A Faster Deterministic Algorithm For 3-SAT

Dominik Scheder

8th Latin American Theoretical Informatics Symposium
April 7, 2008
Overview

1. The SAT problem and some algorithms
2. The searchball algorithm
3. Improving searchball


\((x \lor y \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor \bar{z})\)
SAT – What Is It About?

\[(x \lor y \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor \bar{z})\]

\[\{\{x, y, z\}, \{x, \bar{y}, \bar{z}\}, \{\bar{x}, y, \bar{z}\}\}\]
(x \lor y \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor \bar{z})

\{\{x, y, z\}, \{x, \bar{y}, \bar{z}\}, \{\bar{x}, \bar{y}, \bar{z}\}\}

Assignment \(\alpha\): \(x = 0, y = 1, z = 1\)
(x \lor y \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor \bar{z})

\{\{x, y, z\}, \{x, \bar{y}, \bar{z}\}, \{\bar{x}, \bar{y}, \bar{z}\}\}

Assignment \alpha: x = 0, y = 1, z = 1

Assignment \beta: x = 1, y = 1, z = 0
(x ∨ y ∨ z) ∧ (x ∨ y ∨ z) ∧ (x ∨ y ∨ z)

Assignment $\alpha$: $x = 0$, $y = 1$, $z = 1$

Assignment $\beta$: $x = 1$, $y = 1$, $z = 0$

$\beta$ satisfies the formula, $\alpha$ does not.
A Recursive Algorithm

\[ \{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\} \]
A Recursive Algorithm

\{\{x, y, z\}, \{\bar{x}, y, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\}
A Recursive Algorithm

\[ \{\{x, y, z\}, \{\overline{x}, \overline{y}, z\}, \{x, y, \overline{z}\}, \{\overline{x}, y, z\}\} \]
A Recursive Algorithm

\{\{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\}

\begin{itemize}
  \item \(x = 0\)
  \item \(y = 0\)
  \item \(z = 0\)
\end{itemize}
A Recursive Algorithm

\{ \{ x, y, z \}, \{ x, \bar{y}, z \}, \{ x, y, \bar{z} \}, \{ \bar{x}, y, z \} \}
{\{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\}
A Faster Deterministic Algorithm For 3-SAT

\{\{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\}

Diagram showing the recursive algorithm with decision points based on \(x, y, z\) values.
A Recursive Algorithm

\{\{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\}

Satisfying Assignment: \(x = 0, y = 1, z = 1\)
<p>| $2^n$ | What I just showed you | deterministic |
| 1.618^n | Monien and Speckenmeyer | 1985 | deterministic |
| 1.588^n | Paturi, Pudlák and Zane | 1997 | randomized |
| 1.5^n | Dantsin et. al. | 2002 | deterministic |
| 1.481^n | Dantsin et. al. | 2002 | deterministic |
| 1.473^n | Brueggemann and Kern | 2004 | deterministic |
| 1.465^n | S. | 2008 | deterministic |
| 1.333^n | Schöning | 2002 | randomized |</p>
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Year</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^n$</td>
<td>What I just showed you</td>
<td></td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.618^n$</td>
<td>Monien and Speckenmeyer</td>
<td>1985</td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.588^n$</td>
<td>Paturi, Pudlák and Zane</td>
<td>1997</td>
<td>randomized</td>
</tr>
<tr>
<td>$1.5^n$</td>
<td>Dantsin et. al.</td>
<td>2002</td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.481^n$</td>
<td>Dantsin et. al.</td>
<td>2002</td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.473^n$</td>
<td>Brueggemann and Kern</td>
<td>2004</td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.465^n$</td>
<td>S.</td>
<td>2008</td>
<td>deterministic</td>
</tr>
<tr>
<td>$1.333^n$</td>
<td>Schöning</td>
<td>2002</td>
<td>randomized</td>
</tr>
</tbody>
</table>
Set of truth assignments $= \{0, 1\}^n = n$-dimensional cube

Decompose into two $(n - 1)$-dimensional cubes
Idea of Local Search

Cover $\{0, 1\}^n$ by several Hamming balls

$$B_r(x) = \{ y \in \{0, 1\}^n \mid d_H(x, y) \leq r \}$$
Idea of Local Search

Cover \(\{0, 1\}^n\) by several Hamming balls

\[ B_r(x) = \{ y \in \{0, 1\}^n | d_H(x, y) \leq r \} \]
Idea of Local Search

Cover \( \{0, 1\}^n \) by several Hamming balls

\[
B_r(x) = \{ y \in \{0, 1\}^n \mid d_H(x, y) \leq r \}
\]
Idea of Local Search

Cover \( \{0, 1\}^n \) by several Hamming balls

\[ B_r(x) = \{ y \in \{0, 1\}^n \mid d_H(x, y) \leq r \} \]
Cover \( \{0, 1\}^n \) by several Hamming balls

\[
B_r(x) = \{y \in \{0, 1\}^n | d_H(x, y) \leq r\}
\]
Idea of Local Search

Cover \( \{0, 1\}^n \) by several Hamming balls

\[
B_r(x) = \{ y \in \{0, 1\}^n | d_H(x, y) \leq r \}
\]
Choosing the Optimal Radius

Should we take many small balls or a few large ones?
Choosing the Optimal Radius

Should we take many small balls or a few large ones?

Suppose we need $O(a^r)$ steps per ball.
Choosing the Optimal Radius

Should we take many small balls or a few large ones?

- Suppose we need $O(a')$ steps per ball
- then $r = n/(a + 1)$ is optimal
Choosing the Optimal Radius

Should we take many small balls or a few large ones?

- Suppose we need $O(a^r)$ steps per ball
- then $r = n/(a + 1)$ is optimal
- $O\left(\left(2 - \frac{2}{a+1}\right)^n\right)$ steps for all balls
Choosing the Optimal Radius

Should we take many small balls or a few large ones?

Suppose we need $O(a^r)$ steps per ball

then $r = n/(a + 1)$ is optimal

$O \left( \left(2 - \frac{2}{a+1}\right)^n \right)$ steps for all balls

<table>
<thead>
<tr>
<th>$a^r$</th>
<th>$\left(2 - \frac{2}{a+1}\right)^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3^r$</td>
<td>$1.5^n$</td>
</tr>
<tr>
<td>$2.848^r$</td>
<td>$1.481^n$</td>
</tr>
<tr>
<td>$2.792^r$</td>
<td>$1.473^n$</td>
</tr>
<tr>
<td>$2.74^r$</td>
<td>$1.465^n$</td>
</tr>
</tbody>
</table>

Dominik Scheder
A Faster Deterministic Algorithm For 3-SAT
2. The searchball Algorithm
Definition of “Setting a variable”

\[ F = \{ \{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\} \} \]
Definition of “Setting a variable”

\[ F = \{\{x, y, z\}, \{x, \bar{y}, z\}, \{x, y, \bar{z}\}, \{\bar{x}, y, z\}\} \]

Set variable \( x \) to 0
Defintion of “Setting a variable”

\[F = \{\{x, y, z\}, \{x, \overline{y}, z\}, \{x, y, \overline{z}\}, \{\overline{x}, y, z\}\}\]

Set variable \(x\) to 0

\[\{\{x, y, z\}, \{x, \overline{y}, z\}, \{x, y, \overline{z}\}, \{\overline{x}, y, z\}\}\]
Defintion of “Setting a variable”

\[ F = \{ \{ x, y, z \}, \{ x, \bar{y}, z \}, \{ x, y, \bar{z} \}, \{ \bar{x}, y, z \} \} \]

Set variable \( x \) to 0

\[ F^{[x=0]} = \{ \{ y, z \}, \{ \bar{y}, z \}, \{ y, \bar{z} \} \} \]
The Searchball Algorithm

searchball(Formula $F$, depth $r$, assignment $\alpha$)

1: if $\alpha$ satisfies $F$ then
2: return true
3: else if $\Box \in F$ or $r \leq 0$ then
4: return false
5: else
6: pick some unsatisfied clause $\{u_1, \ldots, u_\ell\} \in F$
7: return $\bigvee_{i=1}^{\ell} \text{searchball}(F[u_i \rightarrow 1], r - 1)$
8: end if
searchball\textit{(Formula }F\textit{, depth }r\textit{, assignment }\alpha\textit{)}

1: if \(\alpha\) satisfies \(F\) then
2: return true
3: else if \(\square \in F\) or \(r \leq 0\) then
4: return false
5: else
6: pick some unsatisfied clause \(\{u_1, \ldots, u_\ell\} \in F\)
7: return \(\bigvee_{i=1}^{\ell} \text{searchball}(F[u_i \rightarrow 1], r - 1)\)
8: end if

From now on, \(\alpha = (1, \ldots, 1)\)
Visualization of Searchball

Radius: \( r = 2 \)
Center of ball: \( \alpha = (1, 1, 1, 1) \)

\[
\{ \{ \bar{x}, \bar{y}, \bar{z} \}, \{ \bar{y}, \bar{z}, \bar{w} \}, \{ x, y, z \}, \{ x, \bar{y}, z \}, \{ x, \bar{y}, w \}, \{ y, \bar{z}, \bar{w} \}, \{ \bar{x}, y, z \} \}
\]
Visualization of Searchball

Radius: $r = 2$

Center of ball: $\alpha = (1, 1, 1, 1)$

\[
\{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, y, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}
\]
Visualization of Searchball

Radius: $r = 2$
Center of ball: $\alpha = (1, 1, 1, 1)$

$\{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, w\}, \{\bar{x}, y, z\}\}$

$x = 0$

Dominik Scheder
A Faster Deterministic Algorithm For 3-SAT
Visualization of Searchball

Radius: \( r = 2 \)
Center of ball: \( \alpha = (1, 1, 1, 1) \)

\[ \{\{\bar{x}, y, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\} \]
Visualization of Searchball

Radius: \( r = 2 \)  
Center of ball: \( \alpha = (1, 1, 1, 1) \)

\( \{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\} \)

\( x = 0 \)  
\( y = 0 \)
Visualization of Searchball

Radius: \( r = 2 \)

Center of ball: \( \alpha = (1, 1, 1, 1) \)

\( \{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, z\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\} \)
Visualization of Searchball

Radius: $r = 2$
Center of ball: $\alpha = (1, 1, 1, 1)$

$$\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, w\}, \{\bar{x}, y, z\}$$
Visualization of Searchball

Radius: $r = 2$

Center of ball: $\alpha = (1, 1, 1, 1)$

\[
\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, w\}, \{\bar{x}, y, z\}\]

$x = 0$

$y = 0$

$z = 0$

Dominik Scheder

A Faster Deterministic Algorithm For 3-SAT
Visualization of Searchball

Radius: \( r = 2 \)

Center of ball: \( \alpha = (1, 1, 1, 1) \)

\{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, z\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}

Dominik Scheder
A Faster Deterministic Algorithm For 3-SAT
Visualization of Searchball

Radius: $r = 2$

Center of ball: $\alpha = (1, 1, 1, 1)$

$\{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}$
Visualization of Searchball

Radius: \( r = 2 \)  
Center of ball: \( \alpha = (1, 1, 1, 1) \)  
\[ \{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, z\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\]
Visualization of Searchball

Radius: \( r = 2 \)
Center of ball: \( \alpha = (1, 1, 1, 1) \)

\[
\{\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}
\]
Visualization of Searchball

Radius: $r = 2$
Center of ball: $\alpha = (1, 1, 1, 1)$

$\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}$

Dominik Scheder
A Faster Deterministic Algorithm For 3-SAT
Visualization of Searchball

Radius: \( r = 2 \)

Center of ball: \( \alpha = (1, 1, 1, 1) \)

\[
\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, z\}, \{x, \bar{y}, z\}, \{x, \bar{y}, \bar{w}\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}\}
\]
Visualization of Searchball

Radius: $r = 2$
Center of ball: $\alpha = (1, 1, 1, 1)$

$$\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, \bar{w}\}, \{\bar{x}, y, z\}$$
Visualization of Searchball

Radius: $r = 2$
Center of ball: $\alpha = (1, 1, 1, 1)$

$$\{\bar{x}, \bar{y}, \bar{z}\}, \{\bar{y}, \bar{z}, \bar{w}\}, \{x, y, \bar{z}\}, \{x, \bar{y}, z\}, \{x, \bar{y}, w\}, \{y, \bar{z}, w\}, \{\bar{x}, y, z\}\}$$

Satisfying Assignment: $x = 1, y = 0, z = 1, w = 0$
How many leaves does this search tree have?

- For a 3-CNF formula, at most $3^r$ leaves
- But: Not every node has 3 children!
- Try to generate small unsatisfied clauses
- Four quite simple rules
3. Improving \textit{searchball}
Rule 1

If \( \{\bar{u}, \bar{v}\} \in F \) (or \( \{\bar{u}\} \in F \), or \( \square \in F \):

\[
\{\bar{u}, \bar{v}\}
\]

\[ u = 0 \quad \text{and} \quad v = 0 \]

\[
L(r) \leq 2L(r - 1) \quad \implies \quad L(r) \leq 2^r
\]
Rule 2

If \( \{u\}, \{\bar{u}, \bar{v}, \bar{w}\} \in F \):

\[
\begin{align*}
L(r) &\leq 2L(r - 1) \quad \implies \quad L(r) \leq 2^r
\end{align*}
\]
Rule 3

If \( \{u, \bar{a}\}, \{\bar{u}, \bar{v}, \bar{w}\} \in F: \)

\[
\begin{align*}
L(r) & \leq 2L(r - 1) + L(r - 2) \\
\implies L(r) & \leq 2.414^r
\end{align*}
\]
Rule 4

If \( \{u, x\}, \{\bar{u}, \bar{v}, \bar{w}\}, \{\bar{x}, \bar{y}, \bar{z}\} \in F:\)

\[
L(r) \leq 2L(r - 1) + 2L(r - 2) \implies L(r) \leq 2.73^r
\]
Otherwise: $F$ is a reduced formula

Run a “preliminary search”: 
Otherwise: $F$ is a \textit{reduced} formula

Run a “preliminary search”:

\begin{itemize}
  \item Path $\equiv$ set of variables.
\end{itemize}
Otherwise: $F$ is a reduced formula

Run a “preliminary search”:

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
Otherwise: $F$ is a \textit{reduced} formula

Run a “preliminary search”: 

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
3. Abort search at reduced formulas.
Otherwise: $F$ is a *reduced* formula

Run a “preliminary search”:

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
3. Abort search at reduced formulas.
Otherwise: $F$ is a *reduced* formula

Run a “preliminary search”:

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
3. Abort search at reduced formulas.
Otherwise: $F$ is a *reduced* formula

Run a “preliminary search”:

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
3. Abort search at reduced formulas.
Otherwise: $F$ is a reduced formula

Run a “preliminary search”:

1. Path $\equiv$ set of variables.
2. Only allow certain “nice” sets.
3. Abort search at reduced formulas.
Repeat Preliminary Search:

\[ F \]

reduced
Repeat Preliminary Search:
Repeat Preliminary Search:

\[ F \]

not exact

\[ r = 0 \]

reduced
Repeat Preliminary Search:

$F$ (reduced)

$r = 0$

not exact

Dominik Scheder
A Faster Deterministic Algorithm For 3-SAT
Dealing With Reduced Formulas

Repeat Preliminary Search:

$r = 0$

not exact

reduced

$F$

$F[γ]$
Dealing With Reduced Formulas

Repeat Preliminary Search:

\[ F \]

not exact

\[ r = 0 \]

reduced

\[ F[\gamma] \]
Focus on $F[\gamma]$
Focus on $F[\gamma]$

- No reduced formulas in this tree.
Focus on $F[\gamma]$

- No reduced formulas in this tree.
- Use the same tree for $F$!
No reduced formulas in this tree.

Use the same tree for $F$!

Use $F[\gamma]$ as a “guide” for $F$. 
Rule 1–4 have good running time.
We use a poor branching on reduced formulas.
If we branch as the “guide” tells us, we will not encounter too many reduced formulas.

Rule 4 dominates the running time:
\[ L(r) \leq 2L(r - 1) + 2L(r - 2). \]
\[ L(r) \in \mathcal{O}(2.74^r) \]
What did we achieve?

- Search $B_r(1, \ldots, 1)$ in $O(2.74^r)$ steps.
- Solve 3-SAT in $O(1.465^n)$ steps.

SO WHAT???

- Find a simpler way to describe the algorithm (“What is really going on?”)
- Apply the idea to other problems?
Muito obrigado pela sua atenção!!!