# ClpZinc

## Search as Constraint Satisfaction

**published in PPDP'15 (Principles and Practice of Declarative Programming)**


Thierry Martinez, François Fages, Sylvain Soliman

EPI Lifeware, Inria Paris–Rocquencourt, France



ANR project Net-WMS-2

Networked Warehouse Management Systems 2: Packing with Complex Shapes.


## Search Strategy for Constraint Programming

Constraint programming

=

Constraint model + Search procedure

high level, MiniZinc, relational — hardly declarative, very dependent to the solver

Search procedure is crucial to solve hard combinatorial (NP-complete) problems.

## The Clp2Zinc Theorem

reified constraints

Constraint model $\mathcal{M}$ — Constraint model $\mathcal{M} \uplus \mathcal{M}_t$

Tree search procedure $t$ $\underset{\text{computationally similar}}{\sim}$ Basic labeling $t'$

Reified constraints: $X = 1 \Leftrightarrow c$ is true.

## ClpZinc

**A Modeling Language for Constraints and Search.**

- **Modeling** search independently from the underlying constraint solver through tree search procedures with state variables.
- Extending MiniZinc with **Horn clauses with constraints** (Prolog-like search description language).

Available compiler targeting most common solvers:

`http://lifeware.inria.fr/~tmartine/clp2zinc`

## Dichotomic Search: The Code

```
dichotomy(X, Min, Max) :-
   dichotomy(X, ceil(log(2, Max - Min + 1))).
dichotomy(X, Depth) :-
   Depth > 0,
   Middle = (min(X) + max(X)) div 2,
   (X <= Middle ; X > Middle),
   dichotomy(X, Depth - 1).
dichotomy(X, 0).
var 0..5: x;
:- dichotomy(x, 0, 5).
```

## Interval Splitting: The Code

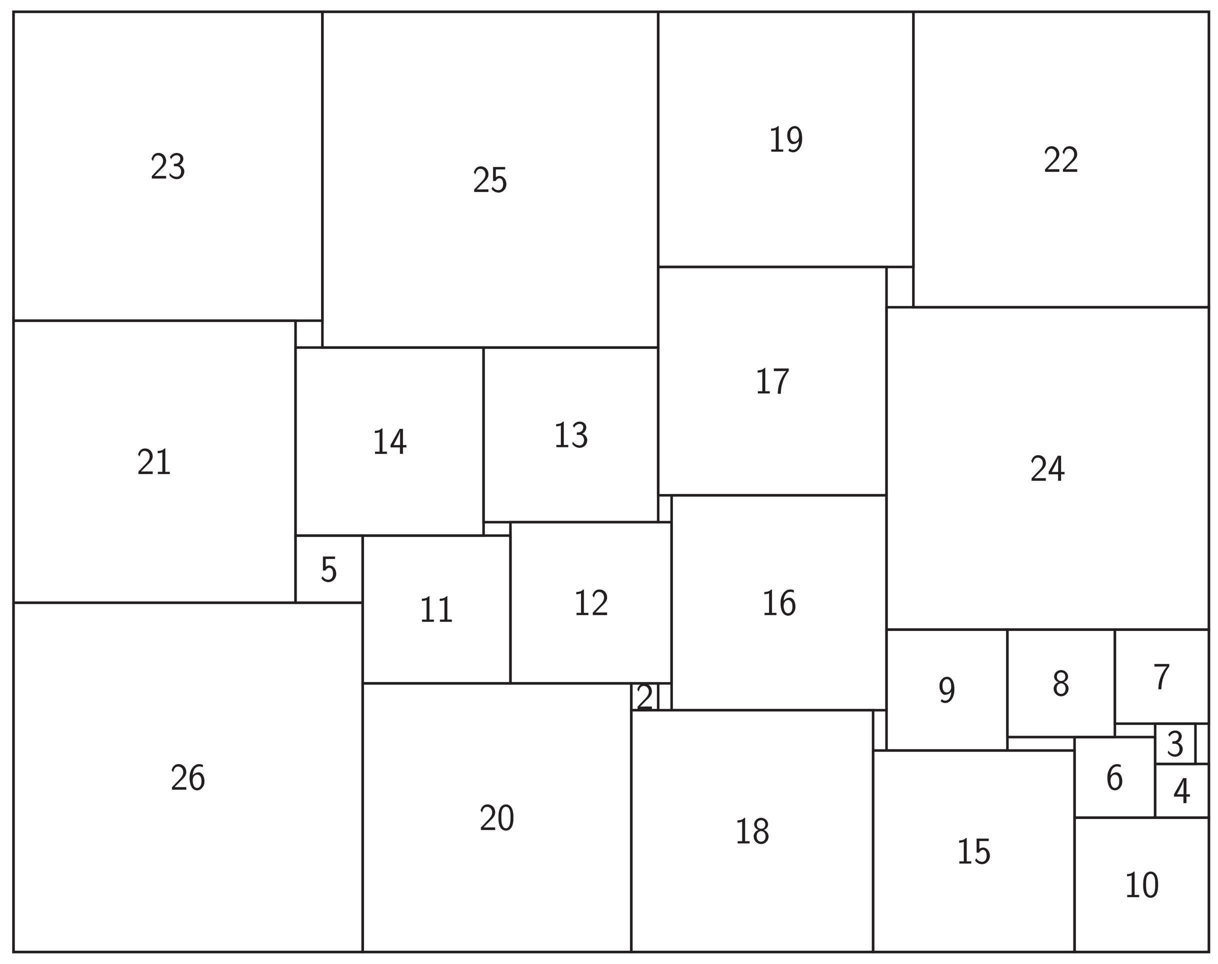
```
interval_splitting(X, Step, Min, Max) :-
   Min + Step <= Max, NextX = min(X) + Step,
   (
      X < NextX
   ;
      X >= NextX,
       interval_splitting(X, Step, Min + Step, Max)
   ).
interval_splitting(X, Step, Min, Max) :-
   Min + Step > Max.
var 0..5: x;
:- interval_splitting(x, 2, 0, 5).
```

## Korf's packing problem

Given an integer $n \geq 1$, find an **enclosing rectangle of smallest area containing $n$ squares** from sizes $1 \times 1$, $2 \times 2$, up to $n \times n$, **without overlap**.

2008. *Search strategies for rectangle packing.* H. Simonis and B. O'Sullivan. Proceedings of CP'08.

One **provably optimal placement** for $n = 26$:




In practice, ClpZinc programs is 2–3× slower than native programs: the constant is small w.r.t. to the combinatorial complexity.

## Meta-interpretation, beyond tree search

**Meta-interpretation:**

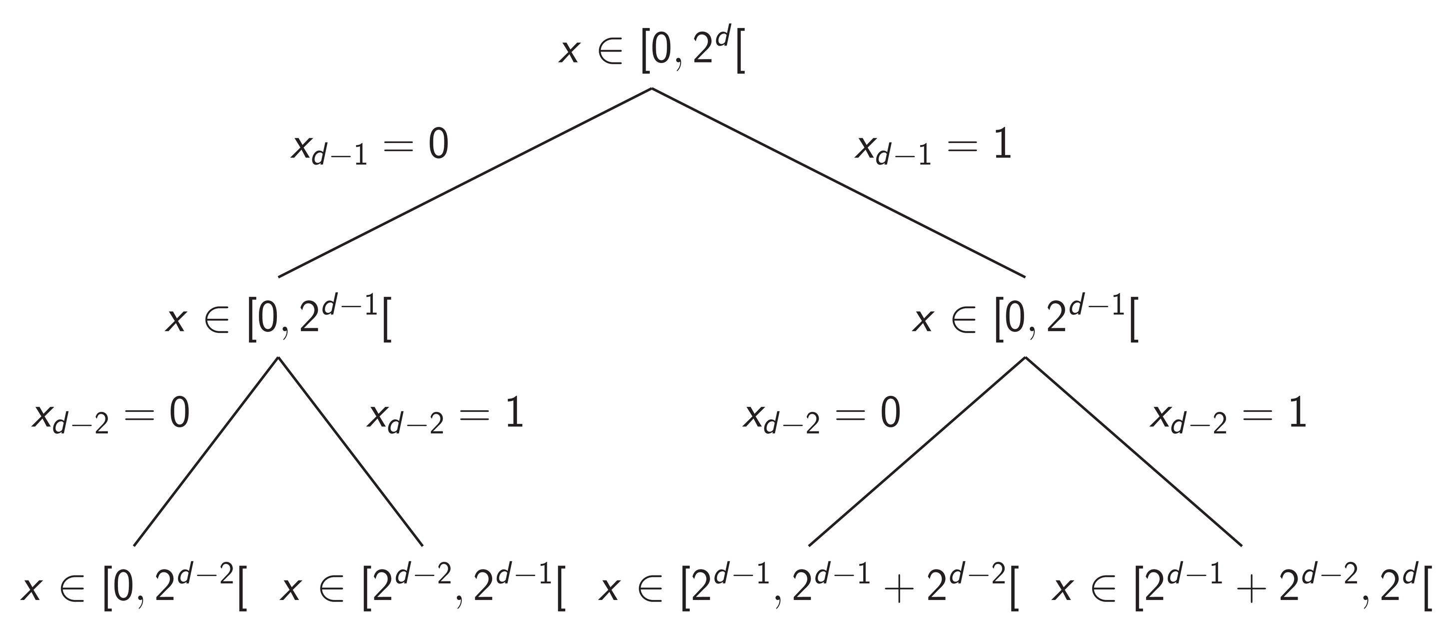
- Limited discrepancy search (LDS)
- Symmetry breaking during search (SBDS)

**State variables, persistent through backtracking:**

Optimization procedure, branch-and-bound.

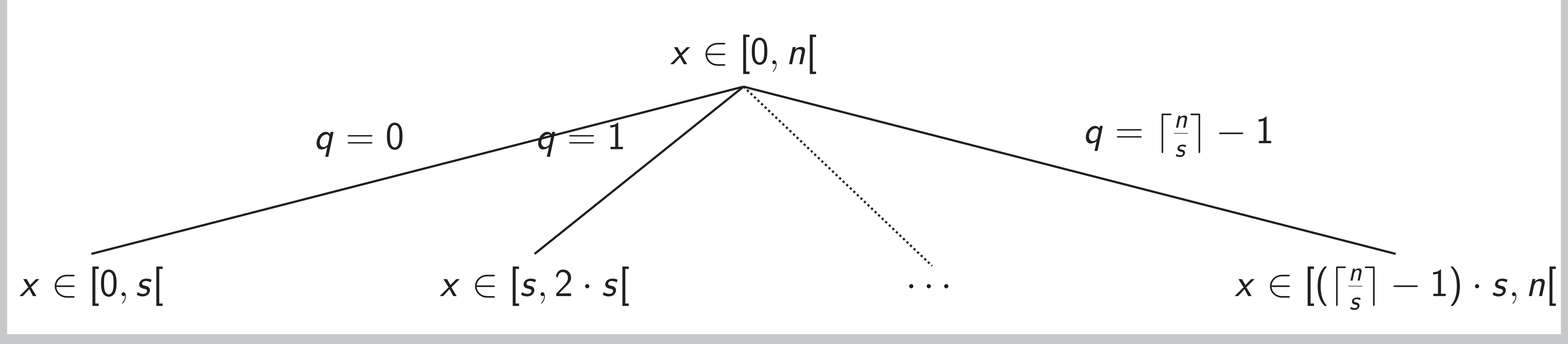
## Dichotomic Search: The Search Tree

For $x \in [0, 2^d[$. Obtained by **domain filtering** and **constraint propagation** of the equality $x = \Sigma_{0 \leq k < d} x_k 2^k$ with $x_k \in \{0, 1\}$.




## Interval Splitting: The Search Tree

For a fixed step $s \geq 1$ and for $x \in [0, n[$. Obtained by **domain filtering** and **constraint propagation** of the equality $x = s \times q + r$ where $r \in [0, s[$.







{Thierry.Martinez, Francois.Fages, Sylvain.Soliman}@inria.fr