Global Consistencies in Boolean Satisfiability

RNDr. Pavel Surynek, Ph.D.
Department of Theoretical Computer Science and Mathematical Logic
Faculty of Mathematics and Physics
Charles University in Prague

http://ktiml.mff.cuni.cz/~surynek
Constraint Satisfaction Problem (CSP)

- Constraint satisfaction problem over the universe of elements \( \mathbb{D} \) is a triple \((X, C, D)\)
  - \( X \) – finite set of variables
  - \( C \) – finite set of constraints
  - \( D \) – is a function \( D : X \rightarrow \mathcal{P}(\mathbb{D}) \)
  - each constraint \( c \in C \) is a construct of the form \(<(x_1^c, x_2^c, \ldots, x_{k(c)}^c), R^c>\)
    - \( k(c) \) is arity of the constraint
    - \( x_i^c \in X \) for \( i = 1, 2, \ldots, k(c) \) and \( R^c \subseteq D(x_1^c) \times D(x_2^c) \times \ldots \times D(x_{k(c)}^c) \)

- The task is to find **assignment of values to variables** from their domains such that all the constraints are satisfied
  - or decide that **no** such **valuation** exists

- Decision variant is an **NP-complete** problem

**Example:**
- \( \mathbb{D} = \{1, 2, 3\} \)
- \( X = \{a, b, c\} \)
- \( C = \{<(a, b), <">; <(b, c), "=">\} \)
- \( D(a) = D(b) = D(c) = \mathbb{D} \)

- **Example:** \( a = 1, b = 2, c = 3 \)
A Boolean formula is given - variables can take either the value **TRUE** or **FALSE**

The task is to find *valuation of variables* such that the formula is **satisfied**

- or decide that no such valuation exists

**Conjunctive normal form (CNF) - standard form of the input formula**

- **variables**: $x_1, x_2, x_3, ...$
- **literals**: $x_1, \neg x_1, x_2, \neg x_2$, ... variable or its negation
- **clauses**: $(x_1 \lor \neg x_2 \lor \neg x_3)$ ... disjunction of literals
- **formula**: $(x_1 \lor \neg x_2) \land (x_1 \lor x_2 \lor \neg x_3)$ ... conjunction of clauses

Clauses represent constraints that must be all satisfied (can be regarded as CSP) – SAT and CSP are mutually reducible
Motivation for Global Consistencies

• CSP paradigm provides many types of **local consistencies**
  ▫ local inference is typically **too weak** for SAT
  ▫ arc-consistency, path-consistency, i,j-consistency
    • insignificant gain in comparison with unit-propagation
    • expensive propagation with respect to the inference strength

• **Global** consistencies (global constraints)
  ▫ provide strong global inference
    • often leads to significant simplification of the problem
  ▫ application of **global consistencies** in SAT is quite rare

• Consistency based on **structural properties**
  ▫ interpret SAT as a graph and find graph structures
Difficult Instances of SAT

- **Difficult** instances for **today’s SAT** (more precisely for 2007’s) solving systems
  - impossible to (heuristically) **guess** the solution
  - heuristics do not succeed ►► search
  - clause learning mechanism needs to learn for a long time

- Typical example: **unsatisfiable SAT** instances encoding Dirichlet’s box principle (**Pigeon-hole principle**)

- **Satisfiable** case
  - **Valuation** of variables = certificate
  - **small witness** through which we can verify satisfiability

- **Unsatisfiable** case
  - no (small) witness (certificate) to guess
  - search/learning is necessary

Today’s new **variable ordering heuristics** and **preprocessing** techniques can succeed on these types of instances.
Our Approach – conflict graphs

- **Input** - Boolean formula in CNF
- **Interpret** as a graph of conflicts
  - vertices = literals
  - edges = conflicts between literals
  - *example:* $x$ and $\neg x$ are in conflict (cannot be satisfied together) ➤➤ put an edge between corresponding vertices
- **Perform initial preprocessing**
  - Singleton unit propagation ➤➤ new conflicts
  - Consistency based on conflict graph
- **Output** - equivalent (simpler) formula or the answer “unsatisfiable”
Initial Preprocessing – improve the graph

• Make the graph of conflicts **dense**
  ▫ apply **singleton unit propagation**
  ▫ discover **hidden** conflicts between literals
  ▫ **denser** conflict graph = **better** for the subsequent step

• (Greedily) **find cliques** in the conflict graph
  ▫ **at most one literal** from a clique can be satisfied
  ▫ contribution of literal $x$...$c(x)$ = number of clauses containing $x$
  ▫ contribution of clique $C$...$c(C) = \max_{x \in C} c(x) $
  ▫ $\sum_{C \in \text{cliques}} c(C) < \text{number of clauses}$ (basic consistency check)

• All the cliques together do not contribute enough to satisfy the input formula ►► the input formula is **unsatisfiable**
Clique Consistency – making projections

- Generalization of “∑_{C ∈ cliques} c(C) < #clauses”
- Choose a sub-formula \( B = \) subset of clauses and project the contribution counting on sub-formula
  - contribution of literal \( x \) to sub-formula \( B \) ...
    ...\( c(x, B) = \) number of clauses of \( B \) containing \( x \)
  - contribution of clique \( C \) to sub-formula \( B \) ...
    ...\( c(C, B) = \max_{x ∈ C} c(x, B) \)
  - when \( ∑_{C ∈ cliques} c(C, B) < \) number of clauses in \( B \)
    ►►\( B \) is unsatisfiable ⇒ input formula is unsatisfiable
- Singleton approach...literal \( x \) is inconsistent
  - \( ∑_{C ∈ cliques} \neg x c(C, B) < (#clauses of B) - c(x, B) \)
Clique Consistency (example)

- Inconsistency (basic case – singleton approach is not applied):
  \[\sum_{C \in \text{cliques}} c(C,B) < \#\text{clauses in } B\]
  - example: clique \( C_1 = \{a, b, c\} \)
    clique \( C_2 = \{p, q, r\} \)
    - \( \{a, b, c\} \) are pair-wise conflicting
    - \( \{p, q, r\} \) are pair-wise conflicting
  - sub-formula
    \[ B = (a \lor p) \land (b \lor q) \land (c \lor r) \]
    \[ c(C_1, B) = 1; c(C_2, B) = 1 \]
  - \( \sum_{C \in \text{cliques}} c(C,B) = 2; \#\text{clauses in } B = 3 \)
- The original formula has no satisfying valuation.
Visualization (1)
using GraphExplorer software (Surynek, 2007-2010)

- „Insert 7 pigeons into 6 holes“
Visualization (2)
using GraphExplorer software (Surynek, 2007-2010)

- After inferring **new conflicts** – **singleton UP**
Visualization (3)

using GraphExplorer software (Surynek, 2007-2010)

- After enforcing **clique consistency**: UNSAT
Complexity of Clique Consistency

• Construction of graph of conflicts
  ▫ **polynomial** worst-case time
• Singleton unit propagation
  ▫ **polynomial** worst-case time
  ▫ however, may be too time consuming for large real-life problems
    • efficient propagation scheme base on 2-literal watching must be used
• Clique consistency with respect to a single sub-formula
  ▫ **polynomial**
• **Problem:** clique consistency with respect to multiple sub-formulae
  ▫ we cannot try all the sub-formulae
  ▫ intelligent selection of promising sub-formulae must be done
Competitive Comparison

carried out in 2007

• **Tested** SAT solving systems
  - MiniSAT
  - zChaff
  - HaifaSAT
  - selection criterion: *available source code*

• **Testing instances** (by Fadi Aloul)
  - Pigeon Hole Principle
  - Urquhart (resists resolution method)
  - Field Programmable Gate Array
## Experimental Evaluation

<table>
<thead>
<tr>
<th>Instance</th>
<th>Decision (seconds)</th>
<th>Speedup ratio w.r.t. MiniSAT</th>
<th>Speedup ratio w.r.t. zChaff</th>
<th>Speedup ratio w.r.t. HaifaSAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>chnl10_11</td>
<td>0.43</td>
<td>79.76</td>
<td>17.53</td>
<td>&gt; 1395.34</td>
</tr>
<tr>
<td>chnl10_12</td>
<td>0.60</td>
<td>169.68</td>
<td>8.51</td>
<td>&gt; 1000.00</td>
</tr>
<tr>
<td>chnl10_13</td>
<td>0.78</td>
<td>256.79</td>
<td>14.70</td>
<td>&gt; 769.23</td>
</tr>
<tr>
<td>chnl11_12</td>
<td>0.70</td>
<td>&gt; 857.14</td>
<td>47.84</td>
<td>&gt; 857.14</td>
</tr>
<tr>
<td>urq3_5</td>
<td>130.15</td>
<td>0.73</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>urq4_5</td>
<td>&gt; 600.00</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>urq5_5</td>
<td>&gt; 600.00</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>urq6_5</td>
<td>&gt; 600.00</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>hole9</td>
<td>0.08</td>
<td>45.5</td>
<td>18.25</td>
<td>5977.00</td>
</tr>
<tr>
<td>hole10</td>
<td>0.13</td>
<td>301.84</td>
<td>57.92</td>
<td>&gt; 4615.38</td>
</tr>
<tr>
<td>hole11</td>
<td>0.20</td>
<td>&gt; 3000.00</td>
<td>161.8</td>
<td>&gt; 3000.00</td>
</tr>
<tr>
<td>hole12</td>
<td>0.30</td>
<td>&gt; 2000.00</td>
<td>1240.6</td>
<td>&gt; 2000.00</td>
</tr>
<tr>
<td>fpga10_11</td>
<td>0.46</td>
<td>97.32</td>
<td>27.34</td>
<td>&gt; 1304.34</td>
</tr>
<tr>
<td>fpga10_12</td>
<td>0.64</td>
<td>186.34</td>
<td>52.84</td>
<td>&gt; 937.50</td>
</tr>
<tr>
<td>fpga10_13</td>
<td>0.84</td>
<td>431.23</td>
<td>90.65</td>
<td>&gt; 714.28</td>
</tr>
<tr>
<td>fpga10_15</td>
<td>1.39</td>
<td>&gt; 431.65</td>
<td>197.72</td>
<td>&gt; 431.65</td>
</tr>
</tbody>
</table>

Opteron 1600 MHz, Mandriva Linux 10.1
Path-consistency in Literal Encoding (1)

- SAT as CSP: **Literal encoding** model \((X,C,D)\)
  - \(X\) ... variables \(\leftrightarrow\) clauses, \(C\) ... constraints \(\leftrightarrow\) values standing for complementary literals are forbidden, \(D\) ... variable domains \(\leftrightarrow\) literals

- Interpret path-consistency in the CSP model of SAT as a **directed graph**
  - **vertices** \(\leftrightarrow\) values in domains, **edges** \(\leftrightarrow\) allowed pairs of values

\[
\begin{align*}
V_{(\neg x_1 \lor \neg x_2)} & \quad V_{(\neg x_2 \lor \neg x_3)} & \quad V_{(\neg x_3 \lor \neg x_1)} \\
\neg x_1 & \quad x_1 & \quad \neg x_2 & \quad \neg x_2 & \quad x_2 & \quad \neg x_3 & \quad \neg x_3 & \quad x_3 \\
\lor & \quad \lor & \quad \lor & \quad \lor & \quad \lor & \quad \lor & \quad \lor & \quad \lor \\
\neg x_1 & \quad x_1 & \quad \neg x_2 & \quad x_2 & \quad \neg x_3 & \quad x_3 & \quad \neg x_1 & \quad \neg x_2 \\
\end{align*}
\]

*example:* 
\[X = V_{(\neg x_1 \lor \neg x_2)}, V_{(x_1 \lor x_2)}, \ldots\]

*example:* 
\[D(V_{(\neg x_1 \lor \neg x_2)}) = \{\neg x_1, \neg x_2\}\]

*example:* 
\[V_{(\neg x_1 \lor \neg x_2)} = \neg x_1\] and 
\[V_{(x_1 \lor x_2)} = x_1\] is **forbidden**
Path-consistency in Literal Encoding (2)

- Let us have a **sequence of variables (path)**
  - pair of values is **path-consistent** w.r.t. to the sequence if there is an oriented path connecting them in the graph interpretation going through the sequence and values itself are connected
- **Ignores** constraints between non-neighboring variables in the sequence of variables
**Modified Path-Consistency for SAT**

- Deduce **more information from constraints**
  - decompose values into **disjoint sets** (called layers ... $L_1, L_2, ..., L_M$)
  - deduce more information from constraints - calculate maximum size of the intersection of the constructed path with individual layers – denoted as $\chi$
- Stronger restriction on paths ▶ **stronger propagation**

![Diagram](image.png)

path ending in this vertex cannot intersect with $L_1$ in more than two values
NP-completeness of the Modified Path Consistency

- **Enforcing** modified path-consistency **is difficult** (NP-complete)
  - The decision problem is whether there exists a path respecting the maximum sizes of intersections with individual layers.

- **Lemma:** The decision variant of the problem belongs to the NP class.
  - The path is of polynomial size with respect to the graph interpretation.
  - It can be checked in polynomial time whether the path conforms to maximum size of intersections with individual layers.

- **Lemma:** The existence of a Hamiltonian path in a graph is reducible to the existence of a path conforming to the maximum sizes of intersections with layers.

**Main idea** of the proof: $G=(V,E)$, where $V=\{v_1,v_2,\ldots,v_n\}$

(i) Construct an instance of modified path consistency in the form of a matrix

(ii) Associate rows of the matrix with layers and set the maximum size of the intersection to 1
Intersection Matrices

• An intersection matrix is defined for each value in the graph interpretation of path-consistency – it is denoted as $\psi(v)$
  ▫ Let $L_1, L_2, ..., L_M$ be a layer decomposition of the graph interpretation
  ▫ Let $K$ be the number of variables involved in the path
  ▫ ► The intersection matrix is of type $M \times K$
• Intersection matrix $\psi(v)$ w.r.t. a pair of values $v_0$ and $v_K$
  ▫ $\psi(v)_{i,j}$ represents the number of paths starting in $v_0$ and ending in $v$ that partially conforms to maximum sizes of intersection with layers such that they intersect with $L_i$ j-times.
• It is not possible to enforce exact conformity to calculated maximum sizes of intersection with layers
  ▫ Therefore we need to talk about partial conformity.
Intersection Matrices Update

- **Intersection matrix** can be updated easily
  - $\Psi(v)$ is calculated from $\Psi(u_1), \Psi(u_2), ..., \Psi(u_m)$ where $u_1, u_2, ..., u_m$ are values from the domain of the previous variable in the path
- If it is detected that no of the paths starting in $v_0$ and ending in $v$ conforms to the maximum size of the intersection with the layer $L_i$ such that $v \in L_i$ then $\Psi(v)$ is set to 0 (matrix)
  - maximum intersection sizes with other layers cannot be violated since intersection size with them does no change
  - **relaxation**: paths that do not conform to maximum sizes of intersections with layers are propagated further
Visualization of Layers
using GraphExplorer software (Surynek, 2007-2010)

- Layer decomposition was constructed with several **most constrained clauses** (now: edges = **forbidden** pairs)
  - several benchmark problems from the **SAT Library**

![hanoi4.cnf](image1)

![jnh1.cnf](image2)

![s3-3-3-8.cnf](image3)
Maximum Intersection Sizes

- Maximum intersection size is calculated using the maximum intersection size for the previous value in the layer
  - it is checked whether the intersection size can be increased by adding the current value

<table>
<thead>
<tr>
<th>SAT instance</th>
<th>Maximum intersection with $L_1={v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ais12.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 1 &amp; 1 &amp; 1 &amp; 1 &amp; 1 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>hanoi4.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 3 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>huge.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>jnh1.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 1 &amp; 2 &amp; 2 &amp; 4 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>par16-1.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 &amp; 5 \end{bmatrix}$</td>
</tr>
<tr>
<td>par16-1-c.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 \end{bmatrix}$</td>
</tr>
<tr>
<td>pret150_75.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>s3-3-3-8.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>ssa7552-160.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 3 &amp; 4 &amp; 4 &amp; 5 &amp; 6 \end{bmatrix}$</td>
</tr>
<tr>
<td>sw100-5.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 3 \end{bmatrix}$</td>
</tr>
<tr>
<td>Urq8_5.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
<tr>
<td>uuf250-0100.cnf</td>
<td>$\begin{bmatrix} 1 &amp; 1 &amp; 2 &amp; 2 &amp; 3 &amp; 3 &amp; 4 &amp; 4 \end{bmatrix}$</td>
</tr>
</tbody>
</table>
Experimental Evaluation (1)

<table>
<thead>
<tr>
<th>SAT Problem</th>
<th>Number of variables</th>
<th>Number of clauses</th>
<th>Pairs filtered by standard PC</th>
<th>Pairs filtered by modified PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>bw_large.a</td>
<td>495</td>
<td>4675</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>hanoi4</td>
<td>718</td>
<td>4934</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>huge</td>
<td>459</td>
<td>7054</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>jnh2</td>
<td>100</td>
<td>850</td>
<td>135</td>
<td>147</td>
</tr>
<tr>
<td>logistics.a</td>
<td>828</td>
<td>6718</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td>medium</td>
<td>116</td>
<td>953</td>
<td>177</td>
<td>227</td>
</tr>
<tr>
<td>par8-1-c</td>
<td>64</td>
<td>254</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>par8-2-c</td>
<td>68</td>
<td>270</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>par8-3-c</td>
<td>75</td>
<td>298</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>par16-1-c</td>
<td>317</td>
<td>1264</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>par16-2-c</td>
<td>349</td>
<td>1392</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>par16-3-c</td>
<td>334</td>
<td>1332</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>ssa0432/003</td>
<td>435</td>
<td>1027</td>
<td>81</td>
<td>1598</td>
</tr>
<tr>
<td>ssa2670/130</td>
<td>1359</td>
<td>3321</td>
<td>4</td>
<td>2656</td>
</tr>
<tr>
<td>ssa2670/141</td>
<td>986</td>
<td>2315</td>
<td>20</td>
<td>8871</td>
</tr>
<tr>
<td>ssa7552/038</td>
<td>1501</td>
<td>3575</td>
<td>16</td>
<td>5652</td>
</tr>
<tr>
<td>ssa7552/158</td>
<td>1363</td>
<td>3034</td>
<td>49</td>
<td>2371</td>
</tr>
</tbody>
</table>

- Comparison of the number of **filtered pairs of values**
  - several benchmark problems from the **SAT Library**
  - comparison of **PC and modified PC** enforced by the basic variant of intersection matrix update algorithm
  - **on some problems** modified PC is significantly **stronger**
  - runtime was slightly higher for modified PC
**Experimental Evaluation (2)**

- Improvement ratio gained by preprocessing of SAT problems by modified PC in comparison with PC
  - the number of decision steps was measured
  - some problems were successfully preprocessed by modified PC

<table>
<thead>
<tr>
<th>Problem</th>
<th>#variables</th>
<th>#clauses</th>
<th>HaifaSat</th>
<th>Minisat2</th>
<th>Rsat_1_03</th>
<th>zChaff</th>
</tr>
</thead>
<tbody>
<tr>
<td>bw_large.a</td>
<td>459</td>
<td>4675</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>hanoi4</td>
<td>718</td>
<td>4934</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>hanoi5</td>
<td>1931</td>
<td>14468</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>huge</td>
<td>459</td>
<td>7054</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>jnh2</td>
<td>100</td>
<td>850</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.3</td>
</tr>
<tr>
<td>logistics.a</td>
<td>828</td>
<td>6718</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>medium</td>
<td>116</td>
<td>953</td>
<td>1.0</td>
<td>1.0</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>par8-1-c</td>
<td>64</td>
<td>254</td>
<td>1.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.7</td>
</tr>
<tr>
<td>par8-2-c</td>
<td>68</td>
<td>270</td>
<td>0.9</td>
<td>1.2</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>par8-3-c</td>
<td>75</td>
<td>298</td>
<td>0.8</td>
<td>1.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>par16-1-c</td>
<td>317</td>
<td>1264</td>
<td>0.1</td>
<td>0.4</td>
<td>2.2</td>
<td>0.1</td>
</tr>
<tr>
<td>par16-2-c</td>
<td>349</td>
<td>1392</td>
<td>1.1</td>
<td>2.3</td>
<td>0.8</td>
<td>0.8</td>
</tr>
<tr>
<td>par16-3-c</td>
<td>334</td>
<td>1332</td>
<td>0.8</td>
<td>1.4</td>
<td>6.6</td>
<td>1.6</td>
</tr>
<tr>
<td>ssa0432-003</td>
<td>435</td>
<td>1027</td>
<td>1.0</td>
<td>228.0</td>
<td>155.0</td>
<td>122.0</td>
</tr>
<tr>
<td>ssa2670-130</td>
<td>1359</td>
<td>3321</td>
<td>51.0</td>
<td>411.0</td>
<td>371.0</td>
<td>323.0</td>
</tr>
<tr>
<td>ssa2670-141</td>
<td>986</td>
<td>2315</td>
<td>289.0</td>
<td>429.0</td>
<td>455.0</td>
<td>489.0</td>
</tr>
<tr>
<td>ssa7552-038</td>
<td>1501</td>
<td>3575</td>
<td>190.0</td>
<td>226.0</td>
<td>173.0</td>
<td>238.0</td>
</tr>
<tr>
<td>ssa7552-158</td>
<td>1363</td>
<td>3034</td>
<td>114.0</td>
<td>129.0</td>
<td>151.0</td>
<td>312.0</td>
</tr>
</tbody>
</table>
References