# Search strategies for floating point constraint systems

CP 2017

**Heytem Zitoun**[1] Claude Michel[1] Michel Rueher[1] Laurent Michel[2]

[1] Université Côte d'Azur

[2] University of Connecticut

# OUTLINE

1. Basics of Floating point numbers
2. Context and motivations
3. Dedicated floating point search strategies
4. Search strategies implementation
5. Experiments and futherwork

# Basics of Floating point numbers

# Definition of floats

Consider the number **−11.5**

In **base 2** we can represent this number in the following way :


**IEEE754 norm** :

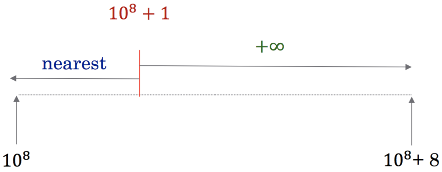
- Simple precision : 32 bits (1 bits for the sign + 23 for mantissa + 8 for exponent)
- Double precision : 64 bits (1 + 52 + 11)

## Absorption

*Absorption occurs when adding two floating point numbers with different orders of magnitude. The result is the biggest number for positive numbers (resp. smallest for negative numbers)*

Example :

$$10^8 + 1 = 10^8$$



# Issues with FP computations (cont.)

## Cancellation : loss of the most significant bits

Occur when subtracting two close numbers with FP error :

$$\left(\overbrace{(1.0 - 10^{-7})}^{0.99999988079071044922 \neq 0.9999999} - 1.0\right) = -0.000001013278 \neq -10^{-7}$$

Using this result to compute other value lead to bigger errors :

$$\left((1.0 - 10^{-7}) - 1.0\right) * 10^{7} = -1.1920928955078125 \neq -1$$

Rump polynomial

$$R(x,y) = \frac{1335}{4} y^6 + (11x^2y^2 - y^6 - 121\,y^4 - 2)x^2 + \frac{11}{2} y^8 + \frac{x}{2y}$$

Over $\mathbb{F}$ with $x = 77617$ and $y = 33096$

# Context and motivation

# Overview

New Approach :

→ **Dedicated search strategies for floating point numbers**

Why :

- Verification of FP program
- Existing search are not well adapted

## Why do we need floating point constraints ?

$\rightarrow$ Verification programs with FP computations (Bounded Model Checking, SMT, ...)

- **Programs** are run over the floats, but are written with reals in mind

Constraints over the reals $\neq$ Constraints over the floats

- $16.0 + x = 16.0$ with $x > 0$
  solutions exist over the floats but no solution exists over reals
- $x^2 = 2$
  no solution exists over the floats ($\sqrt{2}$ is a solution over reals)

# A small program with floating point numbers

```
void foo(a, b, c){
  float r = a + b + c;
  if(r >= 1.0f){
          GoHere
  }else{
          GoHere
  }
}
```

Evaluation over $\mathbb{R}$*eals* and $\mathbb{F}$*loats* can be different

With a $= 10^8$, b $= 1.0$, c $= -10^8$

$a, b, c, r \in \mathbb{R} \rightarrow$ GoHere

$a, b, c, r \in \mathbb{F} \rightarrow$ GoHere

# Searching is a critical issue

Goal : **Searching** a path where the **execution over FP** differ from the **expected behavior** over the reals

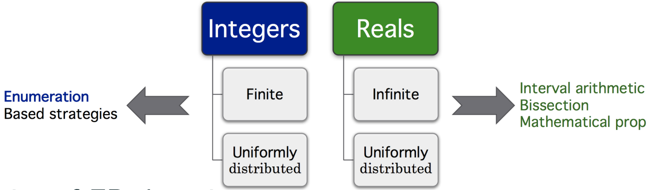
$\rightarrow$ efficient search strategies over floats

**Problem** :
Searching strategies in existing FP solvers are **derived** from the searching strategies over **the reals** and those strategies don't scale !

# Dedicated floating point search strategies

# Why classical strategies are not well adapted ?

Classi




Specificity of FP domain :

- Finite but we work with very large domains
  $[-1, 1]$ more than $10^{18}$ FP numbers
- Non-uniformly distributed
  more than half FP numbers between $[-1, 1]$

$\rightarrow$ Classical strategies don't work for floats

# FP search strategies : intuitions

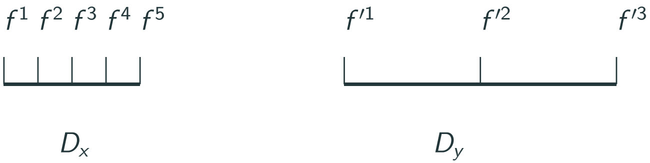
**Take advantage of :**

- the structure of variable domain
- floating point arithmetic issues (absorptions, cancellation)
- structure of the problem (constraints)

$\rightarrow$ Definition/Computation **properties** for domain of variables (width, cardinality, ...) and constraints (degree, occurences, ...)

# Properties based on the domain of variable

let $x$ and $y$ be two FP variables with the following domains

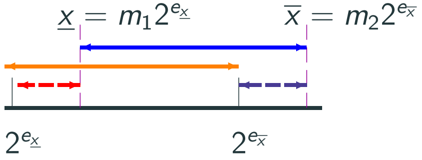



Which criteria should we use to select the variable to split ?

- **Width** : $D_y$ is **bigger** than $D_x$
- **Cardinality** : $|D_x| > |D_y|$
- **Density** : $D_x$ is **more dense** than $D_y$
- **Magnitude** : $magn(y) > magn(x)$

# Computing cardinality

Goal : compute the number of floats between $[\underline{x}, \overline{x}]$




$$|D_x| = \longleftrightarrow \ominus \longleftrightarrow \oplus \longleftrightarrow$$

$$|D_x| = 2^p * (e_{\overline{x}} - e_{\underline{x}}) - m_{\underline{x}} + m_{\overline{x}}$$

# Properties based on constraints

Consider the following system :

$$(x - y) * y = z$$

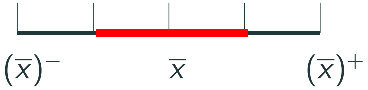
$$y * y = w/x$$

Which variable should we select to split?

- The variable with the highest **degree** ?
- The variable with the largest number of **occurences** ?
- A variable that can lead to an **absorption** ?
- A variable that can lead to a **cancellation** ?

# Computing absorption

let $\mathbf{z} = \mathbf{x} + \mathbf{y}$, $\mathbf{x} \geq \mathbf{0}$ and $x$ absorbs $y$




Red part corresponds to the distance where y is absorbed by x.
Distance $D_{abs}$ : $[\frac{\overline{x}-\overline{x}^+}{2}, \frac{\overline{x}+\overline{x}^+}{2}]$

$y \in D_{abs} \rightarrow$ x absorbs y

# Search strategies

# Strategies based on a single property

**Two strategies :**

- maximizing the property
- minimizing the property

**Example cardinality :**

select the variable with the largest (resp. smallest) the number of floating point numbers.

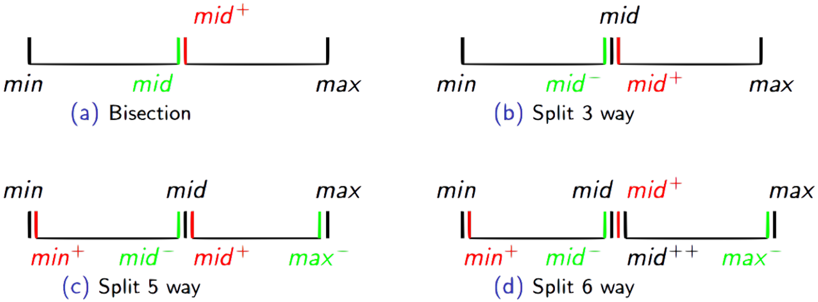
# Strategies based on a combination of properties

## Example : combination of density and absorption

V : set of variables from the system

- AbsWDens :
  - Step 1 $\rightarrow$ $V_2$ set of variables from V with $abs(x \in V) > 0$
  - Step 2 $\rightarrow$ $max_{dens}(x' \in V_2)$
- DensWAbs :
  - Step 1 $\rightarrow$ $V_2$ set of variables from V with $dens(x \in V) > \frac{min_{dens}+max_{dens}}{2}$
  - Step 2 $\rightarrow$ $max_{abs}(x' \in V_2)$

# Splitting strategies




mid : middle of the interval

$f^+$ (resp. $f^-$): the successor (resp. predecessor) of f

# Splitting strategies schema

**Full**

**foreach** *selected variable* $x_i$ **do**
  **Split** the domain of $x_i$ once
**end**

**Semi**

**Select** variable $x_j$
**while** *variable* $x_j$ *is not bound* **do**
  **Split** the domain of $x_j$
**end**

# Implementation & Experiments

# Implementation

We use **Objective-CP** optimization system

- developed by **L. Michel** and **P. Van Hentenryck**
- various different solvers (LP, MIP, CP, ...)
- very flexible search system

We incorporate **FPCS** solver

- developed by **Claude Michel**
- filtering technics implemented (2B and 3B consistency over the floats)
- handling of rounding modes, nonlinear expressions and usual mathematical functions (trigonometric, ...)

# Experiments

- **Combinaisons** : different variable selection strategies + different splitting strategies
- **Reference strategy** : lexicographic + bisection
- Benchmarks from program verification problems
- Time in seconds (timeout 180 seconds)

# Results

| variable choice | | split. | $\sum$ t |
|---|---|---|---|
| strat. | dyn. | | (ms) |
| maxAbs | semi | 6 | 4883 |
| maxAbs | full | 6 | 4930 |
| maxDens | semi | 6 | 5059 |
| densWAbs | full | 6 | 7517 |
| maxCard | semi | 6 | 180191 |
| densWAbs | semi | 6 | 180194 |
| maxDegree | full | 6 | 180307 |
| maxDegree | semi | 6 | 180310 |
| maxAbs | full | 5 | 184613 |
| maxDens | semi | 5 | 184796 |
| ... | | | |
| ref | | | 550988 |
| ... | | | |
| minDegree | semi | 3 | 906285 |
| minOcc | semi | 3 | 906285 |
| maxWidth | semi | 3 | 906607 |
| minCard | semi | 3 | 911526 |
| maxMagn | semi | 3 | 1077852 |
| absWDens | full | 3 | 1080002 |
| maxWidth | semi | 2 | 1080004 |
| minDens | semi | 3 | 1080005 |
| absWDens | full | 5 | 1080147 |
| absWDens | full | 6 | 1440000 |

(a) all

| variable choice | | split. | $\sum$ t |
|---|---|---|---|
| strat. | dyn. | | (ms) |
| maxAbs | semi | 6 | 187 |
| maxCard | semi | 6 | 189 |
| densWAbs | semi | 6 | 191 |
| densWAbs | full | 6 | 196 |
| maxAbs | full | 6 | 202 |
| maxDens | semi | 6 | 217 |
| maxDegree | full | 6 | 305 |
| maxDegree | semi | 6 | 307 |
| maxWidth | full | 6 | 31244 |
| minDens | full | 6 | 38332 |
| ... | | | |
| ref | | | 540011 |
| ... | | | |
| minDens | semi | 3 | 720005 |
| minDegree | semi | 2 | 720005 |
| minDegree | full | 2 | 720005 |
| minAbs | full | 3 | 720005 |
| maxDens | full | 3 | 720006 |
| minOcc | semi | 2 | 720006 |
| minAbs | full | 2 | 720006 |
| absWDens | full | 5 | 720147 |
| maxWidth | semi | 2 | 900002 |
| absWDens | full | 5 | 1080000 |

(b) with solutions

| variable choice | | split. | $\sum$ t |
|---|---|---|---|
| strat. | dyn. | | (ms) |
| maxAbs | semi | 2 | 2376 |
| maxAbs | full | 2 | 2379 |
| maxAbs | full | 3 | 2410 |
| maxDens | semi | 2 | 2439 |
| maxCard | full | 3 | 4405 |
| maxAbs | semi | 5 | 4451 |
| maxAbs | full | 5 | 4467 |
| maxCard | full | 2 | 4594 |
| maxDens | semi | 5 | 4626 |
| maxAbs | semi | 6 | 4696 |
| ... | | | |
| ref | | | 10977 |
| ... | | | |
| maxMagn | semi | 3 | 360000 |
| minMagn | semi | 3 | 360000 |
| maxDegree | semi | 3 | 360000 |
| minDegree | semi | 3 | 360000 |
| minOcc | semi | 3 | 360000 |
| absWDens | semi | 2 | 360000 |
| absWDens | semi | 3 | 360000 |
| absWDens | semi | 5 | 360000 |
| absWDens | semi | 6 | 360000 |
| densWAbs | semi | 3 | 360000 |

(c) without solution

# Experiments

## Analysis

- **Single property** : absorption and density outperform other strategies
- **Combinaisons** : densWAbs improve maxDens results
- **Splitting** : when a solution exists our splitting strategies are better than bisection

Details :

http://www.i3s.unice.fr/~hzitoun/cp2017/benchmark.html

**! Preliminary experiments.**

# Conclusion

# Conclusion

## Contribution

- Introduction of set of **properties** (measure)
- **First dedicated approach** to floating point search strategies based
- Preliminaries experiments are **encouraging**

## Futhercomming work

- more experiments
- development of new searching and splitting strategies