Constraint Programming

Lecture 7:
Exploiting Properties of Constraints
Last Lecture

• **MAC**
  - Combines search and enforcing global arc consistency.
  - **Incremental**: Smart about which arcs are added to the queue following an assignment.

• **AC6**
  - **Fine**-grained (smart about individual domain elements).
  - Lazy: Find **first** support for each domain element.

• **AC3.1/2001**
  - **Coarse**-grained (smart about which arcs to revise).
  - Smart about arc revision.
  - Same time/space complexity as AC6.
This Lecture

• Smarter Global Arc Consistency:
  • Reducing constraint checks.
  • Support inference, not support checking.
Exploiting Properties of Constraints

Functional Constraints
**Functional Constraints**

- Example: $x_1 = x_2$. $D_1, D_2 = \{1, 2, 3\}$.
- How many values in $D_2$ support $x_1 = 1$?
- How many values *could* support $x_1 = 1$ given this constraint?
Functional Constraints

- Example: $x_1 = x_2$. $D_1, D_2 = \{1, 2, 3\}$.
- How many values in $D_2$ support $x_1 = 1$? 1
- How many values *could* support $x_1 = 1$ given this constraint?
Functional Constraints

- Example: $x_1 = x_2$. $D_1, D_2 = \{1, 2, 3\}$.
- How many values in $D_2$ support $x_1 = 1$? 1
- How many values could support $x_1 = 1$ given this constraint? 1
Functional Constraints

- Constraint $c(x_i, x_j)$ is functional wrt $D_i$:
  - For all $d_i$ in $D_i$ there exists at most one $d_j$ in $D_j$ such that $c(x_i = d_i, x_j = d_j)$ is satisfied.

- Examples:
  - $x = y$,
  - $x = y + c$.
  - $x = y - c$.
  - $x = y \times c$.
  - Division? Careful if it’s integer division.
Functional Constraints

• Constraint \( c(x_i, x_j) \) is functional wrt \( D_i \):
  • Forall \( d_i \) in \( D_i \) there exists at most one \( d_j \) in \( D_j \) such that \( c(x_i = d_i, x_j = d_j) \) is satisfied.

• What about \( x = y^2 \)?
  • When is this functional wrt \( D_x \)?
  • When is it not?
Functional Constraints

- Constraint $c(x_i, x_j)$ is **functional** wrt $D_i$:
  - For all $d_i$ in $D_i$ there exists at most one $d_j$ in $D_j$ such that $c(x_i = d_i, x_j = d_j)$ is satisfied.

- What about $x = y^2$?
  - When is this functional wrt $D_x$?
    - When $y \geq 0$, or $y \leq 0$
    - When is it not?
      - If $D_y$ encompasses both +ve and –ve integers that are the roots of values in $D_x$.

- Illustrates that this property depends on the **domains**.
Revising Functional Constraints

• Ordinary arc revision for arc($x_i$, $x_j$):
  • For each element of $D_i$, check each element of $D_j$ in turn until a support is found.
  • Given a functional constraint?
Revising Functional Constraints

• Ordinary arc revision for $\text{arc}(x_i, x_j)$:
  • For each element of $D_i$, check each element of $D_j$ in turn until a support is found.
  • Given a functional constraint, use the function to see if the one possible supporting value is there or not.
    • **One** check.
Exploiting Properties of Constraints

Monotonic Constraints
Monotonic Constraints

• Example: $x_1 < x_2$. $D_1, D_2 = \{1, 2, 3\}$.

• Let’s say we are looking for support for $x_1 = 1$.

• What one check can we make, and why?
Monotonic Constraints

- Example: $x_1 < x_2$. $D_1, D_2 = \{1, 2, 3\}$.
- Let’s say we are looking for support for $x_1 = 1$.
- What one check can we make, and why?
  - Is the largest element in $D_2$ greater than 1?
  - Because if the largest element doesn’t support 1 then none of its elements can.
Monotonic Constraints

• Constraint \( c(x_i, x_j) \) is **monotonic** wrt domains \( D_i, D_j \):
  • For any \( d_i \) in \( D_i \), \( d_j \) in \( D_j \), if \( c(x_i = d_i, x_j = d_j) \) then \( c(x_i = d_i', x_j = d_j') \) for all:
    • \( d_i' \leq d_i \) in \( D_i \) and \( d_j' \geq d_j \) in \( D_j \)
    • I.e. If we decrease \( d_i \) and/or increase \( d_j \), constraint remains satisfied.

• Examples:
  • \( x_1 < x_2. \, D_1, D_2 = \{1, 2, 3\} \).
  • Orderings on vectors of variables, as we will see.
Revising Monotonic Constraints

• Ordinary arc revision for arc($x_i$, $x_j$):
  • For each element of $D_i$, check each element of $D_j$ in turn until a support is found.
  • Given a monotonic constraint?
Revising Monotonic Constraints

• Ordinary arc revision for arc($x_i$, $x_j$):
  • For each element of $D_i$, check each element of $D_j$ in turn until a support is found.
  • Given a monotonic constraint, check against the maximum value of $D_j$. If that doesn’t support it, nothing in $D_j$ will (and vice versa).
    • One check.
AC5 - Historical Note

• A “generic” arc consistency algorithm.
  • Can be instantiated to AC3 or AC4.
  • Can also be specialised to sub-classes of CSP.
    • As originally presented, this was functional and monotonic constraints.
Exploiting Properties of Constraints

Support Inference
The Story So Far

• We’ve seen two cases where support (or the lack of it) can be established with a **single** constraint check.

• Now we’re going to see how to **infer** support rather than check for it.
  • Assumption: inferences made are **cheaper** than constraint checks.
Support Inference Example

• \( x_1 \neq x_2 \). \( D_1, D_2 = \{1, 2, 3\} \).

\[ D_1 \begin{array}{c}
1 \\
2 \\
3 
\end{array} \]

\[ D_2 \begin{array}{c}
1 \\
2 \\
3 
\end{array} \]

• This example adapted from Bessiere et al “Using Inference to Reduce Arc Consistency Computation”, *IJCAI* 1995.
• Perform the AC6 initialisation step on this example, count the number of constraint checks made.
Support Inference Example: AC6 Initialise

- Look in $D_2$ for first support for 1 in $D_1$: 2

- Constraint checks so far: 2
Support Inference Example: AC6 Initialise

- Look in $D_2$ for first support for 2 in $D_1$: 1

- Constraint checks so far: 3
Support Inference Example: AC6 Initialise

- Look in $D_2$ for first support for 3 in $D_1$: 1

- Constraint checks so far: 4
Support Inference Example: AC6 Initialise

- Similarly:

\[ \begin{align*}
    D_1 & \quad 1 & \quad 2 & \quad 3 \\
    1 & \quad \quad & \quad 2 & \quad \quad & \quad 3 \\
    D_2 & \quad 1 & \quad 2 & \quad 3
\end{align*} \]

- Constraint checks total: 8
Bi-directionality: Recap

• $d_i$ in $D_i$ is supported by $d_j$ in $D_j$ if and only if $d_j$ in $D_j$ is supported by $d_i$ in $D_i$.
• Example: $x_1 = x_2$: $D_1$, $D_2 = \{1, 2, 3\}$.
  • 1 in $D_1$ supports 1 in $D_2$.
  • 1 in $D_2$ supports 1 in $D_1$.
• We do not need to know anything about the semantics of the constraints to use this property.
Support Inference Example: Bi-directionality

• Perform the AC6 initialisation step on this example, using bi-directionality inference. Count the number of constraint checks made.

\[ D_1 \quad 1 \quad 2 \quad 3 \]

\[ D_2 \quad 1 \quad 2 \quad 3 \]
Support Inference Example: Bi-directionality

• Look in $D_2$ for first support for 1 in $D_1$.
  • 1 in $D_2$ does not support 1 in $D_1$.
  • So: 1 in $D_1$ does not support 1 in $D_2$.

$D_1$ 1  2  3

$D_2$ 1  2  3

• Constraint checks so far: 1
Support Inference Example: Bi-directionality

- Look in $D_2$ for first support for 1 in $D_1$.
  - 2 in $D_2$ supports 1 in $D_1$.
  - So: 1 in $D_1$ supports 2 in $D_2$.

- Constraint checks so far: 2
Support Inference Example: Bi-directionality

- Look in $D_2$ for first support for 2 in $D_1$.
  - 1 in $D_2$ supports 2 in $D_1$.
  - So: 2 in $D_1$ supports 1 in $D_2$.

- Constraint checks so far: 3
Support Inference Example: Bi-directionality

- Look in $D_2$ for first support for 3 in $D_1$.
  - 1 in $D_2$ supports 3 in $D_1$.
  - So: 3 in $D_1$ supports 1 in $D_2$.

- Constraint checks so far: 4
Support Inference Example: Bi-directionality

- Look in $D_1$ for first support for 3 in $D_2$.
  - 1 in $D_1$ supports 3 in $D_2$.
  - So: 3 in $D_2$ supports 1 in $D_1$

- Constraint checks total: 5
Historical Note: AC-inference Schema

• General schema for enforcing global arc consistency with support inference.
• Instantiate with particular types of inference.
• AC7: bi-directionality.
  • So AC7 is still general.
Support Inference Example: Commutativity

- \( c(x_i = d, x_j = d') \) is satisfied if and only if \( c(x_i = d', x_j = d) \) is satisfied.
- E.g. 1 \( \neq \) 2 and 2 \( \neq \) 1 both satisfy the constraint.
- Given a constraint with this property, support can be inferred efficiently.
Support Inference Example: Commutativity

- Perform the AC6 initialisation step on this example, using bi-directionality inference. Count the number of constraint checks made.

\[ D_1 \begin{array}{ccc}
1 & 2 & 3 \\
\end{array} \]

\[ D_2 \begin{array}{ccc}
1 & 2 & 3 \\
\end{array} \]
Support Inference Example: Commutativity

- Look in $D_2$ for first support for 1 in $D_1$.
  - 1 in $D_2$ does not support 1 in $D_1$.
  - So: 1 in $D_1$ does not support 1 in $D_2$ (bi-directionality/commutativity).

$D_1$ 1 2 3

$D_2$ 1 2 3

- Constraint checks so far: 1
Support Inference Example: Commutativity

• Look in $D_2$ for first support for 1 in $D_1$.
  • 2 in $D_2$ supports 1 in $D_1$.
  • So: 1 in $D_1$ supports 2 in $D_2$ (bi-directionality).

• Constraint checks so far: 2
Support Inference Example: Commutativity

- Look in $D_2$ for first support for 1 in $D_1$.
  - 2 in $D_2$ supports 1 in $D_1$.
  - So: 1 in $D_1$ supports 2 in $D_2$ (bi-directionality).
  - Since we know 1,2 is supported, so is 2,1 (commutativity).

- Constraint checks so far: 2
Support Inference Example: Commutativity

- Look in $D_2$ for first support for 3 in $D_1$.
  - 1 in $D_2$ supports 3 in $D_1$.
  - So: 3 in $D_1$ supports 1 in $D_2$ (bi-directionality).

![](diagram.png)

- Constraint checks so far: 3
Support Inference Example: Commutativity

- Look in $D_2$ for first support for 3 in $D_1$.
  - 1 in $D_2$ supports 3 in $D_1$.
  - So: 3 in $D_1$ supports 1 in $D_2$ (bi-directionality).
  - Since we know 3,1 is supported, so is 1,3 (commutativity).

- Constraint checks so far: 3
Support Inference Example: Irreflexivity

- \( c(x_i = d, x_j = d) \) is not satisfied.
- E.g. \( x_i \neq x_j, x_i < x_j \)
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- $x_1 \neq x_2$. $D_1, D_2 = \{1, 2, 3\}$.

\[
D_1 \begin{array}{ccc}
1 & 2 & 3 \\
\end{array}
\]

\[
D_2 \begin{array}{ccc}
1 & 2 & 3 \\
\end{array}
\]
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- \( x_1 \neq x_2. \ D_1, D_2 = \{1, 2, 3\} \).

\[
\begin{array}{ccc}
D_1 & & D_2 \\
1 & \rightarrow & 1 \\
\downarrow & & \downarrow \\
2 & \rightarrow & 2 \\
\downarrow & & \downarrow \\
3 & \rightarrow & 3
\end{array}
\]
- Infer \( c(x_1 = 1, x_2 = 1) \) is not satisfied. Similarly for \( c(x_1 = 2, x_2 = 2) \), \( c(x_1 = 3, x_2 = 3) \)
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- $x_1 \neq x_2. \ D_1, D_2 = \{1, 2, 3\}.$

2 in $D_2$ supports 1 in $D_1$, so 1 in $D_1$ supports 2 in $D_2$. 
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- \( x_1 \neq x_2 \). \( D_1, D_2 = \{1, 2, 3\} \).

\[
\begin{array}{c}
\text{\( D_1 \)} \\
1 \rightarrow 2 \\
1 \leftarrow 2 \\
1 \leftarrow 3 \\
3 \rightarrow 1 \\
3 \rightarrow 2
\end{array}
\]

- 1 in \( D_2 \) supports 2 in \( D_1 \), so 2 in \( D_1 \) supports 1 in \( D_2 \).
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- $x_1 \neq x_2$. $D_1, D_2 = \{1, 2, 3\}$.

- 1 in $D_2$ supports 3 in $D_1$, so 3 in $D_1$ supports 1 in $D_2$. 
Support Inference Example: Irreflexivity

- Use bi-directionality and irreflexivity, to establish support in this network.
- \( x_1 \neq x_2 \). \( D_1, D_2 = \{1, 2, 3\} \).

- 1 in \( D_1 \) supports 3 in \( D_2 \), so 3 in \( D_2 \) supports 1 in \( D_1 \)
- Constraint checks total: 4
Support Inference Example: Irreflexivity & Commutativity

- Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
- $x_1 \neq x_2$. $D_1, D_2 = \{1, 2, 3\}$. 

\[ D_1 \begin{array}{ccc} 1 & 2 & 3 \end{array} \]

\[ D_2 \begin{array}{ccc} 1 & 2 & 3 \end{array} \]
Support Inference Example: Irreflexivity & Commutativity

- Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
- \( x_1 \neq x_2 \). \( D_1, D_2 = \{1, 2, 3\} \).

\[
\begin{array}{ccc}
   D_1 & 1 & 2 & 3 \\
   D_2 & 1 & 2 & 3 \\
\end{array}
\]

- Infer \( c(x_1 = 1, x_2 = 1) \) is not satisfied (irreflexivity). Similarly for \( c(x_1 = 2, x_2 = 2) \) and \( c(x_1 = 3, x_2 = 3) \).
Support Inference Example: Irreflexivity & Commutativity

- Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
- $x_1 \neq x_2$. $D_1, D_2 = \{1, 2, 3\}$.

- 2 in $D_2$ supports 1 in $D_1$, so 1 in $D_1$ supports 2 in $D_2$ (bi-directionality).
Support Inference Example: Irreflexivity & Commutativity

• Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
• \( x_1 \neq x_2. \ D_1, D_2 = \{1, 2, 3\} \).

\[
\begin{align*}
D_1 & \quad 1 \quad 2 \quad 3 \\
D_2 & \quad 1 \quad 2 \quad 3
\end{align*}
\]

• Since we know 1,2 is supported, so is 2,1 (commutativity).
Support Inference Example: Irreflexivity & Commutativity

- Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
- $x_1 \neq x_2$. $D_1, D_2 = \{1, 2, 3\}$.

\[ \begin{align*}
D_1 & \quad 1 \quad 2 \quad 3 \\
D_2 & \quad 1 \quad 2 \quad 3 \\
\end{align*} \]

- 1 in $D_2$ supports 3 in $D_1$, so: 3 in $D_1$ supports 1 in $D_2$ (bi-directionality).
Support Inference Example: Irreflexivity & Commutativity

• Use bi-directionality, commutativity and irreflexivity, to establish support in this network.
• \( x_1 \neq x_2. \ D_1, D_2 = \{1, 2, 3\}. \)

\[
\begin{array}{c}
D_1 \\
1 \\
2 \\
3 \\
\end{array}
\quad
\begin{array}{c}
D_2 \\
1 \\
2 \\
3 \\
\end{array}
\]

• Since we know 3,1 is supported, so is 1,3 (commutativity).
• Total constraint checks: 2
Support Inference Example: Disequality

• As well as irreflexive, disequality is anti-functional:
  • Given \( x_1 \neq x_2 \) and \( d \) in \( D_1 \), there is only one domain element in \( D_2 \) that, when assigned to \( x_2 \), violates the constraint.

• How can we exploit this fact to reduce checks for \( x_1 \neq x_2 \)?
Support Inference Example: Disequality

- As well as irreflexive, disequality is **anti-functional**:
  - Given \( x_1 \neq x_2 \) and \( d \) in \( D_1 \), there is only one domain element in \( D_2 \) that, when assigned to \( x_2 \), violates the constraint.

- How can we exploit this fact to reduce checks for \( x_1 \neq x_2 \)?
  - If \( |D_2| > 1 \), there is support for every element of \( D_1 \) (and vice versa). **0 checks**.
Support Inference Example: Monotonicity

- We saw earlier how to reduce constraint checks by exploiting monotonicity.
- \( x_1 \leq x_2 \). \( D_1, D_2 = \{1, 2, 3\} \).
  - For each element of \( D_1 \), just check maximum element in \( D_2 \).
  - Let’s say we check \( x_1 = 3 \) and found support via \( x_2 = 3 \). What now can we infer?
Support Inference Example: Monotonicity

• We saw earlier how to reduce constraint checks by exploiting monotonicity.

• $x_1 \leq x_2$. $D_1, D_2 = \{1, 2, 3\}$.
  • For each element of $D_1$, just check maximum element in $D_2$.
  • Let’s say we check $x_1 = 3$ and found support via $x_2 = 3$. What now can we infer?
    • Support for 3 in $D_2$ (bi-directionality).
    • Support for 1, 2 in $D_1$ (monotonicity).
Relevance

• Exploiting properties of constraints is very common in today’s constraint solvers.
• This is because of the prevalence of specialised intensional propagators for individual non-binary constraints.
  • These propagators focus on one constraint type.
  • So they can exploit the properties of that constraint.
  • Eg: all-different, summation, various orderings…
• We will see detailed examples later.
Lecture 7: Summary

• Reducing constraint checks by using properties of constraints:
  • Functional constraints.
  • Monotonic constraints.

• Avoiding constraint checks by using inference:
  • Bi-directionality.
  • Commutativity.
  • Irreflexivity.
  • Anti-functional.
  • Monotonicity.