An Incremental Algorithm for Computing Cylindrical Algebraic Decomposition and Its Application to Quantifier Elimination

Changbo Chen
Joint work with Marc Moreno Maza
ORCCA, University of Western Ontario, Canada

Aug. 2, 2013
SIAM AG 2013, Fort Collins, Colorado, USA
Cylindrical Algebraic Decomposition (CAD) of $\mathbb{R}^n$

A CAD of $\mathbb{R}^n$ is a partition of $\mathbb{R}^n$ such that each cell in the partition is a connected semi-algebraic subset of $\mathbb{R}^n$ and all the cells are cylindrically arranged.

Two subsets $A$ and $B$ of $\mathbb{R}^n$ are called cylindrically arranged if for any $1 \leq k < n$, the projections of $A$ and $B$ on $\mathbb{R}^k$ are either equal or disjoint.
Cylindrical algebraic decomposition (CAD)

Invented by G.E. Collins in 1973 for solving Real Quantifier Elimination (QE) problems.

Previous work on CAD

Adjacency and clustering techniques (D. Arnon, G.E. Collins and S. McCallum 84), Improved projection operator (H. Hong 90; S. McCallum 88, 98; C. Brown 01), Partially built CADs (Collins and Hong 91, A. Strzeboński 00), Improved stack construction (G.E. Collins, J.R. Johnson, and W. Krandick), Efficient projection orders (A. Dolzmann, A. Seidl and T. Sturm 04), Making use of equational constraints (G.E. Collins 98; C. Brown and S. McCallum 05, R. Bradford, J. Davenport, M. England, S. McCallum and D. Wilson 12), Set-theoretical operations by CAD (A. Strzeboński 10), Computing CAD via triangular decompositions (C. Chen, M. Moreno Maza, B. Xia and L. Yang 09), ...

Software

Qepcad, Mathematica, Redlog, SyNRAC, RegularChains (TCAD).
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
CAD based on projection-lifting scheme (PCAD)

**Projection**
- Let $Proj$ be a projection operator.
- Repeatedly apply $Proj$:

\[ F_n(x_1, \ldots, x_n) \xrightarrow{Proj} F_{n-1}(x_1, \ldots, x_{n-1}) \xrightarrow{Proj} \cdots \xrightarrow{Proj} F_1(x_1). \]

**Lifting**
- The real roots of the polynomials in $F_1$ plus the open intervals between them form an $F_1$-invariant CAD of $\mathbb{R}^1$.
- For each cell $C$ of the $F_{k-1}$ invariant CAD of $\mathbb{R}^{k-1}$, isolating the real roots of the polynomials of $F_k$ at a sample point of $C$, produces all the cells of the $F_k$-invariant CAD of $\mathbb{R}^k$ above $C$. 
Motivation: potential drawback of Collins’ scheme

- The projection operator is a function defined independently of the input system.
- As a result, a strong projection operator (Collins-Hong operator) usually produces much more polynomials than needed.
- A weak projection operator (McCalumn-Brown operator) may fail for non-generic cases.

Solution: make case discussion during projection

- Case discussion is common for algorithms computing triangular decomposition.
- At ISSAC’09, we (with B. Xia and L. Yang) introduced case discussion (as in triangular decomposition of polynomials systems) into CAD computation. As a result, the projection phase in classical CAD algorithm is replaced by computing a complex cylindrical tree.
Complex cylindrical tree

- Let $\alpha = (\alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{C}^{n-1}$
- Define $\mathbb{C}[x_1, \ldots, x_n] \xrightarrow{\Phi_{\alpha}} \mathbb{C}[x_n]$, where $p(x_1, \ldots, x_n) \mapsto p(\alpha, x_n)$

Separation

Let $S \subset \mathbb{C}^{n-1}$ and $P \subset k[x_1, \ldots, x_{n-1}, x_n]$ be a finite set of level $n$ polynomials. We say that $P$ separates above $S$ if for each $\alpha \in S$:
- For each $p \in P$, $\Phi_{\alpha}(\text{leading coefficient of } p \text{ w.r.t. } x_n) \neq 0$
- The polynomials $\Phi_{\alpha}(p)$ are squarefree and pairwise coprime.

A $\{y^2 + x, y^2 + y\}$-sign invariant complex cylindrical tree
Rethink classical CAD in terms of complex cylindrical tree

The projection factors are \( a, b, c, 4ac - b^2, ax^2 + bx + c \).

```
\text{a = 0}
\quad \begin{aligned}
\quad \text{b = 0} & \quad c = 0 \quad \text{any } x \\
\quad & \quad c \neq 0 \quad \text{any } x \\
\quad \text{b \neq 0} & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
\quad \text{c = 0}
\quad \begin{aligned}
\quad & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
\quad \text{c \neq 0}
\quad \begin{aligned}
\quad & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
```

```
\text{a \neq 0}
\quad \begin{aligned}
\quad \text{b = 0} & \quad 4ac - b^2 = 0 \\
\quad & \quad 4ac - b^2 \neq 0
\end{aligned}
\quad \text{b \neq 0}
\quad \begin{aligned}
\quad & \quad 4ac - b^2 = 0 \\
\quad & \quad 4ac - b^2 \neq 0
\end{aligned}
```

```
\text{c = 0}
\quad \begin{aligned}
\quad & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
\quad \text{c \neq 0}
\quad \begin{aligned}
\quad & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
```

```
\text{c(4ac - b^2) \neq 0}
\quad \begin{aligned}
\quad & \quad ax^2 + bx + c = 0 \\
\quad & \quad ax^2 + bx + c \neq 0
\end{aligned}
```
The complex cylindrical tree constructed by TCAD

- \( a = 0 \)
  - \( b = 0 \)
    - \( c = 0 \) any \( x \)
    - \( c \neq 0 \) any \( x \)
  - \( b \neq 0 \)
    - any \( c \)
      - \( bx + c = 0 \)
      - \( bx + c \neq 0 \)
- \( a \neq 0 \)
  - any \( b \)
    - \( 4ac - b^2 = 0 \) \( 2ax + b = 0 \)
    - \( 2ax + b \neq 0 \)
    - \( 4ac - b^2 \neq 0 \) \( ax^2 + bx + c = 0 \)
    - \( ax^2 + bx + c \neq 0 \)
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Incremental solving

- $p_1 := x^2 + y^2 + z^2 - 4$
- $p_2 := x^2 + y^2 - z^2 - 1$
- $p_3 := z^3 + xy - 1$

\[ W(T) := V(p_1) \cap V(p_2) \quad V(p_3) \cap W(T) \]

Algorithms using incremental strategy: Triangular Decomposition (D. Lazard, 91; $M^3$, 00; C. Chen & $M^3$, 11); Lifting Fibers (G. Lecerf, 2003); Diagonal Homotopy (A.J. Sommese, J. Verschelde, C. W. Wampler, 08).
The refinement operation

Input

- A $y^2 + x$ sign invariant complex cylindrical tree

\[
T := \begin{cases} 
  x = 0 & \begin{cases} 
    y = 0 & : & y^2 + x = 0 \\
    y \neq 0 & : & y^2 + x \neq 0
  \end{cases} \\
  x \neq 0 & \begin{cases} 
    y^2 + x = 0 & : & y^2 + x = 0 \\
    y^2 + x \neq 0 & : & y^2 + x \neq 0
  \end{cases}
\end{cases}
\]

- A polynomial $y^2 + y$.

Output

The tree $T$ is refined into a new tree, above each path of which both $y^2 + x$ and $y^2 + y$ are sign invariant.
Refine the first path of the tree with $y^2 + y$

\[
\begin{align*}
\begin{cases}
    x = 0 \\
    \left\{ \begin{array}{l}
        y = 0 : y^2 + x = 0 \\
        y \neq 0 : y^2 + x \neq 0
    \end{array} \right. \\
    x \neq 0 \ldots
\end{cases}
\end{align*}
\]

$\Rightarrow$

\[
\begin{align*}
\begin{cases}
    x = 0 \\
    \left\{ \begin{array}{l}
        y = 0 : y^2 + x = 0 \land y^2 + y = 0 \\
        y \neq 0 : y^2 + x \neq 0
    \end{array} \right. \\
    x \neq 0 \ldots
\end{cases}
\end{align*}
\]
Refine the next path of the tree with $y^2 + y$

$$
\begin{align*}
\left\{ \begin{array}{l}
x = 0 \\
y = 0 : & y^2 + x = 0 \land y^2 + y = 0 \\
y \neq 0 : & y^2 + x \neq 0
\end{array} \right.
\end{align*}
\quad \Rightarrow
\begin{align*}
\left\{ \begin{array}{l}
x = 0 \\
y = 0 : & y^2 + x = 0 \land y^2 + y = 0 \\
y = -1 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
\text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{array} \right.
\end{align*}
\quad \Rightarrow
\begin{align*}
\left\{ \begin{array}{l}
x = 0 \\
y = 0 : & y^2 + x = 0 \land y^2 + y = 0 \\
y = -1 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
\text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{array} \right.
\end{align*}
\quad \Rightarrow
\begin{align*}
\left\{ \begin{array}{l}
x = 0 \\
y = 0 : & y^2 + x = 0 \land y^2 + y = 0 \\
y = -1 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
\text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{array} \right.
\end{align*}
The \( \{y^2 + x, y^2 + y\} \) sign invariant cylindrical tree of \( \mathbb{C}^2 \)

\[
\begin{align*}
\text{\( x = 0 \)} & \quad \begin{cases}
  y = 0 : & y^2 + x = 0 \land y^2 + y = 0 \\
  y = -1 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
  \text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{cases} \\
\text{\( x = -1 \)} & \quad \begin{cases}
  y = -1 : & y^2 + x = 0 \land y^2 + y = 0 \\
  y = 1 : & y^2 + x = 0 \land y^2 + y \neq 0 \\
  y = 0 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
  \text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{cases} \\
\text{otherwise} & \quad \begin{cases}
  y^2 + x = 0 : & y^2 + x = 0 \land y^2 + y \neq 0 \\
  y^2 + y = 0 : & y^2 + x \neq 0 \land y^2 + y = 0 \\
  \text{otherwise} : & y^2 + x \neq 0 \land y^2 + y \neq 0
\end{cases}
\end{align*}
\]
1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Compute partial cylindrical tree

A partial cylindrical tree induced by the $F := \{y^2 + x = 0, y^2 + y = 0\}$ is

```
  x = 0       x + 1 = 0
    /         /  \
   /           /    \
  y = 0       y + 1 = 0
```
Transform a complex cylindrical decomposition to a real one

Complex:

\[
\begin{align*}
& x = 0 \quad \left\{ \begin{array}{l}
y = 0 : y^2 + x = 0 \\
y \neq 0 : y^2 + x \neq 0
\end{array} \right.
\\
& x \neq 0 \quad \left\{ \begin{array}{l}
y^2 + x = 0 : y^2 + x = 0 \\
y^2 + x \neq 0 : y^2 + x \neq 0
\end{array} \right.
\end{align*}
\]

Real:

\[
\begin{align*}
& x < 0 \quad \left\{ \begin{array}{l}
y < -\sqrt{|x|} : y^2 + x > 0 \\
y = -\sqrt{|x|} : y^2 + x = 0 \\
y > -\sqrt{|x|} \land y < \sqrt{|x|} : y^2 + x < 0 \\
y = \sqrt{|x|} : y^2 + x = 0 \\
y > \sqrt{|x|} : y^2 + x > 0
\end{array} \right.
\\
& x = 0 \quad \left\{ \begin{array}{l}
y < 0 : y^2 + x > 0 \\
y = 0 : y^2 + x = 0 \\
y > 0 : y^2 + x > 0
\end{array} \right.
\\
& x > 0 \quad \text{for any } y : y^2 + x > 0
\end{align*}
\]
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Propagate the truth values

\((\exists y) f(x, y) \geq 0\)

\((\forall y) f(x, y) \geq 0\)
Generate the solution formula

For the right example, to distinguish the true and false cells, one refines the tree w.r.t. the derivative of $x^2 - 2$. In general, one applies Thom’s lemma.
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Implementation in Maple

The universe tree is always up-to-date

A sub-tree evolves with the universe tree
## Information of the test examples

<table>
<thead>
<tr>
<th>System</th>
<th>nvars</th>
<th>degree</th>
<th>nsys</th>
<th>nterms</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlkashiSinus</td>
<td>9</td>
<td>3</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>Alonso</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>Arnborg-Lazard-rev</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>Barry</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>blood-coagulation-2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Bronstein-Wang</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>cdc2-cyclin</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>circles</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>72</td>
</tr>
<tr>
<td>genLinSyst-3-2</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>GonzalezGonzalez</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>lhlp2</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>lhlp5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>MontesS10</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>MontesS12</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>MontesS15</td>
<td>12</td>
<td>2</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>MontesS4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>MontesS5</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>MontesS7</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>MontesS9</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>nql-5-4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>r-5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>r-6</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>Rose</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>Wang93</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>15</td>
</tr>
</tbody>
</table>
tcd-rec : our ISSAC’09 recursive algorithm

tcd-inc : the incremental algorithm
tcd-inc: the incremental algorithm with set of polynomials as input
tcd-eqs: the incremental algorithm with set of equations as input
tcad-rec: our ISSAC'09 recursive algorithm

tcad-inc: the incremental algorithm
tcad-rec: our ISSAC’09 recursive algorithm

tcad-inc: the incremental algorithm
Outline

1. First Idea: Introduce Case Discussion
2. Second Idea: Compute the Decomposition Incrementally
3. Third Idea: Compute CAD of a Variety
4. QE via TCAD
5. Implementation and Benchmark
6. Application
Stability of control system

www.maplesoft.com/industries/automotive/vehicle_dynamics
Control Lyapunov function

The control system

- $\dot{x} = f(x, u),$
- $x \in \mathbb{R}^n,$
- $u$ implicitly depends on $x$ and $t.$

Control Lyapunov function

A function $V(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfying

- $V(x)$ is positive definite, that is $V(0) = 0$ and $\forall x \neq 0, V(x) > 0.$
- $\dot{V}(0) = 0$ and $\forall x \neq 0,$ $\exists u,$ such that $\dot{V} < 0,$ where $\dot{V} = \frac{\partial V}{\partial x} \cdot f(x, u).$
- $V$ is radially unbounded, that is $\|x\| \rightarrow \infty$ implies that $V \rightarrow \infty.$
Find control Lyapunov function by QeTcad

Consider a dynamical system with variables \( x, y \) and a control parameter \( u \).

We'd like to know if there exists a control Lyapunov function of the form \( \alpha x^2 + by^2 \).

The equivalent QE problem is: \( \forall (x, y) \exists (u) \left( x \neq 0 \text{ or } y \neq 0 \Rightarrow V > 0 \land \frac{d}{dt} V < 0 \right) \).

The input control system.

\[
S := \{ \text{diff}(x(t), t) = -x(t) + u, \ \text{diff}(y(t), t) = -x(t) - y(t)^3 \};
\]

\[
S := \left[ \frac{d}{dt} x(t) = -x(t) + u, \ \frac{d}{dt} y(t) = -x(t) - y(t)^3 \right]
\]

Compute \( \frac{d}{dt} V \).

\[
\begin{align*}
fx & := -x + u; 
fy & := -y^3; 
V & := ax^2 + by^2; 
V_t & := \text{diff}(V, x)*fx + \text{diff}(V, y)*fy;
\end{align*}
\]

\[
\begin{align*}
fx & := -x + u \\
fy & := -y^3 - x \\
V & := ax^2 + by^2 \\
V_t & := 2ax(-x + u) + 2by(-y^3 - x)
\end{align*}
\]

Call QeTcad to find conditions on \( a \) and \( b \).

\[
\begin{align*}
& \text{QuantifierElimination}([x,y], [u], (x<0) \ \&\& (y<0) \ \&\& (V>0) \ \&\& (V_t<0)); \\
& 0 < b \ \&\& 0 < a
\end{align*}
\]

Verify the result for \( a = 1, b = 4 \).

Indeed \( x^2 + 4y^2 \) is a control Lyapunov function.

\[
\begin{align*}
& \text{simplify}([a=1,b=4], V_t); \text{subs}(u=4*y, \%); \\
& -8y^4 + 2ux - 2x^2 - 8xy \\
& -8y^4 - 2x^2
\end{align*}
\]
Conclusion

- We presented an incremental algorithm for computing CADs.
- The core operation of our algorithm is an *Intersect* operation, which refines a complex cylindrical tree by means of a polynomial constraint.
- The Intersect operation provides a systematic solution for propagating equational constraints.
- For many examples, the incremental outperforms both QEPCAD and Mathematica as well as our previous recursive algorithm.
- We have developed a QE routine QETCAD based on TCAD.

Work in progress

- We are working on different optimizations for both TCAD and QETCAD.
## Table: Timings for computing CAD

<table>
<thead>
<tr>
<th>System</th>
<th>nvars</th>
<th>degree</th>
<th>nsys</th>
<th>nterms</th>
<th>qepcad</th>
<th>qepcad-eqs</th>
<th>math-eqs</th>
<th>tcad</th>
<th>tcad-eqs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlkashiSinus</td>
<td>9</td>
<td>3</td>
<td>6</td>
<td>21</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>2.232</td>
<td>&gt; 1 h</td>
<td>58.775</td>
</tr>
<tr>
<td>Alonso</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>7.516</td>
<td>5.284</td>
<td>0.74</td>
<td>61.591</td>
<td>5.776</td>
</tr>
<tr>
<td>Arnborg-Lazard-rev</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>21</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>0.952</td>
<td>&gt; 1 h</td>
<td>17.325</td>
</tr>
<tr>
<td>Barry</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>9</td>
<td>Fail</td>
<td>216.425</td>
<td>0.032</td>
<td>8.580</td>
<td>1.004</td>
</tr>
<tr>
<td>blood-coagulation-2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>15</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>985.709</td>
<td>7.260</td>
<td>2.564</td>
</tr>
<tr>
<td>Bronstein-Wang</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>22</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>333.892</td>
<td>503.863</td>
<td>40.902</td>
</tr>
<tr>
<td>cdc2-cyclin</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>8</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>21.633</td>
<td>5.996</td>
<td>41.211</td>
</tr>
<tr>
<td>circles</td>
<td>2</td>
<td>10</td>
<td>2</td>
<td>72</td>
<td>21.633</td>
<td>5.996</td>
<td>41.211</td>
<td>&gt; 1 h</td>
<td>6.588</td>
</tr>
<tr>
<td>genLinSyst-3-2</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>Fail</td>
<td>Fail</td>
<td>217.062</td>
<td>3013.764</td>
<td>1.136</td>
</tr>
<tr>
<td>GonzalezGonzalez</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>10.528</td>
<td>10.412</td>
<td>0.012</td>
<td>214.213</td>
<td>0.952</td>
</tr>
<tr>
<td>Ihlp2</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>16</td>
<td>960.756</td>
<td>5.076</td>
<td>0.016</td>
<td>3.124</td>
<td>0.512</td>
</tr>
<tr>
<td>Ihlp5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>22</td>
<td>10.300</td>
<td>10.068</td>
<td>0.016</td>
<td>35.338</td>
<td>1.084</td>
</tr>
<tr>
<td>MontesS10</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>14</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>22.797</td>
</tr>
<tr>
<td>MontesS12</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>330.996</td>
</tr>
<tr>
<td>MontesS15</td>
<td>12</td>
<td>2</td>
<td>7</td>
<td>19</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>0.004</td>
<td>&gt; 1 h</td>
<td>395.964</td>
</tr>
<tr>
<td>MontesS4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>11</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>0.004</td>
<td>2682.391</td>
<td>0.888</td>
</tr>
<tr>
<td>MontesS5</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>19</td>
<td>Fail</td>
<td>Fail</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>9.400</td>
</tr>
<tr>
<td>MontesS7</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>12</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>245.807</td>
<td>&gt; 1 h</td>
<td>4.944</td>
</tr>
<tr>
<td>MontesS9</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>12</td>
<td>Fail</td>
<td>Fail</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>110.902</td>
</tr>
<tr>
<td>nql-5-4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>14</td>
<td>93.073</td>
<td>5.420</td>
<td>1303.07</td>
<td>113.675</td>
<td>1.004</td>
</tr>
<tr>
<td>r-5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>18</td>
<td>&gt; 1 h</td>
<td>1802.676</td>
<td>0.016</td>
<td>1282.928</td>
<td>1.208</td>
</tr>
<tr>
<td>r-6</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>22</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>0.024</td>
<td>&gt; 1 h</td>
<td>1.500</td>
</tr>
<tr>
<td>Rose</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>29</td>
<td>Fail</td>
<td>&gt; 1 h</td>
<td>606.361</td>
<td>3.136</td>
<td>152.673</td>
</tr>
<tr>
<td>Wang93</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>15</td>
<td>Fail</td>
<td>Fail</td>
<td>&gt; 1 h</td>
<td>&gt; 1 h</td>
<td>152.673</td>
</tr>
</tbody>
</table>