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

\[ 10^8 + 1 = 10^8 \]
Cancellation: loss of the most significant bits

Occur when subtracting two close numbers with FP error:

\[
\frac{0.99999988079071044922}{0.9999999} \neq \frac{0.99999988079071044922}{0.9999999} \\
(1.0 - 10^{-7}) - 1.0 = -0.000001013278 \neq -10^{-7}
\]

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

\[
(1.0 - 10^{-7}) \times 10^7 = -1.1920928955078125 \neq -1
\]

Rump polynomial

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

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?

→ 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 ≠ 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

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

Evaluation over \( \mathbb{Reals} \) and \( \mathbb{Floats} \) can be different

With \( a = 10^8, b = 1.0, c = -10^8 \)

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

\( a, b, c, r \in \mathbb{F} \rightarrow GoHere \)
Goal: Searching a path where the execution over FP differ from the expected behavior over the reals

→ 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?

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 $[-1, 1]$

→ 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)

→ Definition/Computation properties for domain of variables (width, cardinality, ...) and constraints (degree, occurrences, ...)
Properties based on the domain of variable

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

\begin{align*}
D_x &= \{ f^1, f^2, f^3, f^4, f^5 \} \\
D_y &= \{ f'1, f'2, f'3 \}
\end{align*}

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**: $\text{magn}(y) > \text{magn}(x)$
Computing cardinality

Goal: compute the number of floats between \([x, \bar{x}]\)

\[ x = m_1 2^{e_x} \quad \bar{x} = m_2 2^{e_x} \]

\[ |D_x| = 2^p \ast (e_x - e_{\bar{x}}) - m_x + m_{\bar{x}} \]
Properties based on constraints

Consider the following system:

\[(x - y) \times y = z\]
\[y \times y = w/x\]

Which variable should we select to split?

- The variable with the highest degree?
- The variable with the largest number of occurrences?
- A variable that can lead to an absorption?
- A variable that can lead to a cancellation?
let \( z = x + y \), \( x \geq 0 \) and \( x \) absorbs \( y \)

Red part corresponds to the distance where \( y \) is absorbed by \( x \).

Distance \( D_{abs} : \left[ \frac{x-x^+}{2}, \frac{x+x^+}{2} \right] \)

\( y \in D_{abs} \rightarrow x \) 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**

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

**Semi**

```plaintext
Select variable $x_j$

while variable $x_j$ is not bound do
  Split the domain of $x_j$
end
```
Implementation & Experiments
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, ...)

**Implementation & Experiments**

**CP 2017**
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

<table>
<thead>
<tr>
<th>variable choice</th>
<th>split.</th>
<th>$\sum t$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>strat.</td>
<td>dyn.</td>
<td></td>
</tr>
<tr>
<td>maxAbs</td>
<td>semi</td>
<td>180194</td>
</tr>
<tr>
<td>maxAbs</td>
<td>full</td>
<td>180307</td>
</tr>
<tr>
<td>maxDens</td>
<td>semi</td>
<td>180310</td>
</tr>
<tr>
<td>densWAbs</td>
<td>semi</td>
<td>180194</td>
</tr>
<tr>
<td>maxCard</td>
<td>semi</td>
<td>180191</td>
</tr>
<tr>
<td>maxDens</td>
<td>full</td>
<td>180310</td>
</tr>
<tr>
<td>maxDegree</td>
<td>semi</td>
<td>180310</td>
</tr>
<tr>
<td>maxAbs</td>
<td>full</td>
<td>184613</td>
</tr>
<tr>
<td>maxDens</td>
<td>semi</td>
<td>184796</td>
</tr>
</tbody>
</table>

...                 ref     550988

<table>
<thead>
<tr>
<th>variable choice</th>
<th>split.</th>
<th>$\sum t$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>strat.</td>
<td>dyn.</td>
<td></td>
</tr>
<tr>
<td>minDegree</td>
<td>semi</td>
<td>906285</td>
</tr>
<tr>
<td>minOcc</td>
<td>semi</td>
<td>906285</td>
</tr>
<tr>
<td>maxWidth</td>
<td>semi</td>
<td>906607</td>
</tr>
<tr>
<td>minCard</td>
<td>semi</td>
<td>911526</td>
</tr>
<tr>
<td>maxMagn</td>
<td>semi</td>
<td>1077852</td>
</tr>
<tr>
<td>absWDens</td>
<td>full</td>
<td>1080002</td>
</tr>
<tr>
<td>maxWidth</td>
<td>semi</td>
<td>1080004</td>
</tr>
<tr>
<td>minDens</td>
<td>semi</td>
<td>1080005</td>
</tr>
<tr>
<td>absWDens</td>
<td>full</td>
<td>108147</td>
</tr>
<tr>
<td>absWDens</td>
<td>full</td>
<td>1440000</td>
</tr>
</tbody>
</table>

(a) all

(b) with solutions

(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

Further comming work

• more experiments
• development of new searching and splitting strategies