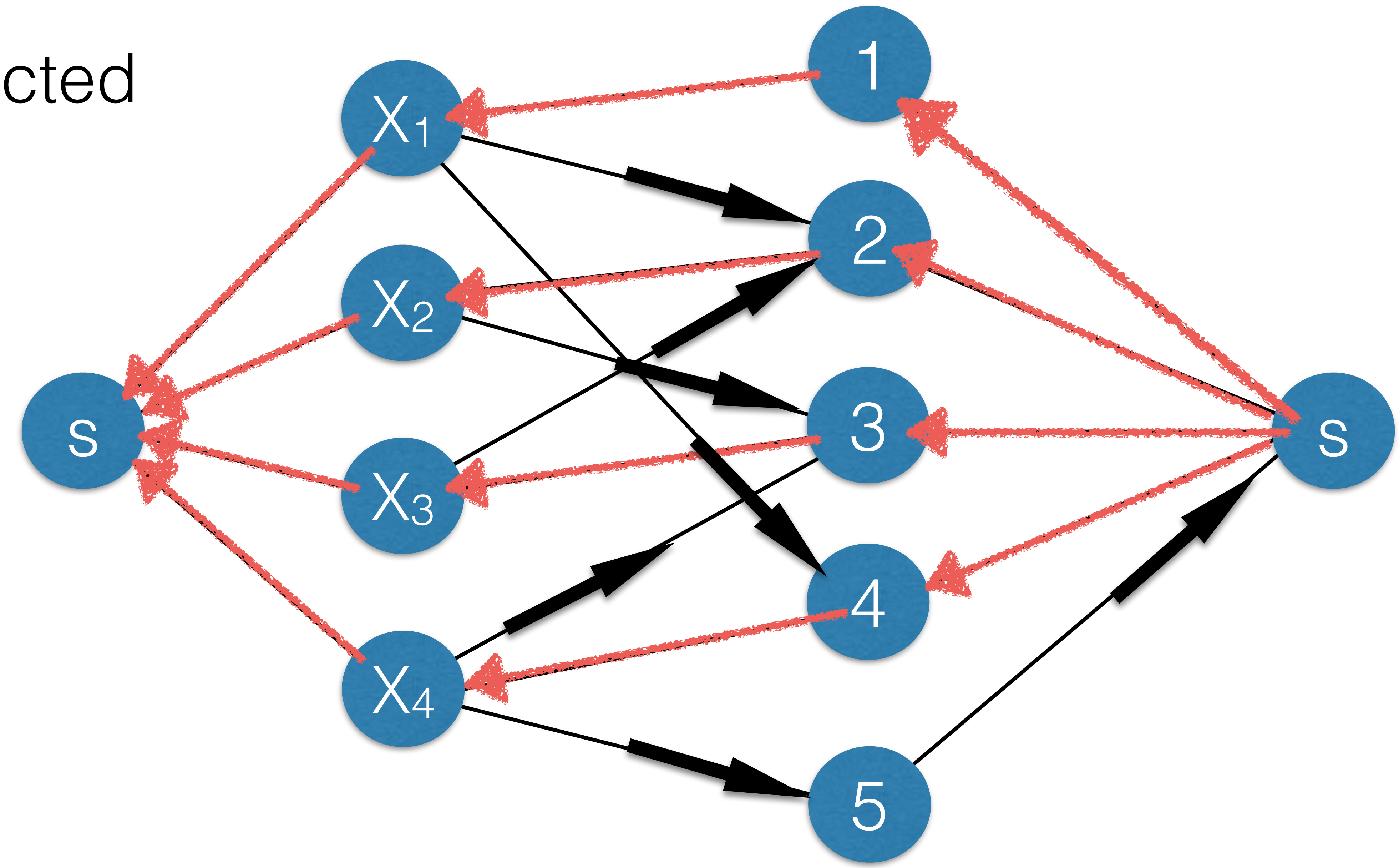
residual graph



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

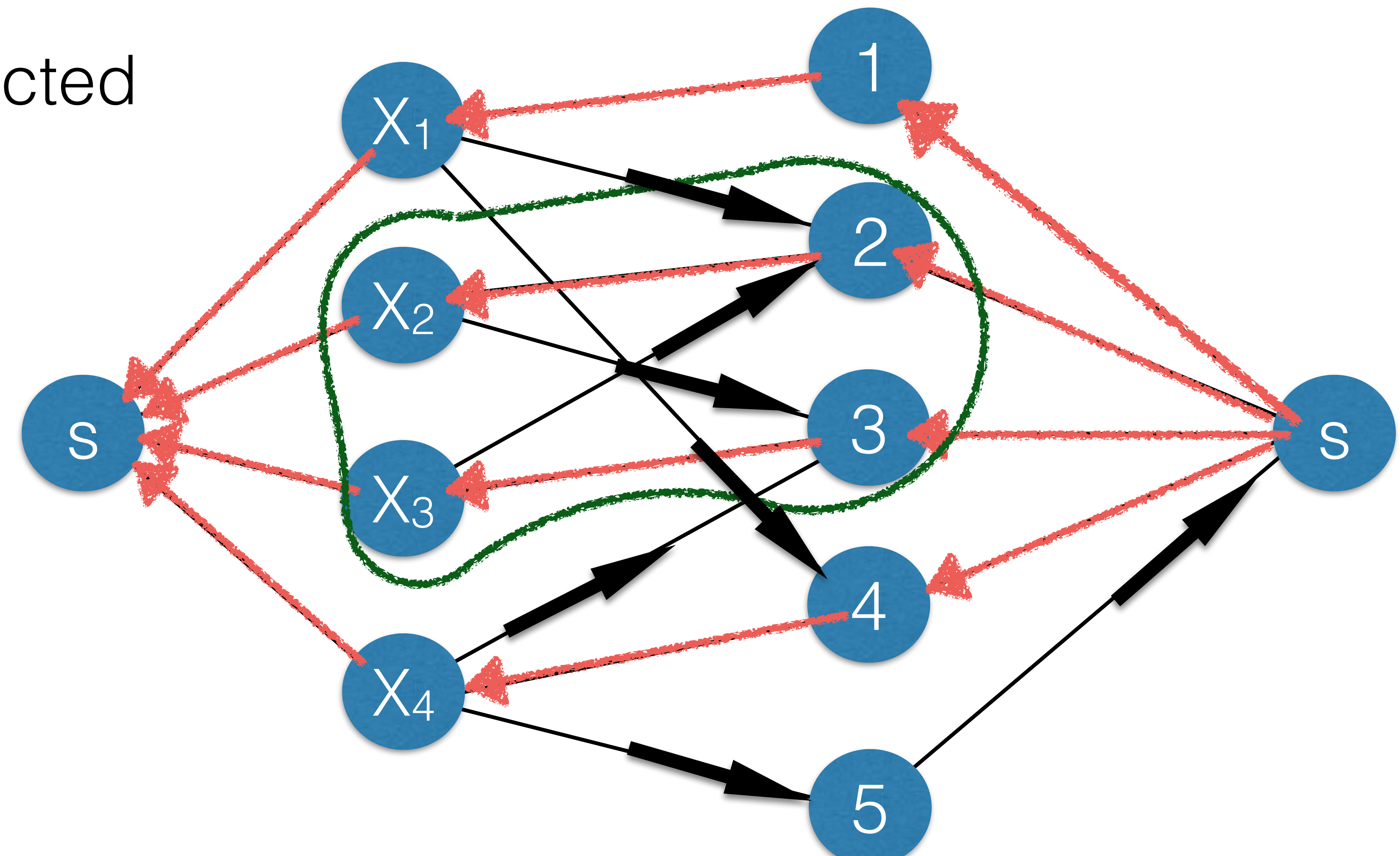
strongly connected components



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

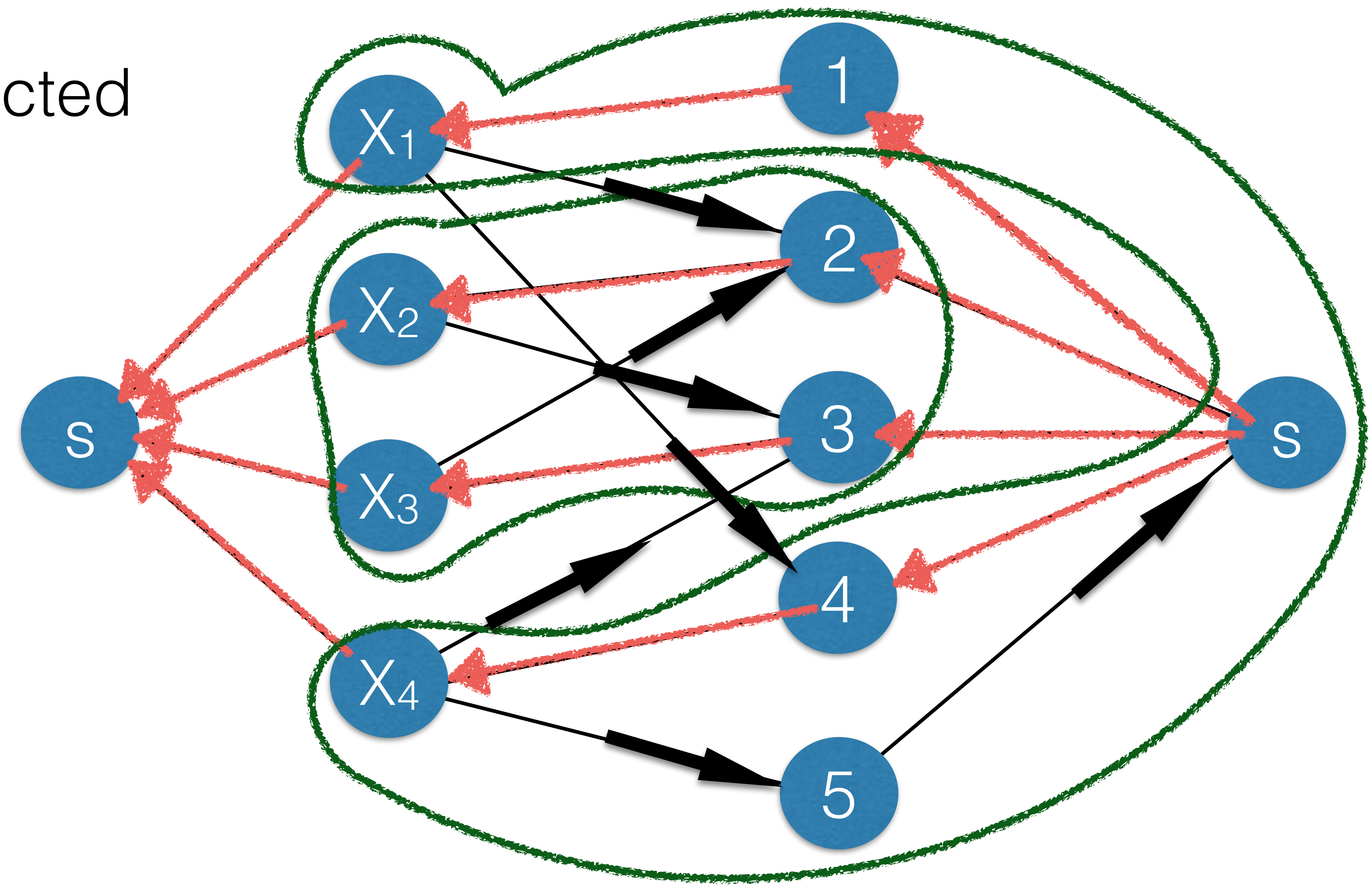
strongly connected components



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
| --- | --- | --- | --- |
| 1 | | | |
| 2 | 2 | 2 | |
| | 3 | 3 | 3 |
| 4 | | | 4 |
| | | | 5 |

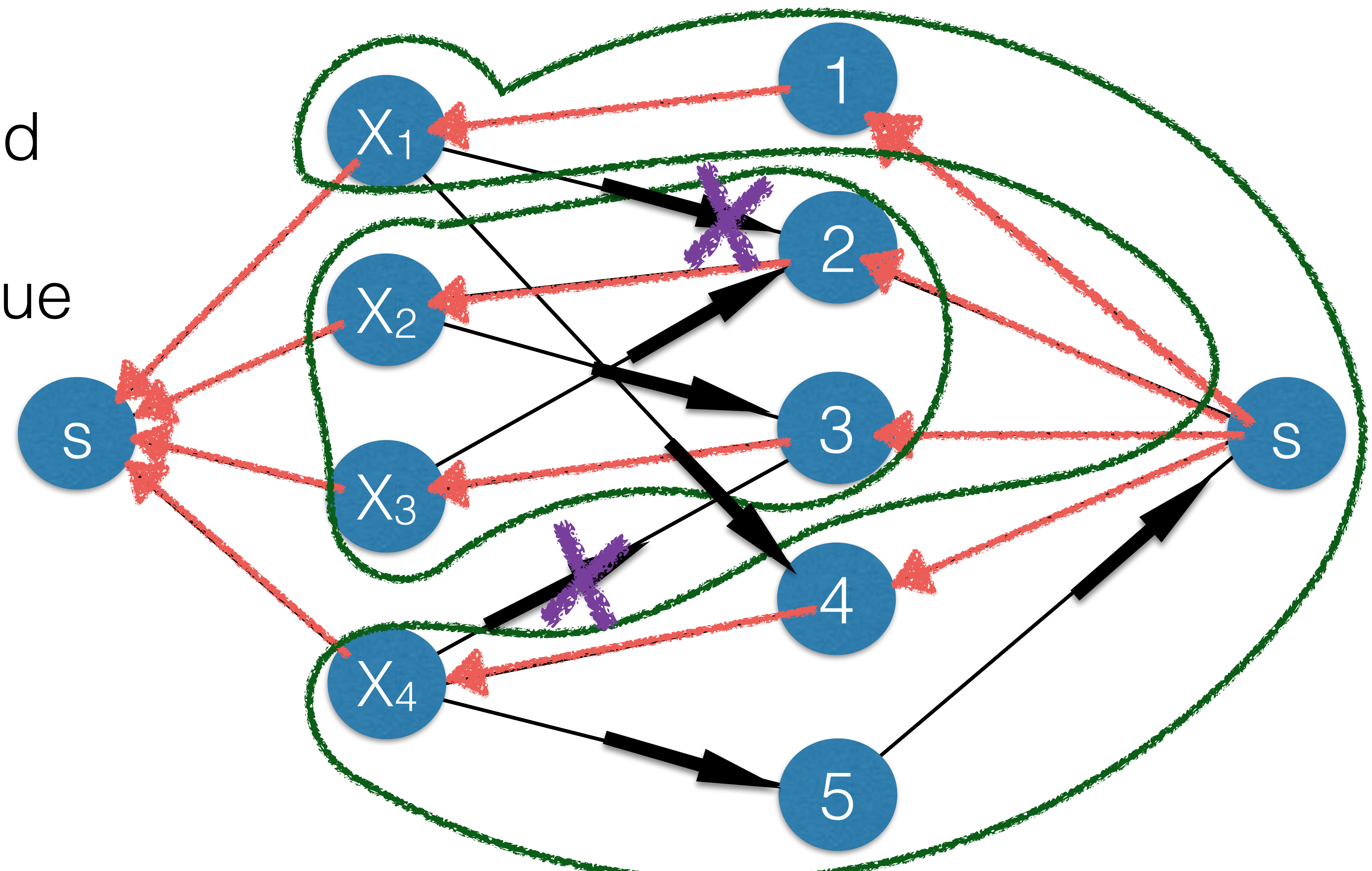
strongly connected components



# AC on Alldifferent

| X1 | X2 | X3 | X4 |
|---|---|---|---|
| 1 | | | |
| ~~2~~ | 2 | 2 | |
| | 3 | 3 | ~~3~~ |
| 4 | | | 4 |
| | | | 5 |

remove isolated edges and the associated value



# The (Big) Picture

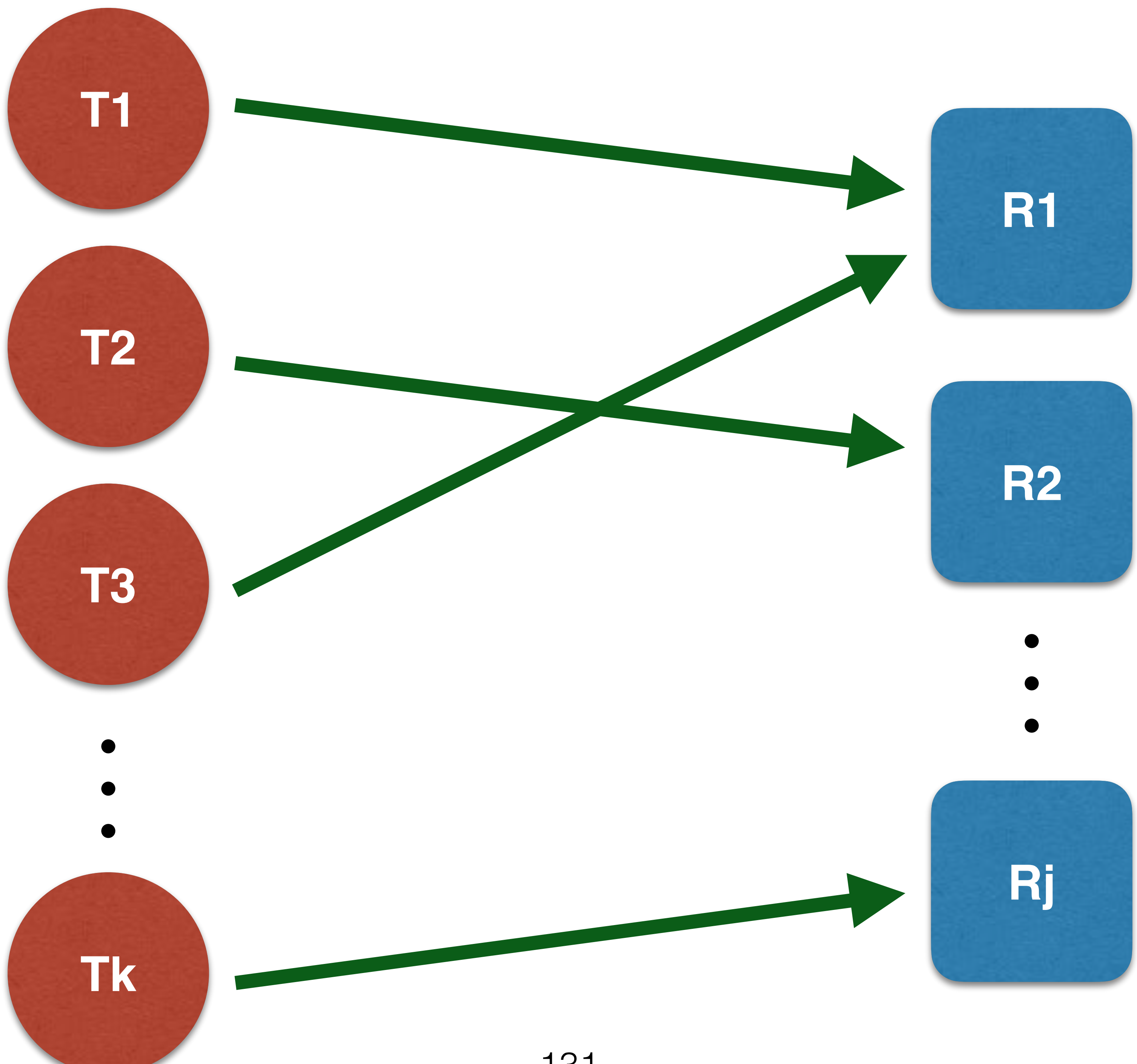
- CP reasoning cannot be simulated by SAT solvers when non-decomposable properties occur

  ★ Pigeon hole is an **`Alldifferent`** constraint!

# Guess efficiency of CP at a glance?

Tasks

flights
courses
hospital
factory
…




Resources

hostesses
teachers
nurses
machines
…