On Market-Inspired Approaches to Propositional Satisfiability

William E. Walsh
Katsutoshi Hirayama
Presenter: Gleb Skobeltsyn 30.01.2003
Contents

- Propositional satisfiability
- MarketSAT protocols
- Uniform Pricing protocol (MS-U)
- Differential Pricing protocol (MS-D)
- Economic interpretation
- Completeness
- Distributed Breakout Algorithm (DBA)
- Experiments
- Conclusions
Propositional satisfiability (1)

Conjunctive Normal Form (CNF):

\[ f = (u_{i_1} \lor \ldots \lor u_{j_1} \lor \overline{u}_{j_{i+1}} \lor \ldots \lor \overline{u}_{n_1}) \land \ldots \land (u_{i_m} \lor \ldots \lor \overline{u}_{n_m}) \]

Task: Find an assignment to the variables of a boolean function \( f \) such that \( f \) evaluates to true (T).

Example: \( f = q_1 \land q_2 \land q_3 \)

\[ q_1 = x \lor y \lor z; \quad q_2 = x \lor \neg y \lor z; \quad q_3 = \neg x \lor \neg y \lor z \]

\[
\begin{align*}
    f &= (x \lor y \lor z) \land (x \lor \neg y \lor z) \land (\neg x \lor \neg y \lor z) \\
    &\quad \text{true \quad true \quad true} \\
    &\quad \text{F \quad y=F, z=F} \\
    f &= (x \lor y \lor z) \land (x \lor \neg y \lor z) \land (\neg x \lor \neg y \lor z) \\
    &\quad \text{true \quad true \quad true} \\
    &\quad \text{T \quad y=F, z=F}
\end{align*}
\]
Propositional satisfiability (2)

Given:
Set of variables $U=\{u_1..u_n\}$. Set of clauses $Q=\{q_1..q_m\}$.
The problem to determine whether exists a truth assignment $t: U\to \{\text{True,False}\}$ that satisfies each $q \in Q$

A variable $u$ fails to satisfy a clause $q$ under truth assignment $t$ iff either:
(1) $t(u)=T$, $u \notin q$ and $\neg u \notin q$, or
(2) $t(u)=F$, $\neg u \notin q$ and $u \in q$

Assume that clause consists of $|q|$ variables.
Clause $q$ is satisfied under $t$ iff at most $|q|-1$ variables fail to satisfy the clause.

Each clause makes available $|q|-1$ licenses to fail to satisfy the clause
MarketSAT protocols

Agents
- For each variable $u$ we have a variable agent $a_u$ that determines the assignment $u$.
- Agents select tentative truth assignments and submit bids to a subset of the auctions corresponding to their chosen truth assignments.

Auctions
- Auctions correspond to clauses.
- Auctions send price quote messages to the agents indicating the current going prices of the licenses.
MarketSAT example (1)

Auction $q_1$
\( x \land y \land z \)

Agent $x$, false

Auction $q_2$
\( x \land \neg y \land z \)

Agent $y$, false

Auction $q_3$
\( \neg x \land \neg y \land z \)

Agent $z$, false
MarketSAT example (2)

Auction \( q_1 \)
\[
\begin{array}{c}
\text{x y z} \\
\text{license1 license2}
\end{array}
\]

Auction \( q_2 \)
\[
\begin{array}{c}
\text{x \neg y z} \\
\text{license1 license2}
\end{array}
\]

Auction \( q_3 \)
\[
\begin{array}{c}
\neg x \neg y z \\
\text{license1 license2}
\end{array}
\]

Agent x, true

Agent y, false

Agent z, false

On Market-Inspired Approaches to Propositional Satisfiability
Presenter: Gleb Skobeltsyn. 30/01/2003
MS-U. Auction rules

Auction $q_1$ $x \ y \ z$

1$ ask price $\alpha$

2$ bid price $\beta$

Available licenses

Auction $q_2$

1$ ask price $\alpha$

Bid price $\beta = 0$

Auction $q_3$

Ask price $\alpha = 0$

Bid price $\beta = 0$

Winner winner

LOOSER!

Agent $x$, false

Agent $y$, false

Agent $z$, false
Choose an assignment that minimizes agent’s assignment cost $A_i$. Assume $p$ is perceived cost.

$$A_i = \max(\text{sum of } p \text{ needed for assignment, } A_{i-1})$$

- $p=\beta$ if agent is winning
- $p=\alpha$ if agent hasn’t submitted a bid
- $p=\alpha$ if agent is loosing and $\alpha > \beta$
- $p=\alpha + \delta$ if agent is loosing and $\alpha = \beta$

**MS-U. Bidding policies**

Initial assignment

Send bids

Wait for price quote message

Choose an assignment

Increase any loosing bid by $\delta$ if needed

No bids?

- yes: finish
- no: choose an assignment that minimizes agent’s assignment cost $A_i$. Assume $p$ is perceived cost.

$$A_i = \max(\text{sum of } p \text{ needed for assignment, } A_{i-1})$$

- $p=\beta$ if agent is winning
- $p=\alpha$ if agent hasn’t submitted a bid
- $p=\alpha$ if agent is loosing and $\alpha > \beta$
- $p=\alpha + \delta$ if agent is loosing and $\alpha = \beta$
MS-D. Auction rules

Auction \( q_1 \)
\[ x \land y \land z \]
1 from \( x \)
1 from \( y \)
1 from \( z \)

Premium price = \( p_1 + 1 \)

Available licenses

Agent \( x \), false

Auction \( q_2 \)
\[ x \land \neg y \land z \]
1 from \( z \)
1 from \( x \)

Premium price = \( p_2 \)

Agent \( y \), false

Auction \( q_3 \)
\[ \neg x \land \neg y \land z \]
1 from \( z \)

Premium price = \( p_3 \)

Agent \( z \), false

On Market-Inspired Approaches to Propositional Satisfiability
Presenter: Gleb Skobeltsyn. 30/01/2003
MS-D. Bidding policies

Place bids of quantity:
- 1 for all licenses it needs for the assignment,
- 0 for all licenses for the opposite assignment.

Choose an assignment that minimizes the sum of the prices of licenses as reported by last price quote.

Resubmit any bid needed for its assignment and for which it received a premium price quote.

If it received a 0 price quote for all its bids for the assignment.

Initial assignment

Send bids

Wait for price quote message

Choose an assignment

Flip?

yes

no

Resubmit bids

No bids?

yes

no

finish
Economic interpretation

- For an agent $a$ to be willing to participate in these protocols, it must obtain some value $v_a$ from participating in a satisfying solution.
- An agent wishes to maximize its **surplus** value.

- In **MS-U** a rational agent would stop bidding when the total cost of assignment exceeds $v_a$.
- In **MS-D** agents pay nothing for a satisfying solution. But if there is some reasonable expectation that the protocol could stop at any time, the bidding policies become more rational.
Completeness

- **MS-U is incomplete**

  \[ f_u = (\neg x \lor y) \land (\neg x \lor z) \land (\neg x \lor w) \land (\neg y \lor x) \land (\neg y \lor z) \land (\neg y \lor w) \land (\neg z \lor x) \land (\neg z \lor y) \land (\neg z \lor w) \land (\neg w \lor x) \land (\neg w \lor y) \land (\neg w \lor z) \]

  The only solutions are: \( x = y = z = w = T \) or \( x = y = z = w = F \)

  Initial values: \( x = z = T, \ y = w = F \)

  1\textsuperscript{st} round, assume that:
  \[
  x \text{ wins } (\neg x \lor y), \ y \text{ wins } (\neg z \lor y), \ z \text{ wins } (\neg z \lor w), \ w \text{ wins } (\neg x \lor w). \]

  2\textsuperscript{nd} round, assume that:
  \[
  x \text{ wins } (\neg w \lor x), \ y \text{ wins } (\neg y \lor x), \ z \text{ wins } (\neg y \lor z), \ w \text{ wins } (\neg w \lor z). \]

- **MS-D is incomplete**

  \[ f_d = (x \lor y \lor z \lor w) \land (\neg x \lor y) \land (\neg x \lor z) \land (\neg x \lor w) \land (\neg y \lor x) \land (\neg y \lor z) \land (\neg y \lor w) \land (\neg z \lor x) \land (\neg z \lor y) \land (\neg z \lor w) \land (\neg w \lor x) \land (\neg w \lor y) \land (\neg w \lor z) \]

  Solution: \( x = y = z = w = T \). Initial values: \( x = T, \ y = z = w = F \)
**Distributed Breakout (DBA)**

- **Nogood** is correspond to failing to satisfy a clause
- **Weights of nogoods** are analogous to prices

---

Initialize the weight of all nogoods to 1

- **current state = solution?**
  - yes: finish
  - no: make any local change that reduces the total weights of violated nogoods

- **current state = local min?**
  - yes: increase weights of all currently violated nogoods
  - no: make any local change that reduces the total weights of violated nogoods
## Experiments

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Success ratio</th>
<th>Average rounds</th>
<th>Median rounds</th>
<th>σ Rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MS-O</strong></td>
<td>50</td>
<td>0.40</td>
<td>3.90×10^4</td>
<td>5.00×10^4</td>
<td>1.81×10^4</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.75</td>
<td>2.75×10^4</td>
<td>1.08×10^4</td>
<td>3.03×10^4</td>
</tr>
<tr>
<td><strong>MS-U (Uniform Pricing)</strong></td>
<td>100</td>
<td>0.53</td>
<td>6.10×10^4</td>
<td>8.21×10^4</td>
<td>4.16×10^4</td>
</tr>
<tr>
<td><strong>MS-D (Differential Pricing)</strong></td>
<td>50</td>
<td>1.00</td>
<td>896</td>
<td>250</td>
<td>3.52×10^3</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.98</td>
<td>3.98×10^3</td>
<td>429</td>
<td>1.17×10^4</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.96</td>
<td>1.04×10^4</td>
<td>1.50×10^3</td>
<td>2.34×10^4</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>0.85</td>
<td>2.74×10^4</td>
<td>3.65×10^3</td>
<td>4.45×10^4</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.85</td>
<td>3.69×10^4</td>
<td>5.94×10^3</td>
<td>5.45×10^4</td>
</tr>
<tr>
<td><strong>DB (Distributed Breakout)</strong></td>
<td>50</td>
<td>1.00</td>
<td>234</td>
<td>64</td>
<td>829</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>0.99</td>
<td>2.14×10^3</td>
<td>299</td>
<td>9.40×10^3</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.98</td>
<td>4.26×10^3</td>
<td>460</td>
<td>1.61×10^3</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>0.96</td>
<td>9.12×10^3</td>
<td>1.42×10^3</td>
<td>2.63×10^4</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>0.93</td>
<td>1.80×10^4</td>
<td>1.22×10^3</td>
<td>4.24×10^4</td>
</tr>
</tbody>
</table>
Conclusions

- We considered two market-inspired protocols for propositional satisfiability MS-U and MS-D.
- There are tradeoffs in terms of performance, decentralization, and the plausibility of assumed agent behaviors.
- Pricing method can significantly affect the performance.
- The MarketSAT protocols are highly decentralized.
Last slide

Questions?