Centre for
Data Analytics

Insight

Insight Centre for Data Analytics

Explanation-Based Weighted Degree

Emmanuel Hebrard[1], Mohamed Siala[2]

1: LAAS-CNRS, Toulouse France
2:Insight Centre for Data Analytics, UCC

November 10, 2017

DCU NUI Galway OÉ Gaillimh UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh UCD DUBLIN SFI Science Foundation Ireland

CPAIOR 2017

- International Conference on Integration of **Artificial Intelligence** and **Operations Research** Techniques in **Constraint Programming**

CPAIOR 2017

- International Conference on Integration of **Artificial Intelligence** and **Operations Research** Techniques in **Constraint Programming**
- Diverse applications: Network Problems (interesting for the UTRC field service project?), Scheduling, Data-mining, Stochastic Optimization

MiniCP

- Pierre Schaus, Pascal Van Hentenryck, Laurent Michel
- 1500 lines of Java code
- Code: `https://bitbucket.org/pschaus/minicp`
- Slides: `http://tinyurl.com/y8n4knhx`
- Purposes: teaching CP, baseline for development projects, etc

Relaxation Methods for Constrained Matrix Factorization Problems: Solving the Phase Mapping Problem in Materials Discovery

- Phase mapping: Central problem in materials discovery
- How to infer the materials' crystal structure based on X-ray sample data.
- The problem: factorize A into the product of W and H such that $A = W \times H$.

Constraint Programming, how it works?

Constraint Programming, how it works?

- A problem is defined with a set of constraints operating over variables associated to domains

Constraint Programming, how it works?

- A problem is defined with a set of constraints operating over variables associated to domains
- Is there a solution that satisfies all constraints (that eventually optimises an objective function)?

Constraint Programming, how it works?

- A problem is defined with a set of constraints operating over variables associated to domains
- Is there a solution that satisfies all constraints (that eventually optimises an objective function)?
- CP solvers: backtracking systems: **search** + propagation

Constraint Programming, how it works?

- A problem is defined with a set of constraints operating over variables associated to domains
- Is there a solution that satisfies all constraints (that eventually optimises an objective function)?
- CP solvers: backtracking systems: **search** + propagation
- Search: decisions to explore the search tree

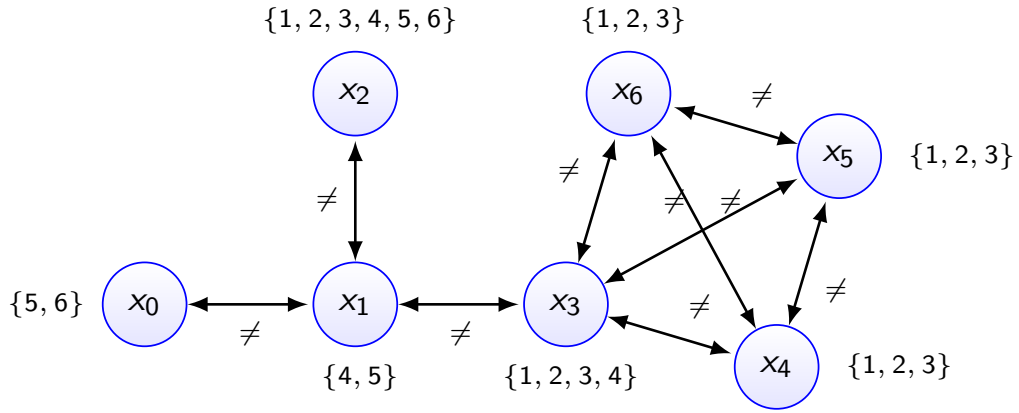
Constraint Programming, how it works?

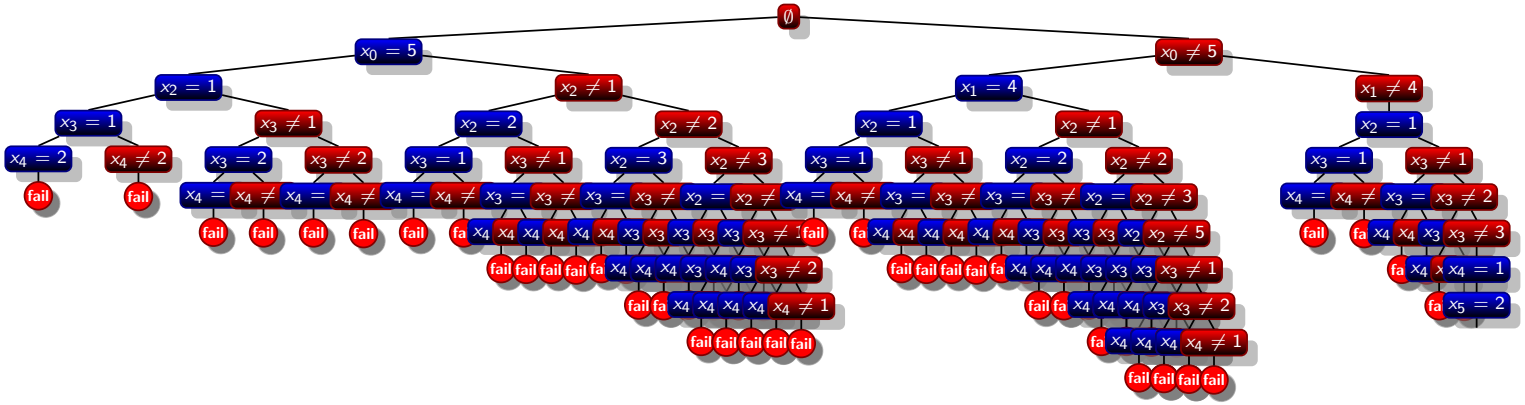
- A problem is defined with a set of constraints operating over variables associated to domains
- Is there a solution that satisfies all constraints (that eventually optimises an objective function)?
- CP solvers: backtracking systems: **search** + propagation
- Search: decisions to explore the search tree
- Search in CP = **variable ordering** + value ordering

Constraint Programming, how it works?

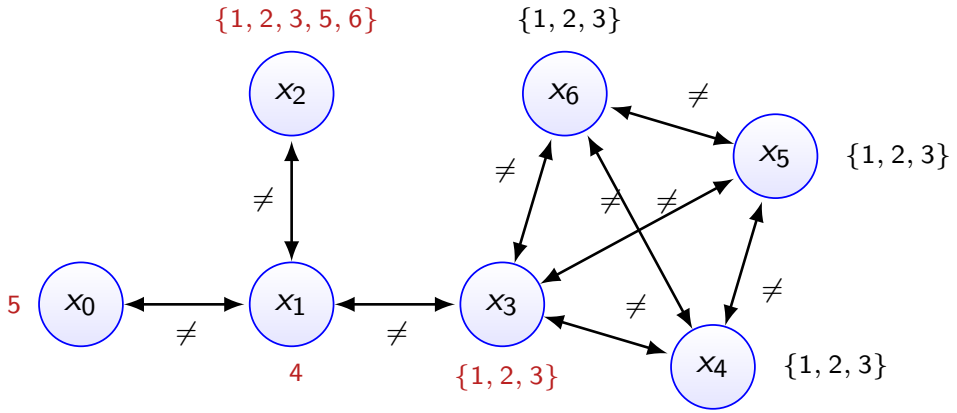
- A problem is defined with a set of constraints operating over variables associated to domains
- Is there a solution that satisfies all constraints (that eventually optimises an objective function)?
- CP solvers: backtracking systems: **search** + propagation
- Search: decisions to explore the search tree
- Search in CP = **variable ordering** + value ordering
- Standard or customized

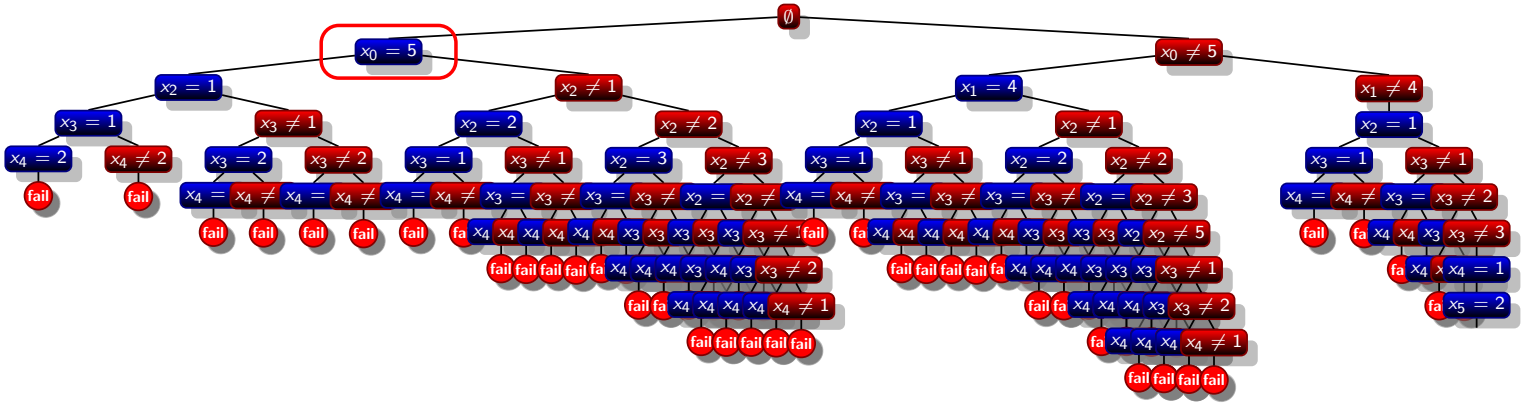
Lexicographic





Lexicographic





Context

Context

Weighted Degree Heuristic (*wdeg*)

Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"

Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"
- *wdeg*: give priority to variables involved in failures

Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"
- *wdeg*: give priority to variables involved in failures
  - $w(C)$: the number of times C caused a failure

Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"
- *wdeg*: give priority to variables involved in failures
  - $w(C)$: the number of times C caused a failure
  - $wdeg(x_i) = \sum_{C \in \mathcal{C}_i} w(C)$ where $\mathcal{C}_i$ are the active constraints involving x_i

Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"
- *wdeg*: give priority to variables involved in failures
  - $w(C)$: the number of times C caused a failure
  - $wdeg(x_i) = \sum_{C \in \mathcal{C}_i} w(C)$ where $\mathcal{C}_i$ are the active constraints involving x_i
  - Select the variable x with lowest ratio $|\mathcal{D}(x)|/wdeg(x)$

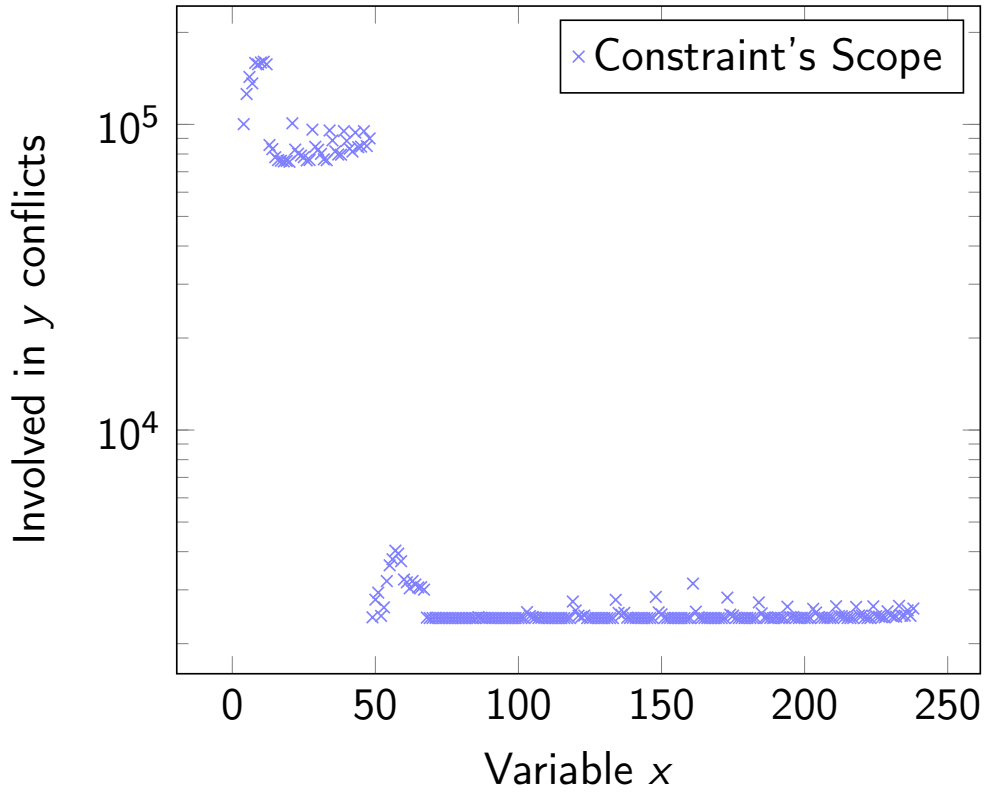
Context

Weighted Degree Heuristic (*wdeg*)

- **Fail-first principle** "To succeed, try first where you are most likely to fail"
- *wdeg*: give priority to variables involved in failures
  - $w(C)$: the number of times C caused a failure
  - $wdeg(x_i) = \sum_{C \in \mathcal{C}_i} w(C)$ where $\mathcal{C}_i$ are the active constraints involving x_i
  - Select the variable x with lowest ratio $|\mathcal{D}(x)|/wdeg(x)$
- **A big issue with global constraints as it does not discriminate between variables**

The Discrimination Issue

- An instance of ghoulomb.mzn (contains a lot of AllDifferent Constraints)



The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

- $d/wdeg(x_1), d/wdeg(x_2) = n - 1/2, d/wdeg(x_i) = n \; \forall i > 2$

The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

- $d/wdeg(x_1), d/wdeg(x_2) = n - 1/2, d/wdeg(x_i) = n \; \forall i > 2$
  - wdeg chooses x_1 and fails on AllDifferent (lower priority)

The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

- $d/wdeg(x_1), d/wdeg(x_2) = n - 1/2, d/wdeg(x_i) = n \; \forall i > 2$
  - wdeg chooses x_1 and fails on AllDifferent (lower priority)
- $d/wdeg(x_1) = (2n-2)/3, d/wdeg(x_i) = n/2 \; \forall i > 2$

The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

- $d/wdeg(x_1), d/wdeg(x_2) = n - 1/2, d/wdeg(x_i) = n \; \forall i > 2$
  - wdeg chooses x_1 and fails on AllDifferent (lower priority)
- $d/wdeg(x_1) = (2n-2)/3, d/wdeg(x_i) = n/2 \; \forall i > 2$
  - **wdeg now branches first on $x_3, \ldots, x_n$!**

The "Flooding" Issue

$$
\begin{aligned}
x_1, x_2 &\in [1, 2n-1] \\
x_3, \ldots, x_n &\in [1, n] \\
& \mathit{AllDifferent}(x_1, \ldots, x_n) \\
& x_1 = x_2
\end{aligned}
$$

- $d/wdeg(x_1), d/wdeg(x_2) = n - 1/2, d/wdeg(x_i) = n \; \forall i > 2$
  - wdeg chooses x_1 and fails on AllDifferent (lower priority)
- $d/wdeg(x_1) = (2n-2)/3, d/wdeg(x_i) = n/2 \; \forall i > 2$
  - **wdeg now branches first on $x_3, \ldots, x_n$!**
- Weight distributed indiscriminately "masks" the degree

Explanation-based Weighted Degree

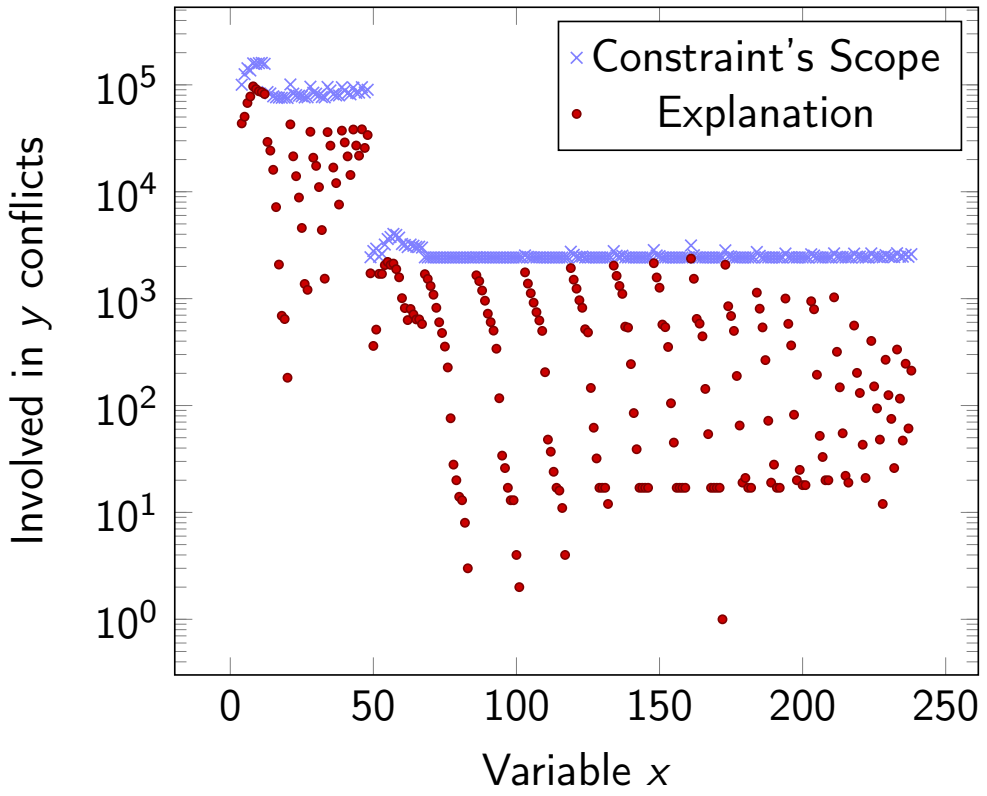
- **A simple solution:** Weight a subset of variables (**explanation**) responsible for failure instead of the whole constraint scope.

Explanation-based Weighted Degree

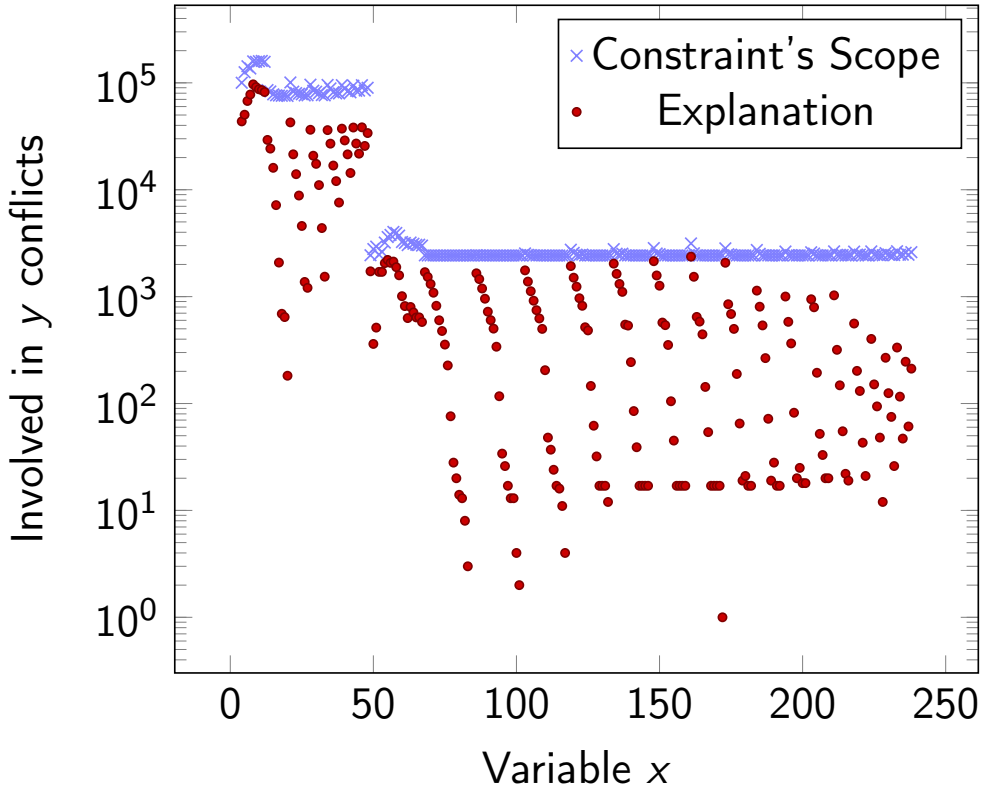
- **A simple solution:** Weight a subset of variables (**explanation**) responsible for failure instead of the whole constraint scope.
- Example: $\sum_{i=1}^{n} x_i \leq k$: Variables whose domain is equal to $\{1\}$ are sole responsible for failure.

Explanation-based Weighted Degree (e-wdeg)

Explanation-based Weighted Degree (e-wdeg)



Explanation-based Weighted Degree (e-wdeg)



- *e-wdeg*: Optimal solution was proven 15s
- *wdeg*: 83% optimality gap after 20 minutes

Explanations for three Global Constraints

Explanations for three Global Constraints

AllDifferent

- Bound Consistency Propagator: Failure if there exists a Hall interval, i.e., a set of at least $b - a + 2$ variables whose domains are included in $\{a, \ldots, b\}$
- Explanation: the set of variables from the Hall interval.

Explanations for three Global Constraints

AllDifferent

- Bound Consistency Propagator: Failure if there exists a Hall interval, i.e., a set of at least $b - a + 2$ variables whose domains are included in $\{a, \ldots, b\}$
- Explanation: the set of variables from the Hall interval.

Element$(\langle x_1, \ldots, x_k \rangle, n, v) \Leftrightarrow x_n = v$,

- We use the conflict set $\{n, v\} \cup \{x_i \mid i \in \mathcal{D}(n)\}$

Explanations for three Global Constraints

$\sum_{i=1}^{k} a_i x_i \leq b$

- Failure if and only if the lower bound of the sum is strictly larger than b.
- Explanation: The set containing every variable x_i such that either a_i is positive and $\min(\mathcal{D}(x_i))$ is strictly larger than its initial value, or a_i is negative and $\max(\mathcal{D}(x_i))$ is strictly lower than its initial value.

Experimental Setup: Branching Heuristics

Experimental Setup: Branching Heuristics

- Activity-based Search (*ABS*) [?]: Priority is given to the variable whose domain was most often reduced during propagation.

Experimental Setup: Branching Heuristics

- Activity-based Search (*ABS*) [?]: Priority is given to the variable whose domain was most often reduced during propagation.
- Impact-Based Search (*IBS*) [?]: Priority is given to the variable with highest expected impact.

Experimental Setup: Branching Heuristics

- Activity-based Search (*ABS*) [?]: Priority is given to the variable whose domain was most often reduced during propagation.
- Impact-Based Search (*IBS*) [?]: Priority is given to the variable with highest expected impact.
- Last Conflict (*LC*) [?] : Once both branches ($x = a$ and $x \neq a$) have failed, the variable x is always preferred (it needs a default heuristic).

Experimental Setup: Branching Heuristics

- Activity-based Search (*ABS*) [?]: Priority is given to the variable whose domain was most often reduced during propagation.
- Impact-Based Search (*IBS*) [?]: Priority is given to the variable with highest expected impact.
- Last Conflict (*LC*) [?] : Once both branches ($x = a$ and $x \neq a$) have failed, the variable x is always preferred (it needs a default heuristic).
- Conflict Ordering Search (*COS*) [?]: Every variable is stamped by the total number of failures it caused. The variable with the highest stamp is selected first (it needs a default heuristic)

Experimental Setup

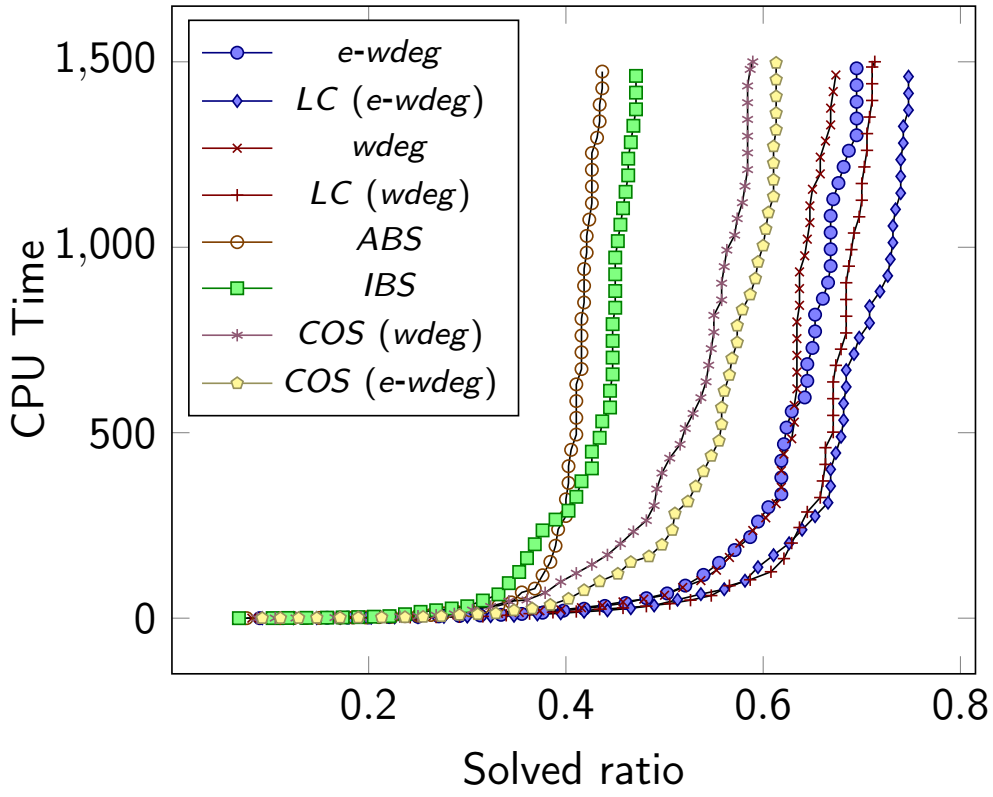
- All Minizinc Challenge instances (2012 to 2015): 323 optimization problems; 76 decision problems
- Comparison with *wdeg*, *ABS*, *IBS*, *COS* and *LC* with *wdeg* and *e-wdeg* as default heuristic for the two latter.
- Lexicographic value ordering for every heuristic, except *IBS* and *ABS*.
- Randomization by choosing uniformly between the two best choices.
- Each configuration was given 5 randomized runs.
- 25 minutes as time cutoff

Experimental Setup

- All Minizinc Challenge instances (2012 to 2015): 323 optimization problems; 76 decision problems
- Comparison with *wdeg*, *ABS*, *IBS*, *COS* and *LC* with *wdeg* and *e-wdeg* as default heuristic for the two latter.
- Lexicographic value ordering for every heuristic, except *IBS* and *ABS*.
- Randomization by choosing uniformly between the two best choices.
- Each configuration was given 5 randomized runs.
- 25 minutes as time cutoff
- **In case you wonder why the cluster is busy: with some other parameters, the total CPU time to complete experiments is about 3 years and a half!**

Decision Problems

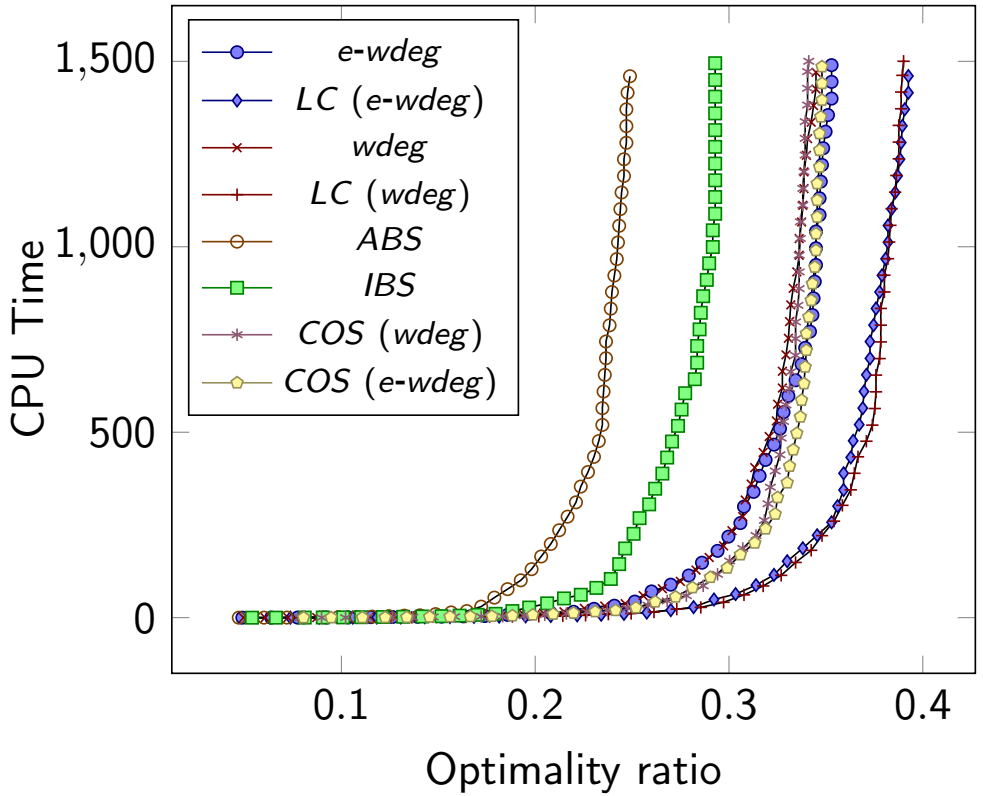
Decision Problems



- *e-wdeg* better than *wdeg*
- Weighted degree heuristics are among the state of the art in CP

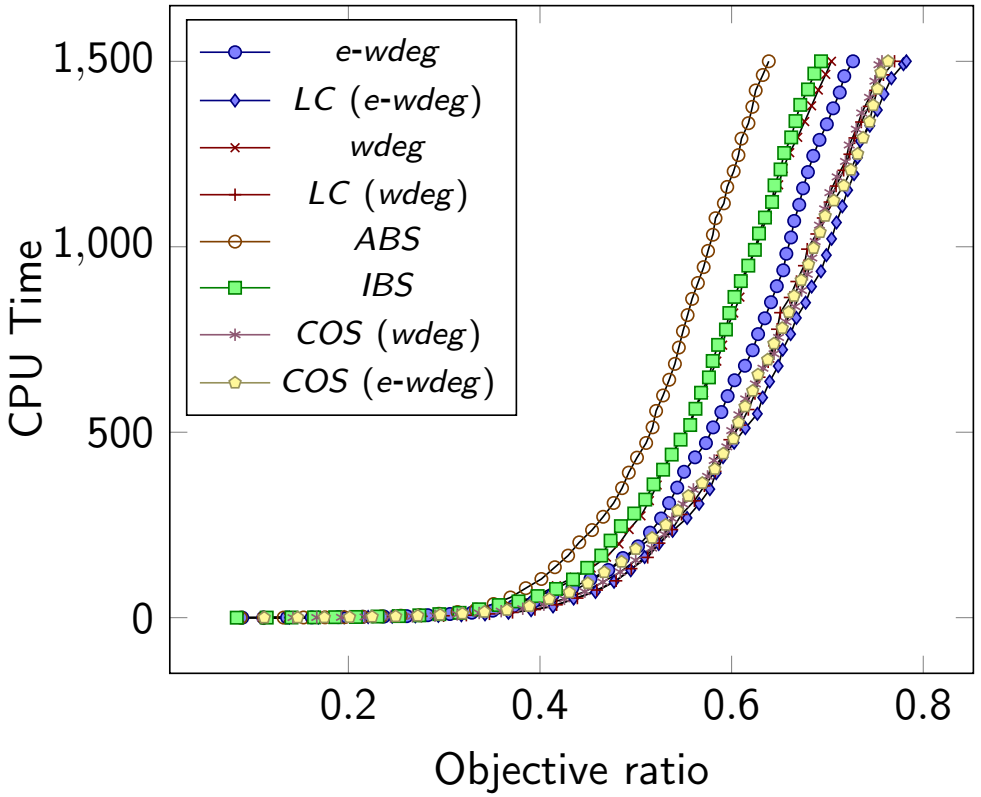
Optimisation Problems (Number of proofs)

Optimisation Problems (Number of proofs)



Optimisation Problems (Objective value)

Optimisation Problems (Objective value)



Conclusions & Future Research

Conclusions & Future Research

Take-away messages

- Explanation-based weighted degree helps!
- Easy to implement
- Last Conflict $\oplus$ *e-wdeg* outperforms the others (different configurations, XCSP3 benchmarks)

Conclusions & Future Research

Take-away messages

- Explanation-based weighted degree helps!
- Easy to implement
- Last Conflict $\oplus$ *e-wdeg* outperforms the others (different configurations, XCSP3 benchmarks)

Future research

- Design better explanations
- Explain more constraints
- How to choose between different explanations?