Quadratization of Pseudo-Boolean Functions

Endre Boros

RUTCOR, Rutgers University

University of Primorska, November 19, 2012

Joint work with A. Fix, A. Gruber, G. Tavares and R. Zabih
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
     - Representations and Bounds
     - Origin of Graph Cut Models
     - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Quadratic Unconstrained Binary Optimization (QUBO)

**Variables and Literals**
- **Variables**: \( x_1, x_2, \ldots, x_n \in \{0, 1\} \).
- **Negations**: \( \bar{x}_i = 1 - x_i \in \{0, 1\} \) for \( i = 1, \ldots, n \).
- **Literals**: \( x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n \).

**Quadratic Pseudo-Boolean Function (QPBF):**
\[
f : \{0, 1\}^n \rightarrow \mathbb{R}
\]
\[
f(x_1, \ldots, x_n) = c_0 + \sum_{j=1}^{n} c_j x_j + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j
\]

**Quadratic Unconstrained Binary Optimization (QUBO)**
\[
\min_{(x_1, \ldots, x_n) \in \{0, 1\}^n} f(x_1, \ldots, x_n)
\]
## Quadratic Unconstrained Binary Optimization (QUBO)

### Variables and Literals

- **Variables**: \(x_1, x_2, \ldots, x_n \in \{0, 1\}\).
- **Negations**: \(\overline{x}_i = 1 - x_i \in \{0, 1\}\) for \(i = 1, \ldots, n\).
- **Literals**: \(x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\)

### Quadratic Pseudo-Boolean Function (QPBF):

\[
f(x_1, \ldots, x_n) = c_0 + \sum_{j=1}^{n} c_j x_j + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j
\]

### Quadratic Unconstrained Binary Optimization (QUBO)

\[
\min_{(x_1, \ldots, x_n) \in \{0, 1\}^n} f(x_1, \ldots, x_n)
\]
# Quadratic Unconstrained Binary Optimization (QUBO)

## Variables and Literals
- **Variables**: $x_1, x_2, \ldots, x_n \in \{0, 1\}$.
- **Negations**: $\overline{x}_i = 1 - x_i \in \{0, 1\}$ for $i = 1, \ldots, n$.
- **Literals**: $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$

## Quadratic Pseudo-Boolean Function (QPBF):
\[
f : \{0, 1\}^n \rightarrow \mathbb{R}
\]
\[
f(x_1, \ldots, x_n) = c_0 + \sum_{j=1}^{n} c_j x_j + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j
\]

## Quadratic Unconstrained Binary Optimization (QUBO)
\[
\min_{(x_1, \ldots, x_n) \in \{0, 1\}^n} f(x_1, \ldots, x_n)
\]
Variables and Literals

- **Variables**: $x_1, x_2, \ldots, x_n \in \{0, 1\}$.
- **Negations**: $\overline{x}_i = 1 - x_i \in \{0, 1\}$ for $i = 1, \ldots, n$.
- **Literals**: $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$.

Quadratic Pseudo-Boolean Function (QPBF):

$$f(x_1, \ldots, x_n) = c_0 + \sum_{j=1}^{n} c_j x_j + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j$$

Quadratic Unconstrained Binary Optimization (QUBO)

$$\min_{(x_1, \ldots, x_n) \in \{0, 1\}^n} f(x_1, \ldots, x_n)$$
**Quadratic Unconstrained Binary Optimization (QUBO)**

### Variables and Literals
- **Variables**: \( x_1, x_2, \ldots, x_n \in \{0, 1\} \).
- **Negations**: \( \overline{x}_i = 1 - x_i \in \{0, 1\} \) for \( i = 1, \ldots, n \).
- **Literals**: \( x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \)

### Quadratic Pseudo-Boolean Function (QPBF):

\[
f : \{0, 1\}^n \rightarrow \mathbb{R}
\]

\[
f(x_1, \ldots, x_n) = c_0 + \sum_{j=1}^{n} c_j x_j + \sum_{1 \leq i < j \leq n} c_{ij} x_i x_j
\]

### Quadratic Unconstrained Binary Optimization (QUBO)

\[
\min_{(x_1, \ldots, x_n) \in \{0, 1\}^n} f(x_1, \ldots, x_n)
\]
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
     - Origin of Graph Cut Models
     - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
**Posiforms**: Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}$
Representations and Bounds

**Posiforms**: Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$

$$f = -2 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 \quad \text{QPBF}$$
Representations and Bounds

**Posiforms**: Nonnegative (except maybe the constant terms) multi-linear polynomials in \(2n\) literals \(x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\)

\[
\begin{align*}
  f & = -2 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 \quad \text{QPBF} \\
  & = -5 + \overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_1 x_2 + x_1 x_3 + x_2 x_3 \quad \text{quadratic posiform}
\end{align*}
\]
# Representations and Bounds

**Posiforms:** Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}$

<table>
<thead>
<tr>
<th>$f$</th>
<th>$-2 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3$</th>
<th>QPBF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$-5 + \overline{x_1} + \overline{x_2} + \overline{x_3} + x_1x_2 + x_1x_3 + x_2x_3$</td>
<td>quadratic posiform</td>
</tr>
<tr>
<td></td>
<td>$-4 + \overline{x_3} + \overline{x_1}\overline{x_2} + x_1x_3 + x_2x_3$</td>
<td>quadratic posiform</td>
</tr>
</tbody>
</table>
### Representations and Bounds

**Posiforms:** Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$

<table>
<thead>
<tr>
<th>$f$</th>
<th>Quadratic posiform</th>
<th>Cubic posiform</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-2 - x_1 - x_2 - x_3 + x_1 x_2 + x_1 x_3 + x_2 x_3$</td>
<td>QPBF</td>
<td></td>
</tr>
<tr>
<td>$-5 + \overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_1 x_2 + x_1 x_3 + x_2 x_3$</td>
<td>quadratic posiform</td>
<td></td>
</tr>
<tr>
<td>$-4 + \overline{x}_3 + \overline{x}_1 \overline{x}_2 + x_1 x_3 + x_2 x_3$</td>
<td>quadratic posiform</td>
<td></td>
</tr>
<tr>
<td>$-3 + x_1 x_2 x_3 + \overline{x}_1 \overline{x}_2 \overline{x}_3$</td>
<td>cubic posiform</td>
<td></td>
</tr>
</tbody>
</table>
### Representations and Bounds

**Posiforms:** Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x}_1, ..., x_n, \overline{x}_n$

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f = -2 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3$</td>
<td>QPBF</td>
</tr>
<tr>
<td>$= -5 + \overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_1x_2 + x_1x_3 + x_2x_3$</td>
<td>quadratic posiform</td>
</tr>
<tr>
<td>$= -4 + \overline{x}_3 + \overline{x}_1\overline{x}_2 + x_1x_3 + x_2x_3$</td>
<td>quadratic posiform</td>
</tr>
<tr>
<td>$= -3 + x_1x_2x_3 + \overline{x}_1\overline{x}_2\overline{x}_3$</td>
<td>cubic posiform</td>
</tr>
</tbody>
</table>

**Roof Dual Bound:** $C_2(f) \leq f$  

-Hammer, Hansen and Simeone, 1984-

$C_2(f)$ = largest $C$ s.t. $f = C + \phi$ for some **quadratic posiform** $\phi$. 

---

**QUBO**

**Polynomial Time Preprocessing**

**What is Quadratization?**

**Quadratization Techniques**
Representations and Bounds

**Posiforms**: Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n$

$$f = -2 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3$$  
**QPBF**

$$f = -5 + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + x_1x_2 + x_1x_3 + x_2x_3$$  
quadratic posiform

$$f = -4 + x_3 + \bar{x}_1\bar{x}_2 + x_1x_3 + x_2x_3$$  
quadratic posiform

$$f = -3 + x_1x_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3$$  
cubic posiform

**Roof Dual Bound**: $C_2(f) \leq f$  
(Hammer, Hansen and Simeone, 1984)

$C_2(f) = \text{largest } C \text{ s.t. } f = C + \phi$ for some **quadratic posiform** $\phi$.

**Cubic Dual Bound**: $C_3(f) \leq f$  
(Boros, Crama and Hammer, 1992)

$C_3(f) = \text{largest } C \text{ s.t. } f = C + \psi$ for some **cubic posiform** $\psi$. 
Representations and Bounds

**Posiforms**: Nonnegative (except maybe the constant terms) multi-linear polynomials in $2n$ literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$

$$f = -2 - x_1 - x_2 - x_3 + x_1x_2 + x_1x_3 + x_2x_3 \quad \text{QPBF}$$

$$= -5 + \overline{x}_1 + \overline{x}_2 + \overline{x}_3 + x_1x_2 + x_1x_3 + x_2x_3 \quad \text{quadratic posiform}$$

$$= -4 + \overline{x}_3 + \overline{x}_1\overline{x}_2 + x_1x_3 + x_2x_3 \quad \text{quadratic posiform}$$

$$= -3 + x_1x_2x_3 + \overline{x}_1\overline{x}_2\overline{x}_3 \quad \text{cubic posiform}$$

**Roof Dual Bound**: $C_2(f) \leq f$  
( Hammer, Hansen and Simeone, 1984)

$C_2(f) =$ largest $C$ s.t. $f = C + \phi$ for some quadratic posiform $\phi$.

**Cubic Dual Bound**: $C_3(f) \leq f$  
( Boros, Crama and Hammer, 1992)

$C_3(f) =$ largest $C$ s.t. $f = C + \psi$ for some cubic posiform $\psi$.

$$C_2(f) \leq C_3(f) \leq \cdots \leq C_n(f) = \min f$$
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
A QPBF is submodular IFF all quadratic coefficients are nonpositive. \textit{(Doit Yourself, anytime)}

To a submodular QPBF $f$ associate a network $G_f$ as follows

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$. \textit{(Hammer, 1965)}
A QPBF is submodular IFF all quadratic coefficients are nonpositive. *(Do it Yourself, anytime)*

To a submodular QPBF $f$ associate a network $G_f$ as follows

$$f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3$$

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$. *(Hammer, 1965)*
Network Model for Submodular QUBO (Hammer, 1965)

- A QPBF is submodular IFF all quadratic coefficients are nonpositive. \(\text{(Do it Yourself, anytime)}\)
- To a submodular QPBF \(f\) associate a network \(G_f\) as follows
  \[
  f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3
  \]
  \[
  = 4\overline{x}_1 + 5\overline{x}_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
  \]
- There is a one-to-one correspondence between values of \(f\) and \(s - t\) cut values of \(G_f\). \(\text{(Hammer, 1965)}\)
Network Model for Submodular QUBO (Hammer, 1965)

- A QPBF is submodular IFF all quadratic coefficients are nonpositive. 
  \[\text{(Do it Yourself, anytime)}\]
- To a submodular QPBF \( f \) associate a network \( G_f \) as follows
  \[
  f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3
  \]
  \[
  = 4\overline{x}_1 + 5x_2 + 3\overline{x}_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
  \]
  \[
  = 4s\overline{x}_1 + 5x_2\overline{t} + 3\overline{x}_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
  \]
- There is a one-to-one correspondence between values of \( f \) and \( s - t \) cut values of \( G_f \).
  \[\text{(Hammer, 1965)}\]
A QPBF is submodular IFF all quadratic coefficients are nonpositive. \(\text{(Doit Yourself, anytime)}\)

To a submodular QPBF \(f\) associate a network \(G_f\) as follows:

\[
f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3
\]
\[
= 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
\]
\[
= 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
\]

There is a one-to-one correspondence between values of \(f\) and \(s - t\) cut values of \(G_f\). \(\text{(Hammer, 1965)}\)
A QPBF is submodular IFF all quadratic coefficients are nonpositive.

(Do it Yourself, anytime)

To a submodular QPBF $f$ associate a network $G_f$ as follows:

$$f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3$$

$$= 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

$$= 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$.  

(Hammer, 1965)
Network Model for Submodular QUBO (Hammer, 1965)

- A QPBF is submodular IFF all quadratic coefficients are nonpositive.  
  \[ f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3 \]

- To a submodular QPBF \( f \) associate a network \( G_f \) as follows

\[ f = 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3 \]

\[ = 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3 \]

- There is a one-to-one correspondence between values of \( f \) and \( s - t \) cut values of \( G_f \).  

(Doit Yourself, anytime)
A QPBF is submodular IFF all quadratic coefficients are nonpositive. \(\text{(Do it Yourself, anytime)}\)

To a submodular QPBF \(f\) associate a network \(G_f\) as follows

\[
f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3
\]

\[
= 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
\]

\[
= 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3
\]

There is a one-to-one correspondence between values of \(f\) and \(s-t\) cut values of \(G_f\). \(\text{(Hammer, 1965)}\)
A QPBF is submodular IFF all quadratic coefficients are nonpositive. 

(Do it Yourself, anytime)

To a submodular QPBF $f$ associate a network $G_f$ as follows

$$f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3$$

$$= 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

$$= 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$. 

(Hammer, 1965)
A QPBF is submodular IFF all quadratic coefficients are nonpositive. 

(Do it Yourself, anytime)

To a submodular QPBF $f$ associate a network $G_f$ as follows

$$f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3$$

$$= 4\overline{x}_1 + 5x_2 + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

$$= 4s\overline{x}_1 + 5x_2\overline{t} + 3x_1\overline{x}_2 + \overline{x}_1x_3 + 2x_2\overline{x}_3$$

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$. 

(Hammer, 1965)
A QPBF is submodular IFF all quadratic coefficients are nonpositive.  

To a submodular QPBF $f$ associate a network $G_f$ as follows

$$f = 4 - x_1 + 7x_2 + x_3 - 3x_1x_2 - x_1x_3 - 2x_2x_3$$

$$= 4\bar{x}_1 + 5x_2 + 3x_1\bar{x}_2 + \bar{x}_1x_3 + 2x_2\bar{x}_3$$

$$= 4s\bar{x}_1 + 5x_2t + 3x_1\bar{x}_2 + \bar{x}_1x_3 + 2x_2\bar{x}_3$$

There is a one-to-one correspondence between values of $f$ and $s - t$ cut values of $G_f$.  

$$f(0, 1, 0) = C(\{s, 2\}, \{1, 3, t\}) = 11$$
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2x_1x_0 + 6x_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1) \]
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2\overline{x}_1x_0 + 6\overline{x}_2x_0 + 4\overline{x}_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) \((u \neq v)\) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha / 2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

$$\phi = 2x_0x_0 + 2\bar{x}_1x_0 + 6\bar{x}_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3$$

we associate a directed network $N_\phi$ on vertex set

$$V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1)$$

- Homogenize it by $x_0$.
- Associate to each term $\alpha uv \ (u \neq v)$ two arcs $(u, \overline{v})$ and $(v, \overline{u})$ with capacities $c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2$.
- Associate to $\gamma x_0x_0$ one arc $(x_0, \overline{x}_0)$ with capacity $c(x_0, \overline{x}_0) = \gamma$ and add arc $(\overline{x}_0, x_0)$ with capacity $c(\overline{x}_0, x_0) = +\infty$. 
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2\overline{x}_1x_0 + 6x_2x_0 + 4\overline{x}_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{ x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) (\( u \neq v \)) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform
\[ \phi = 2x_0x_0 + 2\overline{x}_1x_0 + 6\overline{x}_2x_0 + 4\overline{x}_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]
we associate a directed network \( N_\phi \) on vertex set
\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) (\( u \neq v \)) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
To a quadratic posiform

\[ \phi = 2x_0x_0 + 2x_1x_0 + 6x_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) \((u \neq v)\) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2x_1x_0 + 6x_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) \((u \neq v)\) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
To a quadratic posiform

\[ \phi = 2x_0^2 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{ x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) \( (u \neq v) \) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha/2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2\overline{x}_1x_0 + 6x_2\overline{x}_0 + 4\overline{x}_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{ x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \} \quad (x_0 \equiv 1) \]

- Homogenize it by \( x_0 \).
- Associate to each term \( \alpha uv \) (\( u \neq v \)) two arcs \((u, \overline{v})\) and \((v, \overline{u})\) with capacities \( c(u, \overline{v}) = c(v, \overline{u}) = \alpha / 2 \).
- Associate to \( \gamma x_0x_0 \) one arc \((x_0, \overline{x}_0)\) with capacity \( c(x_0, \overline{x}_0) = \gamma \) and add arc \((\overline{x}_0, x_0)\) with capacity \( c(\overline{x}_0, x_0) = +\infty \).
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2x_1x_0 + 6x_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \} \quad (x_0 \equiv 1) \]

\( N_\phi \) is a symmetric network: twin pair of paths, cycles and flows

- If \( u_0, u_1, \ldots, u_k \) is a directed path (cycle) in \( N_\phi \) then so is \( \overline{u}_k, \overline{u}_{k-1}, \ldots, \overline{u}_1, \overline{u}_0 \).
- Every feasible circulation in \( N_\phi \) has its symmetric twin also feasible, and hence their convex combination is a feasible symmetric circulation.

\[
\begin{align*}
    x_1 + x_1x_3 + x_3 &= x_0x_1 + x_1x_3 + x_3x_0 + x_0x_0 \\
    &= x_0x_1 + x_1x_3 + x_3x_0 + x_0x_0 \\
    &= x_1x_3 + 1
\end{align*}
\]
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

$$\phi = 2x_0x_0 + 2\bar{x}_1x_0 + 6x_2x_0 + 4\bar{x}_3x_0 + 8x_1x_2 + 6\bar{x}_1x_3 + 2x_2x_3$$

we associate a directed network $N_\phi$ on vertex set

$$V(N_\phi) = \{x_0, \bar{x}_0, x_1, \bar{x}_1, ..., x_n, \bar{x}_n\} \quad (x_0 \equiv 1)$$

$N_\phi$ is a symmetric network: twin pair of paths, cycles and flows

- If $u_0, u_1, ..., u_k$ is a directed path (cycle) in $N_\phi$ then so is $\bar{u}_k, \bar{u}_{k-1}, ..., \bar{u}_1, \bar{u}_0$.
- Every feasible circulation in $N_\phi$ has its symmetric twin also feasible, and hence their convex combination is a feasible symmetric circulation.

$$\bar{x}_1 + x_1\bar{x}_3 + \bar{x}_3 = x_0\bar{x}_1 + x_1\bar{x}_3 + \bar{x}_3x_0 + \bar{x}_0x_0$$
$$= x_0x_1 + x_1x_3 + x_3x_0 + x_0x_0$$
$$= x_1x_3 + 1$$
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0x_0 + 2x_1x_0 + 6x_2x_0 + 4x_3x_0 + 8x_1x_2 + 6x_1x_3 + 2x_2x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \overline{x}_0, x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\} \quad (x_0 \equiv 1) \]

\( N_\phi \) is a symmetric network: twin pair of paths, cycles and flows

- If \( u_0, u_1, \ldots, u_k \) is a directed path (cycle) in \( N_\phi \) then so is \( \overline{u}_k, \overline{u}_{k-1}, \ldots, \overline{u}_1, \overline{u}_0 \).
- Every feasible circulation in \( N_\phi \) has its symmetric twin also feasible, and hence their convex combination is a feasible symmetric circulation.

\[
\begin{align*}
\overline{x}_1 + \overline{x}_1x_3 + x_3 &= x_0\overline{x}_1 + x_1x_3 + x_3x_0 + \overline{x}_0x_0 \\
&= \overline{x}_0x_1 + \overline{x}_1x_3 + x_3x_0 + \overline{x}_0x_0 \\
&= x_1x_3 + 1
\end{align*}
\]
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

To a quadratic posiform

\[ \phi = 2x_0 x_0 + 2x_1 x_0 + 6x_2 x_0 + 4x_3 x_0 + 8x_1 x_2 + 6x_1 x_3 + 2x_2 x_3 \]

we associate a directed network \( N_\phi \) on vertex set

\[ V(N_\phi) = \{x_0, \bar{x}_0, x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\} \quad (x_0 \equiv 1) \]

\( N_\phi \) is a symmetric network: twin pair of paths, cycles and flows

- If \( u_0, u_1, \ldots, u_k \) is a directed path (cycle) in \( N_\phi \) then so is \( \bar{u}_k, \bar{u}_{k-1}, \ldots, \bar{u}_1, \bar{u}_0 \).

- Every feasible circulation in \( N_\phi \) has its symmetric twin also feasible, and hence their convex combination is a feasible symmetric circulation.

\[
\bar{x}_1 + x_1 x_3 + x_3 = x_0 \bar{x}_1 + x_1 x_3 + x_3 x_0 + x_0 \bar{x}_0 \\
= \bar{x}_0 x_1 + \bar{x}_1 x_3 + x_3 x_0 + x_0 x_0 \\
= x_1 x_3 + 1
\]
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

Claims

- Two quadratic posiforms $\phi$ and $\psi$ represent the same QPBF if and only if $N_\psi$ is the residual network of $N_\phi$ corresponding to a symmetric feasible circulation.

- The roof dual value $C_2(f)$ is the maximum flow value on arc $(\overline{x}_0, x_0)$ in a feasible circulation in $N_\phi$, where $\phi$ is an arbitrary quadratic posiform of $f$.

- If $N_\psi$ is the residual network corresponding to such a maximum circulation, then the strong components of $N_\psi \setminus \{(x_0, \overline{x}_0)\}$ induce a decomposition of $f$, in which each component can be minimized independently of the others to obtain a minimum of $f$.

- Recursive application of roof-duality does not provide further improvements!
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

Claims

- Two quadratic posiforms $\phi$ and $\psi$ represent the same QPBF if and only if $N_\psi$ is the residual network of $N_\phi$ corresponding to a symmetric feasible circulation.

- The roof dual value $C_2(f)$ is the maximum flow value on arc $(\bar{x}_0, x_0)$ in a feasible circulation in $N_\phi$, where $\phi$ is an arbitrary quadratic posiform of $f$.

- If $N_\psi$ is the residual network corresponding to such a maximum circulation, then the strong components of $N_\psi \setminus \{(x_0, \bar{x}_0)\}$ induce a decomposition of $f$, in which each component can be minimized independently of the others to obtain a minimum of $f$.

- Recursive application of roof-duality does not provide further improvements!
**Claims**

- Two quadratic posiforms $\phi$ and $\psi$ represent the same QPBF if and only if $N_\psi$ is the residual network of $N_\phi$ corresponding to a symmetric feasible circulation.

- The roof dual value $C_2(f)$ is the maximum flow value on arc $(\overline{x}_0, x_0)$ in a feasible circulation in $N_\phi$, where $\phi$ is an arbitrary quadratic posiform of $f$.

- If $N_\psi$ is the residual network corresponding to such a maximum circulation, then the strong components of $N_\psi \setminus \{(x_0, \overline{x}_0)\}$ induce a decomposition of $f$, in which each component can be minimized independently of the others to obtain a minimum of $f$.

- Recursive application of roof-duality does not provide further improvements!
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

---

**Claims**

- Two quadratic posiforms \( \phi \) and \( \psi \) represent the same QPBF if and only if \( N_\psi \) is the residual network of \( N_\phi \) corresponding to a symmetric feasible circulation.

- The roof dual value \( C_2(f) \) is the maximum flow value on arc \((\overline{x}_0, x_0)\) in a feasible circulation in \( N_\phi \), where \( \phi \) is an arbitrary quadratic posiform of \( f \).

- If \( N_\psi \) is the residual network corresponding to such a maximum circulation, then the strong components of \( N_\psi \setminus \{(x_0, \overline{x}_0)\} \) induce a decomposition of \( f \), in which each component can be minimized independently of the others to obtain a minimum of \( f \).

- Recursive application of roof-duality does not provide further improvements!

---

Example Claims:

- Examples of claims about quadratic posiforms and their representations in QPBF.

---

**Implication Networks**

- Graphical representation of implication networks with nodes labeled from 0 to 3, and arcs with weights.

---

**Additional Information**

- A detailed explanation of the implications and applications of implication networks in QUBO problems.

Implication Networks (Boros, Hammer, Sun, 1989, 1992)

**Claims**

- Two quadratic posiforms $\phi$ and $\psi$ represent the same QPBF if and only if $N_\psi$ is the residual network of $N_\phi$ corresponding to a symmetric feasible circulation.

- The roof dual value $C_2(f)$ is the maximum flow value on arc $(\overline{x}_0, x_0)$ in a feasible circulation in $N_\phi$, where $\phi$ is an arbitrary quadratic posiform of $f$.

- If $N_\psi$ is the residual network corresponding to such a maximum circulation, then the strong components of $N_\psi \setminus \{(x_0, \overline{x}_0)\}$ induce a decomposition of $f$, in which each component can be minimized independently of the others to obtain a minimum of $f$.

  cf. persistency (Hammer, Hansen and Simeone, 1984)

  cf. decomposition (Billionet and Sutter, 1992)

- Recursive application of roof-duality does not provide further improvements!
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

Clubs
- Two quadratic posiforms \( \phi \) and \( \psi \) represent the same QPBF if and only if \( N_\psi \) is the residual network of \( N_\phi \) corresponding to a symmetric feasible circulation.
- The roof dual value \( C_2(f) \) is the maximum flow value on arc \((\overline{x}_0, x_0)\) in a feasible circulation in \( N_\phi \), where \( \phi \) is an arbitrary quadratic posiform of \( f \).
- If \( N_\psi \) is the residual network corresponding to such a maximum circulation, then the strong components of \( N_\psi \setminus \{(x_0, \overline{x}_0)\} \) induce a decomposition of \( f \), in which each component can be minimized independently of the others to obtain a minimum of \( f \).

  cf. persistency (Hammer, Hansen and Simeone, 1984)

  cf. decomposition (Billionet and Sutter, 1992)

- Recursive application of roof-duality does not provide further improvements!
Implication Networks (Boros, Hammer, Sun, 1989, 1992)

Claims

- Two quadratic posiforms $\phi$ and $\psi$ represent the same QPBF if and only if $N_\psi$ is the residual network of $N_\phi$ corresponding to a symmetric feasible circulation.

- The roof dual value $C_2(f)$ is the maximum flow value on arc $(\overline{x}_0, x_0)$ in a feasible circulation in $N_\phi$, where $\phi$ is an arbitrary quadratic posiform of $f$.

- If $N_\psi$ is the residual network corresponding to such a maximum circulation, then the strong components of $N_\psi \setminus \{(x_0, \overline{x}_0)\}$ induce a decomposition of $f$, in which each component can be minimized independently of the others to obtain a minimum of $f$.

  cf. persistency (Hammer, Hansen and Simeone, 1984)
  cf. decomposition (Billionet and Sutter, 1992)

- Recursive application of roof-duality does not provide further improvements!
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- Build implication network
- Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
- Probe remaining variables and repeat all of the above as long as there is some change.
- Output remaining strong components, if any.
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- **Build implication network**
  - Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
  - Probe remaining variables and repeat all of the above as long as there is some change.
  - Output remaining strong components, if any.
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- Build implication network
- Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
- Probe remaining variables and repeat all of the above as long as there is some change.
- Output remaining strong components, if any.
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- Build implication network
- Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
- Probe remaining variables and repeat all of the above as long as there is some change.
- Output remaining strong components, if any.
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- Build implication network
- Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
- Probe remaining variables and repeat all of the above as long as there is some change.
- Output remaining strong components, if any.
Components of the Algorithm

The **purpose** of the preprocessing algorithm is to **fix** some of the variables at their optimum values and **decompose** the remaining problem into several smaller problems which do not share variables.

- Build implication network
- Compute maximum flow; **fix variables by persistency** (increase capacities of some arcs)
- Probe remaining variables and repeat all of the above as long as there is some change.
- Output remaining strong components, if any.

If the input QPBF is submodular, then the above procedure will fix all the variables at their optimal values in the first round, without any probing.
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
### Via Minimization in VLSI Design

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n$</th>
<th>Roof Duality (strong)</th>
<th>Roof Duality (weak)</th>
<th>Probing (forcing)</th>
<th>Probing (equalities)</th>
<th>ALL TOOLS</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>via.c1y</td>
<td>829</td>
<td>93.6%</td>
<td>6.4%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0.03</td>
</tr>
<tr>
<td>via.c2y</td>
<td>981</td>
<td>94.7%</td>
<td>5.3%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0.06</td>
</tr>
<tr>
<td>via.c3y</td>
<td>1328</td>
<td>94.6%</td>
<td>5.4%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0.09</td>
</tr>
<tr>
<td>via.c4y</td>
<td>1367</td>
<td>96.4%</td>
<td>3.6%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0.09</td>
</tr>
<tr>
<td>via.c5y</td>
<td>1203</td>
<td>93.1%</td>
<td>6.9%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
<td>0.08</td>
</tr>
<tr>
<td>via.c1n</td>
<td>828</td>
<td>57.4%</td>
<td>9.6%</td>
<td>32.4%</td>
<td>0.6%</td>
<td>100%</td>
<td>0.49</td>
</tr>
<tr>
<td>via.c2n</td>
<td>980</td>
<td>12.4%</td>
<td>4.4%</td>
<td>83.1%</td>
<td>0.1%</td>
<td>100%</td>
<td>7.14</td>
</tr>
<tr>
<td>via.c3n</td>
<td>1327</td>
<td>6.8%</td>
<td>5.7%</td>
<td>87.3%</td>
<td>0.2%</td>
<td>100%</td>
<td>18.17</td>
</tr>
<tr>
<td>via.c4n</td>
<td>1366</td>
<td>11.1%</td>
<td>1.3%</td>
<td>87.6%</td>
<td>0%</td>
<td>100%</td>
<td>23.08</td>
</tr>
<tr>
<td>via.c5n</td>
<td>1202</td>
<td>3.4%</td>
<td>1.4%</td>
<td>95.0%</td>
<td>0.2%</td>
<td>100%</td>
<td>17.13</td>
</tr>
</tbody>
</table>

---

## Vertex Cover in Planar Graphs

<table>
<thead>
<tr>
<th>n</th>
<th>Variables Fixed (%)</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>68.4</td>
<td>4.06</td>
</tr>
<tr>
<td>2000</td>
<td>67.4</td>
<td>12.24</td>
</tr>
<tr>
<td>3000</td>
<td>65.5</td>
<td>30.90</td>
</tr>
<tr>
<td>4000</td>
<td>62.7</td>
<td>60.45</td>
</tr>
</tbody>
</table>


3 Pentium 4, 2.8 GHz, Windows XP, 512 MB
### Jumbo Vertex Cover in Planar Graphs

<table>
<thead>
<tr>
<th>Vertices</th>
<th>Computing Times (min)(^4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Planar Density</td>
</tr>
<tr>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>50,000</td>
<td>0.7</td>
</tr>
<tr>
<td>100,000</td>
<td>2.9</td>
</tr>
<tr>
<td>250,000</td>
<td>19.5</td>
</tr>
<tr>
<td>500,000</td>
<td>79.3</td>
</tr>
</tbody>
</table>

\(^4\) Averages over 3 experiments on a Xeon 3.06 GHz, XP, 3.5 GB RAM; ALL problems had 100% of their variables fixed.
## Documentation

### What is Quadratization?

Quadratization refers to the process of transforming a non-quadratic objective function in an optimization problem into a quadratic form, making it easier to solve using quadratic unconstrained binary optimization (QUBO) methods.

### Quadratization Techniques

- **One Dimensional Ising Models**

### Table: Average Computing Time

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Number of Spins</th>
<th>Branch, Cut &amp; Price $^5$</th>
<th>Biq Maq $^5$</th>
<th>QUBO $^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>100</td>
<td>699</td>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>92 079</td>
<td>388</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>N/A</td>
<td>993</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>N/A</td>
<td>6 567</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>N/A</td>
<td>34 572</td>
<td>21</td>
</tr>
<tr>
<td>3.0</td>
<td>100</td>
<td>256</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>13 491</td>
<td>293</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>61 271</td>
<td>1 034</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>55 795</td>
<td>3 594</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>55 528</td>
<td>8 496</td>
<td>5</td>
</tr>
</tbody>
</table>


$^6$ ALL problems were solved by QUBO.
## Larger One Dimensional Ising Models

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>$n$</th>
<th>Variables not fixed</th>
<th>QUBO Time (s)$^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>500</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>22</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>24</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>1250</td>
<td>20</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>32</td>
<td>124</td>
</tr>
<tr>
<td>3.0</td>
<td>500</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>1250</td>
<td>0</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>0</td>
<td>59</td>
</tr>
</tbody>
</table>

$^7$ Pentium M, 1.6 GHz 760 MB RAM
Outline

1 Quadratic Unconstrained Binary Optimization
   • Quadratic Pseudo-Boolean Functions
   • Representations and Bounds
   • Origin of Graph Cut Models
   • Network Model for General QUBO

2 Polynomial Time Preprocessing
   • Components of the Algorithm
   • Computational Results

3 What is Quadratization?
   • Quadratization
   • Submodular Functions

4 Quadratization Techniques
   • Penalty Function
   • Termwise Quadratization
   • Multiple Split of Terms
   • Splitting Off Common Parts
   • Results
Quadratization of PBFs

Given $f : \{0, 1\}^n \to \mathbb{R}$ find quadratic $g : \{0, 1\}^{n+m} \to \mathbb{R}$ such that

$$f(x) = \min_{y \in \{0,1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.$$ 

- Keep $m$ small!
- Have $g$ as submodular as possible!
- Do not introduce large coefficients!
- Have it ALL!

Rosenberg, 1975: All PBFs have polynomial sized quadratizations.

Zivny, Cohen and Jeavons, 2009: Not all submodular PBFs have submodular quadratizations.

Ishikawa, 2009, 2011: All PBFs have small quadratizations with no large coefficients.
Quadratization of PBFs

Given \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) find quadratic \( g : \{0, 1\}^{n+m} \rightarrow \mathbb{R} \) such that
\[
f(x) = \min_{y \in \{0, 1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.
\]

♦ Keep \( m \) small!
♦ Have \( g \) as submodular as possible!
♥ Do not introduce large coefficients!
♠ Have it ALL!

Rosenberg, 1975: All PBFs have polynomial sized quadratizations.
Zivny, Cohen and Jeavons, 2009: Not all submodular PBFs have submodular quadratizations.
Ishikawa, 2009, 2011: All PBFs have small quadratizations with no large coefficients.
Quadratization of PBFs

- Given $f : \{0, 1\}^n \to \mathbb{R}$ find quadratic $g : \{0, 1\}^{n+m} \to \mathbb{R}$ such that

$$f(x) = \min_{y \in \{0,1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.$$ 

- Keep $m$ small!
- Have $g$ as submodular as possible!
- Do not introduce large coefficients!
- Have it ALL!

Rosenberg, 1975: All PBFs have polynomial sized quadratizations.
Zivny, Cohen and Jeavons, 2009: Not all submodular PBFs have submodular quadratizations.
Ishikawa, 2009, 2011: All PBFs have small quadratizations with no large coefficients.
**Quadratization of PBFs**

- Given $f : \{0, 1\}^n \rightarrow \mathbb{R}$ find **quadratic** $g : \{0, 1\}^{n+m} \rightarrow \mathbb{R}$ such that

$$f(x) = \min_{y \in \{0,1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.$$ 

- ♣ Keep $m$ **small**!
- ♦ Have $g$ as **submodular** as possible!
- ♥ Do not introduce **large** coefficients!
- ♠ Have it **ALL**!

**Rosenberg, 1975**: All PBFs have **polynomial sized** quadratizations.

**Zivny, Cohen and Jeavons, 2009**: Not all submodular PBFs have **submodular** quadratizations.

**Ishikawa, 2009, 2011**: All PBFs have **small** quadratizations with no large coefficients.
Quadratization of PBFs

- Given \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) find quadratic \( g : \{0, 1\}^{n+m} \rightarrow \mathbb{R} \) such that

\[
    f(x) = \min_{y \in \{0,1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.
\]

- ♣ Keep \( m \) small!
- ♦ Have \( g \) as submodular as possible!
- ♥ Do not introduce large coefficients!
- ♠ Have it ALL!

Rosenberg, 1975: All PBFs have polynomial sized quadratizations.

Zivny, Cohen and Jeavons, 2009: Not all submodular PBFs have submodular quadratizations.

Ishikawa, 2009, 2011: All PBFs have small quadratizations with no large coefficients.
Quadratization of PBFs

- Given $f : \{0, 1\}^n \to \mathbb{R}$ find quadratic $g : \{0, 1\}^{n+m} \to \mathbb{R}$ such that

$$f(x) = \min_{y \in \{0, 1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.$$ 

- ♠ Keep $m$ small!
- ♦ Have $g$ as submodular as possible!
- ♥ Do not introduce large coefficients!
- ♣ Have it ALL!

**Rosenberg, 1975:** All PBFs have **polynomial sized** quadratizations.

**Zivny, Cohen and Jeavons, 2009:** Not all submodular PBFs have submodular quadratizations.

**Ishikawa, 2009, 2011:** All PBFs have **small** quadratizations with no large coefficients.
Quadratization of PBFs

Given \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) find quadratic \( g : \{0, 1\}^{n+m} \rightarrow \mathbb{R} \) such that

\[
f(x) = \min_{y \in \{0, 1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.
\]

♣ Keep \( m \) small!
♦ Have \( g \) as submodular as possible!
♥ Do not introduce large coefficients!
♠ Have it ALL!

Rosenberg, 1975: All PBFs have polynomial sized quadratizations.

Zivny, Cohen and Jeavons, 2009: Not all submodular PBFs have submodular quadratizations.

Ishikawa, 2009, 2011: All PBFs have small quadratizations with no large coefficients.
Quadratization of PBFs

Given $f : \{0, 1\}^n \rightarrow \mathbb{R}$ find quadratic $g : \{0, 1\}^{n+m} \rightarrow \mathbb{R}$ such that

$$f(x) = \min_{y \in \{0,1\}^m} g(x, y) \quad \forall \ x \in \{0, 1\}^n.$$ 

- ♠️ Keep $m$ small!
- ♦️ Have $g$ as submodular as possible!
- ♥️ Do not introduce large coefficients!
- ♣️ Have it ALL!

**Rosenberg, 1975**: All PBFs have polynomial sized quadratizations.

**Zivny, Cohen and Jeavons, 2009**: Not all submodular PBFs have submodular quadratizations.

**Ishikawa, 2009, 2011**: All PBFs have small quadratizations with no large coefficients.
Outline

1. Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2. Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3. What is Quadratization?
   - Quadratization
   - Submodular Functions

4. Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Submodular PBFs

- A PBF $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is submodular if
  \[ f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \; x, y \in \{0, 1\}^n. \]

- Polynomial recognition if $\deg(f) \leq 3$. 
  \( \text{(Billionnet and Minoux, 1985)} \)

- Recognition is NP-hard if $\deg(f) \geq 4$. 
  \( \text{(Gallo and Simeone, 1989; Crama 1989)} \)

- A QPBF is submodular iff it has no positive quadratic terms. 
  \( \text{(Nemhauser and Wolsey, 1981)} \)

- A submodular QPBO is solved by the network based preprocessing. 
  \( \text{(Hammer, 1965)} \)

- Which PBFs have submodular quadratization?
- How to recognize if a PBF has a submodular quadratization?
Submodular PBFs

- A PBF $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is submodular if
  \[ f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \ x, y \in \{0, 1\}^n. \]

- **Polynomial recognition** if $\deg(f) \leq 3$.  
  \[(\text{Billionnet and Minoux, 1985})\]

- Recognition is NP-hard if $\deg(f) \geq 4$.  
  \[(\text{Gallo and Simeone, 1989; Crama 1989})\]

- A QPBF is submodular iff it has no positive quadratic terms.  
  \[(\text{Nemhauser and Wolsey, 1981})\]

- A submodular QPBO is solved by the network based preprocessing.  
  \[(\text{Hammer, 1965})\]

- Which PBFs have submodular quadratization?
- How to recognize if a PBF has a submodular quadratization?
**Submodular PBFs**

- A PBF \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) is **submodular** if
  \[
  f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \ x, y \in \{0, 1\}^n.
  \]

- **Polynomial recognition** if \( \deg(f) \leq 3. \)  
  \((\text{Billionnet and Minoux, 1985})\)

- **Recognition is NP-hard** if \( \deg(f) \geq 4. \)  
  \((\text{Gallo and Simeone, 1989; Crama 1989})\)

- A QPBF is submodular iff it has no positive quadratic terms.  
  \((\text{Nemhauser and Wolsey, 1981})\)

- A submodular QPBO is solved by the network based preprocessing.  
  \((\text{Hammer, 1965})\)

- Which PBFs have submodular quadratization?
- How to recognize if a PBF has a submodular quadratization?
## Submodular PBFs

- A PBF \( f : \{0, 1\}^n \rightarrow \mathbb{R} \) is submodular if
  \[
  f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \ x, y \in \{0, 1\}^n.
  \]

- **Polynomial recognition** if \( \deg(f) \leq 3 \).
  \((\text{Billionnet and Minoux, 1985})\)

- **Recognition is NP-hard** if \( \deg(f) \geq 4 \).
  \((\text{Gallo and Simeone, 1989; Crama 1989})\)

- A QPBF is submodular iff it has no positive quadratic terms.
  \((\text{Nemhauser and Wolsey, 1981})\)

- A submodular QPBO is solved by the network based preprocessing.
  \((\text{Hammer, 1965})\)

- Which PBFs have submodular quadratization?
- How to recognize if a PBF has a submodular quadratization?
Submodular PBFs

- A PBF $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is submodular if
  \[ f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \quad x, y \in \{0, 1\}^n. \]

- **Polynomial recognition** if $\text{deg}(f) \leq 3$. \hfill (Billionnet and Minoux, 1985)

- **Recognition is NP-hard** if $\text{deg}(f) \geq 4$. \hfill (Gallo and Simeone, 1989; Crama 1989)

- A QPBF is submodular iff it has no positive quadratic terms. \hfill (Nemhauser and Wolsey, 1981)

- A submodular QPBO is solved by the network based preprocessing. \hfill (Hammer, 1965)

- Which PBFs have submodular quadratization?
- How to recognize if a PBF has a submodular quadratization?
Submodular PBFs

- A PBF \( f : \{0, 1\}^n \to \mathbb{R} \) is submodular if
  \[
f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \ x, y \in \{0, 1\}^n.
  \]

- **Polynomial recognition** if \( \deg(f) \leq 3 \).
  \((\text{Billionnet and Minoux, 1985})\)

- **Recognition is NP-hard** if \( \deg(f) \geq 4 \).
  \((\text{Gallo and Simeone, 1989}; \text{Crama 1989})\)

- A QPBF is submodular iff it has no positive quadratic terms.
  \((\text{Nemhauser and Wolsey, 1981})\)

- A submodular QPBO is solved by the network based preprocessing.
  \((\text{Hammer, 1965})\)

- **Which PBFs have submodular quadratization?**

  - How to recognize if a PBF has a submodular quadratization?
Submodular PBFs

- A PBF $f : \{0, 1\}^n \to \mathbb{R}$ is submodular if

  $$f(x \land y) + f(x \lor y) \leq f(x) + f(y) \quad \forall \ x, y \in \{0, 1\}^n.$$ 

- **Polynomial recognition** if $\deg(f) \leq 3$.
  
  *(Billionnet and Minoux, 1985)*

- **Recognition is NP-hard** if $\deg(f) \geq 4$.
  
  *(Gallo and Simeone, 1989; Crama 1989)*

- A QPBF is submodular iff it has no positive quadratic terms.
  
  *(Nemhauser and Wolsey, 1981)*

- A submodular QPBO is solved by the network based preprocessing.
  
  *(Hammer, 1965)*

- Which PBFs have submodular quadratization?

- How to recognize if a PBF has a submodular quadratization?
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Rosenberg's Penalty Functions Method (1975)

\[ p(x, y, w) = xy - 2xw - 2yw + 3w = \begin{cases} 
0 & \text{if } w = xy, \\
\geq 1 & \text{if } w \neq xy 
\end{cases} \]

\[ f(x, y, \ldots) = xyA + B = \min_{w \in \{0, 1\}} wA + B + Mp(x, y, w) \]

if \( M \) is large enough.

- Many positive quadratic terms with large coefficients (recursion!), even if the input is subodular.

- NP-hard to find a quadratization in this way with the minimum number of new variables.

- Not possible to substitute the product of 3 or more variables with a single new variable.
Rosenberg’s Penalty Functions Method (1975)

\[ p(x, y, w) = xy - 2xw - 2yw + 3w = \begin{cases} 
0 & \text{if } w = xy, \\
\geq 1 & \text{if } w \neq xy 
\end{cases} \]

\[ f(x, y, \ldots) = xyA + B = \min_{w \in \{0,1\}} wA + B + Mp(x, y, w) \]

if \( M \) is large enough.

- Many positive quadratic terms with large coefficients (recursion!), even if the input is subodular.

- **NP-hard** to find a quadratization in this way with the minimum number of new variables.

- Not possible to substitute the product of 3 or more variables with a single new variable.
Rosenberg’s Penalty Functions Method (1975)

\[ p(x, y, w) = xy - 2xw - 2yw + 3w = \begin{cases} 
0 & \text{if } w = xy, \\
\geq 1 & \text{if } w \neq xy
\end{cases} \]

\[ f(x, y, ...) = xyA + B = \min_{w \in \{0,1\}} wA + B + Mp(x, y, w) \]

if \( M \) is large enough.

- Many positive quadratic terms with large coefficients (recursion!), even if the input is subodular.

- \textbf{NP-hard} to find a quadratization in this way with the minimum number of new variables.

- Not possible to substitute the product of 3 or more variables with a single new variable.
Rosenberg’s Penalty Functions Method (1975)

\[
p(x, y, w) = xy - 2xw - 2yw + 3w = \begin{cases} 
0 & \text{if } w = xy, \\
\geq 1 & \text{if } w \neq xy
\end{cases}
\]

\[
f(x, y, ...) = xyA + B = \min_{w \in \{0,1\}} wA + B + Mp(x, y, w)
\]

if \( M \) is large enough.

- Many positive quadratic terms with large coefficients (recursion!), even if the input is subodular.

- **NP-hard** to find a quadratization in this way with the minimum number of new variables.

- Not possible to substitute the product of 3 or more variables with a single new variable.
Rosenberg’s Penalty Functions Method (1975)

\[ p(x, y, w) = xy - 2xw - 2yw + 3w = \begin{cases} 
0 & \text{if } w = xy, \\
\geq 1 & \text{if } w \neq xy
\end{cases} \]

\[ f(x, y, ... ) = xyA + B = \min_{w \in \{0, 1\}} wA + B + Mp(x, y, w) \]

if \( M \) is large enough.

- Many positive quadratic terms with large coefficients (recursion!), even if the input is subodular.

- **NP-hard** to find a quadratization in this way with the minimum number of new variables.

- Not possible to substitute the product of 3 or more variables with a single new variable.
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
**Negative Terms**

- **Kolmogorov and Zabih (2004), Fredman and Drineas (2005):**
  
  \[-x_1x_2\cdots x_d = \min_{w \in \{0,1\}} w(d - 1 - x_1 - x_2 \cdots - x_d)\]

- **Rother, Kohli, Feng and Jia (2009):**
  
  \[-\prod_{j \in N} \overline{x}_j \prod_{j \in P} x_j = \min_{u,v \in \{0,1\}} -uv + u \sum_{j \in N} x_j + v \sum_{j \in P} \overline{x}_j\]

- Only one or two new variables per term; at most one positive quadratic term; no large coefficients.

**Theorem (vs. Billionet and Minoux (1985))**

*Cubic submodular functions have submodular quadratization of polynomial size with no large coefficients.*
Negative Terms

- Kolmogorov and Zabih (2004), Fredman and Drineas (2005):
  \[-x_1x_2 \cdots x_d = \min_{w \in \{0,1\}} w(d - 1 - x_1 - x_2 \cdots - x_d)\]

- Rother, Kohli, Feng and Jia (2009):
  \[-\prod_{j \in N} \overline{x}_j \prod_{j \in P} x_j = \min_{u,v \in \{0,1\}} -uv + u \sum_{j \in N} x_j + v \sum_{j \in P} \overline{x}_j\]

  Only one or two new variables per term; at most one positive quadratic term; no large coefficients.

Theorem (vs. Billionet and Minoux (1985))

Cubic submodular functions have submodular quadratization of polynomial size with no large coefficients.
**Negative Terms**

- **Kolmogorov and Zabih (2004), Fredman and Drineas (2005):**
  \[-x_1 x_2 \cdots x_d = \min_{w \in \{0, 1\}} w(d - 1 - x_1 - x_2 \cdots - x_d)\]

- **Rother, Kohli, Feng and Jia (2009):**
  \[-\prod_{j \in N} \overline{x}_j \prod_{j \in P} x_j = \min_{u, v \in \{0, 1\}} -uv + u \sum_{j \in N} x_j + v \sum_{j \in P} \overline{x}_j\]

- **Only one or two new variables per term; at most one positive quadratic term; no large coefficients.**

**Theorem (vs. Billionet and Minoux (1985))**

*Cubic submodular functions have submodular quadratization of polynomial size with no large coefficients.*
Negative Terms

- **Kolmogorov and Zabih (2004), Fredman and Drineas (2005):**
  \[-x_1 x_2 \cdots x_d = \min_{w \in \{0,1\}} w(d - 1 - x_1 - x_2 \cdots - x_d)\]

- **Rother, Kohli, Feng and Jia (2009):**
  \[-\prod_{j \in N} \overline{x}_j \prod_{j \in P} x_j = \min_{u,v \in \{0,1\}} -uv + u \sum_{j \in N} x_j + v \sum_{j \in P} \overline{x}_j\]

- **Only one or two new variables per term; at most one positive quadratic term; no large coefficients.**

Theorem (vs. Billionet and Minoux (1985))

*Cubic submodular functions have submodular quadratization of polynomial size with no large coefficients.*
Positive Terms

- **Ishikawa (2009, 2011):**

\[ \prod_{j=1}^{d} x_j = S_2(x) + \min_{w \in \{0,1\}^k} B(w) - 2A(w)S_1(x) + \rho [S_1(x) - d + 1] \]

where \( d = 2k + 2 - \rho \), \( \rho \in \{0,1\} \), and

\[
\begin{align*}
S_1(x) &= \sum_{j=1}^{d} x_j \\
S_2(x) &= \sum_{1 \leq i < j \leq d} x_i x_j \\
A(w) &= \sum_{j=1}^{k} w_j \\
B(w) &= \sum_{j=1}^{k} (4j - 1) w_j
\end{align*}
\]

- Only \( \approx d/2 \) new variables per term; no large coefficients; many positive quadratic terms.
**Positive Terms**

- **Ishikawa (2009, 2011):**

\[
\prod_{j=1}^{d} x_j = S_2(x) + \min_{w \in \{0, 1\}^k} B(w) - 2A(w)S_1(x) + \rho [S_1(x) - d + 1]
\]

where \( d = 2k + 2 - \rho \), \( \rho \in \{0, 1\} \), and

\[
S_1(x) = \sum_{j=1}^{d} x_j \quad S_2(x) = \sum_{1 \leq i < j \leq d} x_i x_j
\]

\[
A(w) = \sum_{j=1}^{k} w_j \quad B(w) = \sum_{j=1}^{4j-1} w_j
\]

- **Only \( \approx d/2 \) new variables per term; no large coefficients; many positive quadratic terms.**
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Multiple Splits

Assume that $\phi_i(w) \in \{0, 1\}$ for $i \in [q]$, $w \in \{0, 1\}^p$ such that

$$\min_{w \in \{0, 1\}^p} \sum_{i=1}^{q} \phi_i(w) = 1,$$

and

$$\forall I \subsetneq [q] \exists w^* \in \{0, 1\}^p \text{ s.t. } \sum_{i \in I} \phi_i(w^*) = 0.$$

For instance $\phi_1 = w_1$, $\phi_2 = w_2$, and $\phi_3 = w_1 w_2$ is such a system.

Theorem

If $P_i, i \in [q]$ are subsets of indices covering $[d]$, then we have

$$\prod_{j=1}^{d} x_j = \min_{w \in \{0, 1\}^p} \sum_{i=1}^{q} \phi_i(w) \prod_{j \in P_i} x_j.$$

With $p = \lceil \log q \rceil$ new variables we can split a degree $d = kq$ term into $q$ terms of degree $k + p$. 
**Multiple Splits**

Assume that $\phi_i(w) \in \{0, 1\}$ for $i \in [q]$, $w \in \{0, 1\}^p$ such that

$$\min_{w \in \{0, 1\}^p} \sum_{i=1}^q \phi_i(w) = 1,$$

and

$$\forall I \subsetneq [q] \ \exists w^* \in \{0, 1\}^p \text{ s.t. } \sum_{i \in I} \phi_i(w^*) = 0.$$

For instance $\phi_1 = w_1$, $\phi_2 = w_2$, and $\phi_3 = \overline{w_1} \overline{w_2}$ is such a system.

**Theorem**

If $P_i, i \in [q]$ are subsets of indices covering $[d]$, then we have

$$\prod_{j=1}^d x_j = \min_{w \in \{0, 1\}^p} \sum_{i=1}^q \phi_i(w) \prod_{j \in P_i} x_j.$$

With $p = \lceil \log q \rceil$ new variables we can split a degree $d = kq$ term into $q$ terms of degree $k + p$. 

**QUBO Polynomial Time Preprocessing**

What is Quadratization?

Quadratization Techniques
Multiple Splits

Assume that $\phi_i(w) \in \{0, 1\}$ for $i \in [q]$, $w \in \{0, 1\}^p$ such that

$$\min_{w \in \{0, 1\}^p} \sum_{i=1}^{q} \phi_i(w) = 1,$$

and

$$\forall I \subsetneq [q] \exists w^* \in \{0, 1\}^p \text{ s.t. } \sum_{i \in I} \phi_i(w^*) = 0.$$

For instance $\phi_1 = w_1$, $\phi_2 = w_2$, and $\phi_3 = \overline{w_1} \overline{w_2}$ is such a system.

Theorem

If $P_i$, $i \in [q]$ are subsets of indices covering $[d]$, then we have

$$\prod_{j=1}^{d} x_j = \min_{w \in \{0, 1\}^p} \sum_{i=1}^{q} \phi_i(w) \prod_{j \in P_i} x_j.$$

With $p = \lceil \log q \rceil$ new variables we can split a degree $d = kq$ term into $q$ terms of degree $k + p$. 
**Multiple Splits**

Assume that $\phi_i(w) \in \{0, 1\}$ for $i \in [q]$, $w \in \{0, 1\}^p$ such that

$$\min_{w \in \{0,1\}^p} \sum_{i=1}^q \phi_i(w) = 1,$$

and

$$\forall I \subsetneq [q] \quad \exists w^* \in \{0, 1\}^p \text{ s.t. } \sum_{i \in I} \phi_i(w^*) = 0.$$

For instance $\phi_1 = w_1$, $\phi_2 = w_2$, and $\phi_3 = \overline{w_1} \overline{w_2}$ is such a system.

**Theorem**

If $P_i$, $i \in [q]$ are subsets of indices covering $[d]$, then we have

$$\prod_{j=1}^d x_j = \min_{w \in \{0,1\}^p} \sum_{i=1}^q \phi_i(w) \prod_{j \in P_i} x_j.$$

With $p = \lceil \log q \rceil$ new variables we can split a degree $d = kq$ term into $q$ terms of degree $k + p$. 
Outline

1 Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2 Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3 What is Quadratization?
   - Quadratization
   - Submodular Functions

4 Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
Let $C \subseteq [n]$, $\mathcal{H} \subseteq 2^{[n]\setminus C}$, and consider the following fragment of a pseudo-Boolean function:

$$g(x) = \sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in C \cup H} x_j,$$

where $\alpha_H \geq 0$ for all $H \in \mathcal{H}$.

**Theorem (Set of Positive Terms)**

$$g(x) = \min_{w \in \{0,1\}} \left( \sum_{H \in \mathcal{H}} \alpha_H \right) w \prod_{j \in C} x_j + \sum_{H \in \mathcal{H}} \alpha_H w \prod_{j \in H} x_j.$$

**Theorem (Set of Negative Terms)**

$$-g(x) = \min_{w \in \{0,1\}} \sum_{H \in \mathcal{H}} \alpha_H w \left( 1 - \prod_{j \in C} x_j - \prod_{j \in H} x_j \right).$$
Let $C \subseteq [n]$, $\mathcal{H} \subseteq 2^{[n]\setminus C}$, and consider the following fragment of a pseudo-Boolean function:

$$g(x) = \sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in C \cup H} x_j,$$

where $\alpha_H \geq 0$ for all $H \in \mathcal{H}$.

**Theorem (Set of Positive Terms)**

$$g(x) = \min_{w \in \{0,1\}} \left( \sum_{H \in \mathcal{H}} \alpha_H \right) w \prod_{j \in C} x_j + \sum_{H \in \mathcal{H}} \alpha_H w \prod_{j \in H} x_j.$$

**Theorem (Set of Negative Terms)**

$$-g(x) = \min_{w \in \{0,1\}} \sum_{H \in \mathcal{H}} \alpha_H w \left( 1 - \prod_{j \in C} x_j - \prod_{j \in H} x_j \right).$$
Let $C \subseteq [n]$, $\mathcal{H} \subseteq 2^{[n]\setminus C}$, and consider the following fragment of a pseudo-Boolean function:

$$g(x) = \sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in C \cup H} x_j,$$

where $\alpha_H \geq 0$ for all $H \in \mathcal{H}$.

**Theorem (Set of Positive Terms)**

$$g(x) = \min_{w \in \{0,1\}} \left( \sum_{H \in \mathcal{H}} \alpha_H \right) w \prod_{j \in C} x_j + \sum_{H \in \mathcal{H}} \alpha_H \overline{w} \prod_{j \in H} x_j.$$

**Theorem (Set of Negative Terms)**

$$-g(x) = \min_{w \in \{0,1\}} \sum_{H \in \mathcal{H}} \alpha_H w \left( 1 - \prod_{j \in C} x_j - \prod_{j \in H} x_j \right).$$
Outline

1. Quadratic Unconstrained Binary Optimization
   - Quadratic Pseudo-Boolean Functions
   - Representations and Bounds
   - Origin of Graph Cut Models
   - Network Model for General QUBO

2. Polynomial Time Preprocessing
   - Components of the Algorithm
   - Computational Results

3. What is Quadratization?
   - Quadratization
   - Submodular Functions

4. Quadratization Techniques
   - Penalty Function
   - Termwise Quadratization
   - Multiple Split of Terms
   - Splitting Off Common Parts
   - Results
**Corollary**

A PBF in $n$ variables, with $t$ terms of degree $d$ has a quadratization with

$$\approx n + k\binom{n}{k} + \frac{td}{k}$$

new variables and with at most $n - 1$ positive quadratic terms, for any $k \geq 1$.

Ishikawa’s method provides a quadratization with $\approx n + \frac{td}{2}$ new variables and $\max\{\binom{n}{2}, t\binom{d}{2}\}$ positive quadratic terms.

<table>
<thead>
<tr>
<th></th>
<th>New variables</th>
<th># positive terms</th>
<th># terms</th>
<th>% fixed by QPBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ishikawa</td>
<td>224,346</td>
<td>421,897</td>
<td>1,133,811</td>
<td>80.4%</td>
</tr>
<tr>
<td>Our method</td>
<td>236,806</td>
<td>38,343</td>
<td>677,183</td>
<td>96.1%</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>+6%</td>
<td>−90%</td>
<td>−40%</td>
<td>+20%</td>
</tr>
</tbody>
</table>

Figure: Performance comparison of reductions, on Ishikawa’s benchmarks. Relative performance of our method is shown as $\Delta$. (Joint work with Alexander Fix and Ramin Zabih (Cornell University).)
Corollary

A PBF in \(n\) variables, with \(t\) terms of degree \(d\) has a quadratization with 
\[\approx n + k\binom{n}{k} + \frac{td}{k}\] new variables and with at most \(n - 1\) positive quadratic terms, for any \(k \geq 1\).

Ishikawa’s method provides a quadratization with 
\[\approx n + \frac{td}{2}\] new variables and \[\max\{\binom{n}{2}, t\binom{d}{2}\}\] positive quadratic terms.

<table>
<thead>
<tr>
<th></th>
<th>New variables</th>
<th># positive terms</th>
<th># terms</th>
<th>% fixed by QPBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ishikawa</td>
<td>224,346</td>
<td>421,897</td>
<td>1,133,811</td>
<td>80.4%</td>
</tr>
<tr>
<td>Our method</td>
<td>236,806</td>
<td>38,343</td>
<td>677,183</td>
<td>96.1%</td>
</tr>
<tr>
<td>(\Delta)</td>
<td>+6%</td>
<td>-90%</td>
<td>-40%</td>
<td>+20%</td>
</tr>
</tbody>
</table>

Figure: Performance comparison of reductions, on Ishikawa’s benchmarks. Relative performance of our method is shown as \(\Delta\). (Joint work with Alexander Fix and Ramin Zabih (Cornell University).)
Corollary

A PBF in $n$ variables, with $t$ terms of degree $d$ has a quadratization with
$\approx n + k(n) + \frac{td}{k}$ new variables and with at most $n - 1$ positive quadratic
terms, for any $k \geq 1$.

Ishikawa’s method provides a quadratization with $\approx n + \frac{td}{2}$ new variables
and $\max\{\binom{n}{2}, t\binom{d}{2}\}$ positive quadratic terms.

<table>
<thead>
<tr>
<th></th>
<th>New variables</th>
<th># positive terms</th>
<th># terms</th>
<th>% fixed by QPBO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ishikawa</td>
<td>224,346</td>
<td>421,897</td>
<td>1,133,811</td>
<td>80.4%</td>
</tr>
<tr>
<td>Our method</td>
<td>236,806</td>
<td>38,343</td>
<td>677,183</td>
<td>96.1%</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>+6%</td>
<td>−90%</td>
<td>−40%</td>
<td>+20%</td>
</tr>
</tbody>
</table>

Figure: Performance comparison of reductions, on Ishikawa’s benchmarks.
Relative performance of our method is shown as $\Delta$. (Joint work with Alexander
Fix and Ramin Zabih (Cornell University).)