DM826 – Spring 2011

Modeling and Solving Constrained Optimization Problems

## Lecture 7

# Filtering Algorithms

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

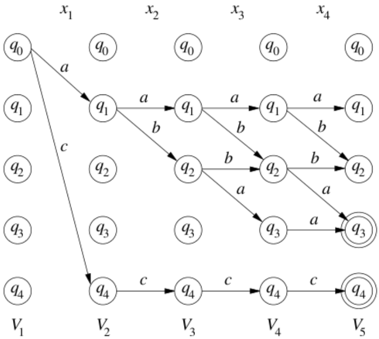
[*Based on slides by Stefano Gualandi, Politecnico di Milano*]

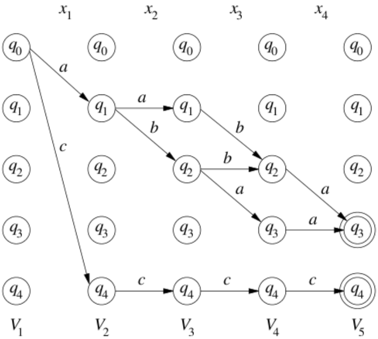
# Filtering

gcc

# Filtering

`regular`

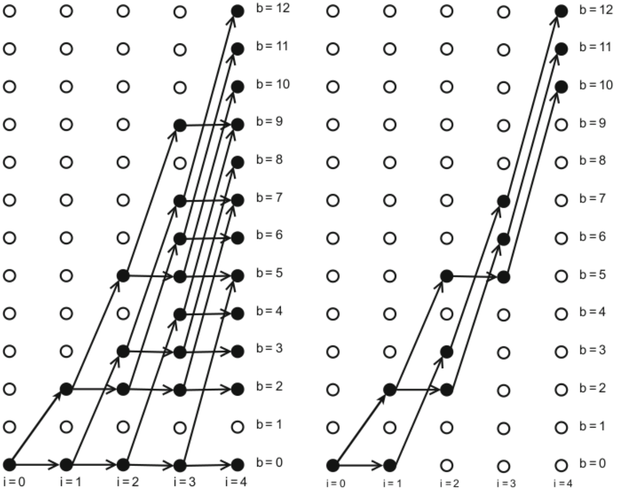


# Filtering

`subsetsum`

$10 \leq 2x_1 + 3x_2 + 4x_3 + 5x_4 \leq 12$



# Filtering



`element`

# Filtering Algorithm Design



1. Filtering algorithms based on a generic algorithm
   Simple "square" constraint using `element`:

   $$\texttt{element}(y, [2, 4, 8, 16, 32], x), x \in \{1, 2, 3, 4, 5\}$$

2. Filtering algorithms based on existing algorithms
   Reuse existing algorithms for filtering (e.g., flows algorithms, dynamic programming).

3. Filtering algorithms based on ad-hoc algorithms
   Pay particular attention to incrementality and amortized complexity

4. Filtering algorithms based on model reformulation
   See the Constraint Decomposition approach

# Outline

1. Soft Constraints

2. Optimization Constraints

# Soft Constraints

### Soft constraint

A *soft constraint* is a constraint that may be violated. We measure the violation of each constraint, and the goal is to minimize the total amount of violation of all soft-constraints.

### Definition

A *violation measure* for a soft-constraint $C(x_1, \dots, x_n)$ is a function

$$\mu : D(x_1) \times \cdots \times D(x_n) \to \mathbb{Q}.$$

This measure is represented by a *cost* variable $z$.

# Violation measures

- The variable-based violation measure $\mu_{var}$ counts the minimum number of variables that need to change their value in order to satisfy the constraint.

- The decomposition-based violation measure $\mu_{dec}$ counts the number of constraints in the binary decomposition that are violated.

# The `soft-alldifferent`

**Definition**

Let $x_1, x_2, \ldots, x_n, z$ be variables with respective finite domains $D(x_1), D(x_2), \ldots, D(x_n), D(z)$. Let $\mu$ be a violation measure for the `alldifferent` constraint. Then

$$\texttt{soft-alldifferent}(x_1, \ldots, x_n, z, \mu) = \{(d_1, \ldots, d_n, d) \mid \forall i. d_i \in D(x_i), d \in D(z), \mu(d_1, \ldots, d_n) \leq d\}$$

is the soft alldifferent constraint with respect to $\mu$.

# The soft-alldifferent: an example

**Example**

Consider the following CSP

$$x_1 \in \{a, b\}, x_2 \in \{a, b\}, x_3 \in \{a, b\}, x_4 \in \{a, b, c\}, z \in \mathbb{Z}^+$$
$$\texttt{soft-alldifferent}(x_1, x_2, x_3, x_4, \mu, z)$$
$$\min z$$

We have for instance $\mu_{var}(b, b, b, b) = 3$ and $\mu_{dec}(b, b, b, b) = 6$.

# Comet library of soft-constraints

| Constraints | Domain | Range | Bound |
|---|---|---|---|
| softAtLeast | X | | |
| softAtMost | X | | |
| softCardinality | X | | |
| softStretch | X | | |

You can simulate the soft-alldifferent using the softCardinality.

A nice feature of the softCardinality is that it gives access to the reduced cost of variable-value assignments.

# Outline

1. Soft Constraints

2. Optimization Constraints

# Optimization Constraints

Optimization Constraint bring the costs of variable-value pair into the declarative semantic of the constraints.

The filtering does take into account the cost, and a tuple may be inconsistent because it does not lead to a solution of "at least" a given cost.

# Optimization Constraints in Comet

| Constraints | |
|---|---|
| minAssignment | X |
| minCircuit | X |
| costRegular | X |
| binpackingLB | X |

# Filtering

soft_alldiff
global cardinality with costs

# Reduced-Cost Based Filtering [Focacci&al1999]

Definition

Let $X = \{x_1, ..., x_n\}$ be a set of variables with corresponding finite domains $D(x_1), ..., D(x_n)$. We assume that each pair $(x_i, j)$ with $j \in D(x_i)$ induces a cost $c_{ij}$. We now extend any global constraint $C$ on $X$ to an optimization constraint `opt_C` by introducing a cost variable $z$ and defining

$$\texttt{opt\_C}(x_1, ..., x_n, z, c) = \{(d_1, ..., d_n, d) | (d_1, ..., d_n) \in C(x_1, ..., x_n),$$

$$\forall i. d_i \in D(x_i), d \in D(z), \sum_{i=1,...,n} x_{id_i} \leq d\}.$$

# Linear Relaxation



We introduce binary variables $y_{ij}$ for all $i \in \{1, ..., n\}$ and $j \in D(x_i)$, such that

$$
\begin{aligned}
&x_i = j \Leftrightarrow y_{ij} = 1, && \forall i = 1, \ldots, n, \forall j \in D(x_i), \\
&x_i \neq j \Leftrightarrow y_{ij} = 0, && \forall i = 1, \ldots, n, \forall j \in D(x_i), \\
&\sum_{j \in D(x_i)} y_{ij} = 1, && \forall i = 1, \ldots, n.
\end{aligned}
$$

+ constraint dependent linear equation

The reduced-cost are given w.r.t. the objective:

$$\sum_{i=1,\ldots,n} \sum_{j \in D(x_i)} c_{ij} y_{ij}$$

# Filtering by Reduced-Cost (aka “variable fixing”)

Recall that reduced-costs estimate the increase of the objective function when we force a variable into the solution.

Let $\bar{c}_{ij}$ be the reduced cost for the variable-value pair $x_i = j$, and let $z^*$ be the optimal value of the current linear relaxation.

We apply the following filtering rule:

$$\textbf{if } z^* + \bar{c}_{ij} > \max D(z) \textbf{ then } D(x_i) \leftarrow D(x_i) \setminus \{j\}.$$

# References



Régin J.C. (2011). **Global constraints: A survey**. In *Hybrid Optimization*, edited by P.M. Pardalos, P. van Hentenryck, and M. Milano, vol. 45 of **Optimization and Its Applications**, pp. 63–134. Springer New York.

van Hoeve W. and Katriel I. (2006). **Global constraints**. In *Handbook of Constraint Programming*, chap. 6. Elsevier.