DM841

Discrete Optimization

Part II

Lecture 7

Constraint Propagation Algorithms

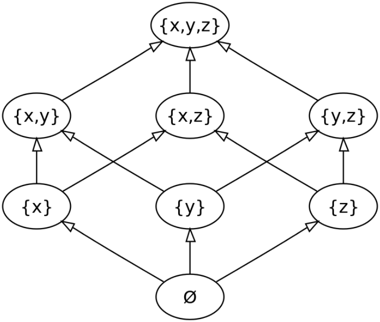
Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Orders

Domain-based tightening define a partial order (poset) because isomorphic to inclusion $\subseteq$, which is a partial order

(For a, b, elements of a poset P, if $a \leq b$ or $b \leq a$, then a and b are comparable. Otherwise they are incomparable)



Orders

Possible to define a partial order also on the local consistency property:

Definition

- Φ_1 is at least as strong as another Φ_2 if for any $\mathcal{P}$: $\Phi_1(\mathcal{P}) \leq \Phi_2(\mathcal{P})$:
  ie, $X_{\Phi_1(\mathcal{P})} = X_{\Phi_2(\mathcal{P})}, \quad \mathcal{DE}_{\Phi_1(\mathcal{P})} \subseteq \mathcal{DE}_{\Phi_2(\mathcal{P})}, \quad \mathcal{C}_{\Phi_1(\mathcal{P})} = \mathcal{C}_{\Phi_2(\mathcal{P})}$
  (any instantiation I on $Y \subseteq X_{\Phi_2(\mathcal{P})}$ locally inconsistent in $\Phi_2(\mathcal{P})$ is locally inconsistent in $\Phi_1(\mathcal{P})$)
- Φ_1 is stricly stronger than Φ_2 if it is at least as strong as and there exists a $\mathcal{P}$: $\Phi_1(\mathcal{P}) \leq \Phi_2(\mathcal{P})$.
- Φ_1 and Φ_2 are incomparable if there exists a $\mathcal{P}$' and $\mathcal{P}$'' such that $\Phi_1(\mathcal{P}') < \Phi_2(\mathcal{P}')$ and $\Phi_2(\mathcal{P}'') < \Phi_1(\mathcal{P}'')$.

Outline

1. Local Consistency

2. Arc Consistency Algorithms

Node Consistency

We call a CSP node consistent if for every variable x every unary constraint on x coincides with the domain of x.

Example

- $\langle \mathcal{C}, x_1 \geq 0, \ldots, x_n \geq 0; x_1 \in \mathbb{N}, \ldots, x_n \in \mathbb{N} \rangle$
  and $\mathcal{C}$ does not contain other unary constraints
  node consistent

- $\langle \mathcal{C}, x_1 \geq 0, \ldots, x_n \geq 0; x_1 \in \mathbb{N}, \ldots, x_n \in \mathbb{Z} \rangle$
  and $\mathcal{C}$ does not contain other unary constraints
  not node consistent

A CSP is node consistent iff it is closed under the applications of the Node Consistency rule (propagator):

$$\frac{\langle C; x \in D \rangle}{\langle C; x \in C \cap D \rangle}$$

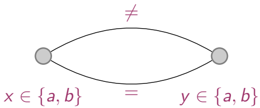
(the rule is parameterised by a variable x and a unary constraint C)

Arc Consistency

Arc consistency: every value in a domain is consistent with every binary constraint.

- $C = c(x, y)$ with $\mathcal{DE} = \{D(x), D(y)\}$ is arc consistent iff
  - $\forall a \in D(x)$ there exists $b \in D(y)$ such that $(a, b) \in C$
  - $\forall b \in D(y)$ there exists $a \in D(x)$ such that $(a, b) \in C$
- $\mathcal{P}$ is arc consistent iff it is AC for all its binary constraints

In general arc consistency does not imply global consistency.
An arc consistent but inconsistent CSP:



A consistent but not arc consistent CSP:

Arc Consistency

A CSP is arc consistent iff it is closed under the applications of the Arc Consistency rules (propagators):

$$\frac{\langle C; x \in D(x), y \in D(y)\rangle}{\langle C; x \in D'(x), y \in D(y)\rangle}$$

where $D'(x) := \{a \in D(x) \mid \exists b \in D(y), (a, b) \in C\}$

$$\frac{\langle C; x \in D(x), y \in D(y)\rangle}{\langle C; x \in D(x), y \in D'(y)\rangle}$$

where $D'(y) := \{b \in D(y) \mid \exists a \in D(x), (a, b) \in C\}$

Generalized Arc Consistency (GAC)

Given arbitrary (non-normalized, non-binary) $\mathcal{P}$, $C \in \mathcal{C}$, $x_i \in X(C)$

(Value) $v \in D(x_i)$ is consistent with C in $\mathcal{DE}$ iff $\exists$ a valid tuple τ for C: $v_i = \tau[x_i]$. τ is called support for (x_i, v_i)

(Variable) $\mathcal{DE}$ is GAC on C for x_i iff all values in $D(x_i)$ are consistent with C in $\mathcal{DE}$ (i.e., $D(x_i) \subseteq \pi_{\{x_i\}}(C \cap \pi_{\{X(C)\}}(\mathcal{DE}))$)

(Problem) $\mathcal{P}$ is GAC iff $\mathcal{DE}$ is GAC for all v in X on all $C \in \mathcal{C}$

$\mathcal{P}$ is arc inconsistent iff the only domain tighter than $\mathcal{DE}$ which is GAC for all variables on all constraints is the empty set.

(aka, hyperarc consistency, domain consistency)

Example

$\langle x = 1, y \in \{0,1\}, z \in \{0,1\}; \mathcal{C} = \{x \wedge y = z\}\rangle$
is hyperarc consistent

$\langle x \in \{0,1\}, y \in \{0,1\}, z \in \{0,1\}; \mathcal{C} = \{x \wedge y = z\}\rangle$
is not hyper-arc consistent

Example: arc consistency $\neq$ 2-consistency, AC $<$ 2C on non-normalized binary CSP, and incomparable on arbitrary CSP

Generalized Arc Consistency

A CSP is arc consistent iff it is closed under the applications of the Arc Consistency rules (propagators):

$$\frac{\langle C; x_1 \in D(x), \ldots, x_k \in D(x_k)\rangle}{\langle C; x_1 \in D(x_1), \ldots, x_{i-1} \in D(x_{i-1}), x_i \in D'(x_i), x_{i+1} \in D(x_{i+1}), \ldots, x_k \in D(x_k)\rangle}$$

where $D'(x_i) := \{a \in D(x_i) | \exists \tau \in C, a = \tau[x_i]\}$

References

Apt K.R. (2003). **Principles of Constraint Programming**. Cambridge University Press.

Barták R. (2001). **Theory and practice of constraint propagation**. In *Proceedings of CPDC2001 Workshop*, pp. 7–14. Gliwice.

Bessiere C. (2006). **Constraint propagation**. In *Handbook of Constraint Programming*, edited by F. Rossi, P. van Beek, and T. Walsh, chap. 3. Elsevier. Also as Technical Report LIRMM 06020, March 2006.

Resume

- Definitions
  (CSP, restrictions, projections, istantiation, local consistency)
- Tigthtenings
- Global consistent (any instantiation local consistent can be extended to a solution) needs exponential time
  $\rightsquigarrow$ local consistency defined by condition Φ of the problem
- Tightenings by constraint propagation: reduction rules + rules iterations
  - reduction rules $\Leftrightarrow$ Φ consistency
  - rules iteration: reach fixed point, that is, closure of all tightenings that are Φ consistent
- Domain-based Φ: (generalized) arc consistency

Outline

Arc Consistency

Arc Consistency rule 1 (propagator):

$$\frac{\langle C; x \in D(x), y \in D(y)\rangle}{\langle C; x \in D'(x), y \in D(y)\rangle}$$

where $D'(x) := \{a \in D(x) | \exists b \in D(y), (a, b) \in C\}$

This can also be written as ($\bowtie$ represents the join):

$$D(x) \leftarrow D(x) \cap \pi_{\{x\}}(\bowtie(C, D(y)))$$

Arc Consistency rule 2 (propagator):

$$\frac{\langle C; x \in D(x), y \in D(y)\rangle}{\langle C; x \in D(x), y \in D'(y)\rangle}$$

where $D'(y) := \{b \in D(y) | \exists a \in D(x), (a, b) \in C\}$

This can also be written as:

$$D(y) \leftarrow D(y) \cap \pi_{\{y\}}(\bowtie(C, D(x)))$$

Generalized Arc Consistency

(Generalized) Arc Consistency rule (propagator):

$$\frac{\langle C; x_1 \in D(x), \ldots, x_k \in D(x_k)\rangle}{\langle C; x_1 \in D(x_1), \ldots, x_{i-1} \in D(x_{i-1}), x_i \in D'(x_i), x_{i+1} \in D(x_{i+1}), \ldots, x_k \in D(x_k)\rangle}$$

where $D'(x_i) := \{a \in D(x_i) | \exists \tau \in C, a = \tau[x_i]\}$

This can also be written as:

$$D(x_i) \leftarrow D(x_i) \cap \pi_{\{x_i\}}(C \cap \pi_{X(C)}(\mathcal{DE}))$$

Exercise – Binary CSP

Theorem

Show how an arbitrary (non-binary) CSP can be polynomially converted into an equivalent binary CSP.

AC1 – Reduction rule

Revision: making a constraint arc consistent by propagating the domain from a variable to anohter
Corresponds to:

$$D(x) \leftarrow D(x) \cap \pi_{\{x\}}(\bowtie(C, D(y)))$$

for a given variable x and constraint C
Assume normalized network:

REVISE($(x_i), x_j$)

input: a subnetwork defined by two variables $X = \{x_i, x_j\}$, a distinguished variable x_i, domains: D_i and D_j, and constraint R_{ij}

output: D_i, such that, x_i arc-consistent relative to x_j

1. **for** each $a_i \in D_i$
2. **if** there is no $a_j \in D_j$ such that $(a_i, a_j) \in R_{ij}$
3. **then** delete a_i from D_i
4. **endif**
5. **endfor**

Complexity: $O(d^2)$ or $O(rd^r)$
d values, r arity

AC1 – Rules Iteration

Binary case

AC-1($\mathcal{R}$)

input: a network of constraints $\mathcal{R} = (X, D, C)$

output: $\mathcal{R}'$ which is the loosest arc-consistent network equivalent to $\mathcal{R}$

1. **repeat**
2. **for** every pair $\{x_i, x_j\}$ that participates in a constraint
3. Revise$((x_i), x_j)$ (or $D_i \leftarrow D_i \cap \pi_i(R_{ij} \bowtie D_j)$)
4. Revise$((x_j), x_i)$ (or $D_j \leftarrow D_j \cap \pi_j(R_{ij} \bowtie D_i)$)
5. **endfor**
6. **until** no domain is changed

- Complexity (Mackworth and Freuder, 1986): $O(end^3)$
  e number of arcs, n variables
  (ed^2 each loop, a single succesful removal causes all loop again $\rightsquigarrow$ nd number of loops)
- best-case $= O(ed)$
- Arc-consistency is at least $O(ed^2)$ in the worst case

AC3 (Macworth, 1977)

General case – Arc oriented (coarse-grained)

function Revise3(**in** x_i: variable; c: constraint): **Boolean** ;
begin
    CHANGE ← **false**;
    foreach $v_i \in D(x_i)$ **do**
        if $\nexists \tau \in c \cap \pi_{X(c)}(D)$ *with* $\tau[x_i] = v_i$ **then**
            remove v_i from $D(x_i)$;
            CHANGE ← **true**;
    return CHANGE ;
end

function AC3/GAC3(**in** X: set): **Boolean** ;
begin
    /* initalisation */;
    $Q \leftarrow \{(x_i, c) \mid c \in C, x_i \in X(c)\}$;
    /* propagation */;
    while $Q \neq \emptyset$ **do**
        select and remove (x_i, c) from Q;
        if Revise(x_i, c) **then**
            if $D(x_i) = \emptyset$ **then** return **false** ;
            else $Q \leftarrow Q \cup \{(x_j, c') \mid c' \in C \wedge c' \neq c \wedge x_i, x_j \in X(c') \wedge j \neq i\}$;
    return **true** ;
end

$O(er^3 d^{r+1})$ time
$O(er)$ space

AC3
Example

$$\mathcal{P} = \langle X = (x, y, z),\ \mathcal{DE} = \{D(x) = D(y) = \{1, 2, 3, 4\}, D(z) = \{3\}\},$$
$$\mathcal{C} = \{C_1 \equiv x \leq y, C_2 \equiv y \neq z\}\}\rangle$$