LP + Branch-and-Cut for solving certain hard Quadratic Unconstrained Binary Optimization (QUBO) problems

Gabriel Tavares$^{(\star)}$ Endre Boros$^{(\star\star)}$
Peter L. Hammer[1]

Fair Isaac*

Rutgers University**
RUTCOR – Rutgers Center for Operations Research

Fields Industrial Optimization Seminar, Toronto, Oct-2008

[1] (1936–2006)

Outline

Outline

What is QUBO?

QUBO (or Quadratic Unconstrained Binary Optimization) is the problem

$$\min_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x}),$$

concerning the minimization of a quadratic pseudo–Boolean function f given by

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

where c_0, c_i for $i = 1,\cdots,n$ and c_{ij} for $1 \leqslant i < j \leqslant n$ are given reals.

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Motivation: Wide variety of Applications

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Outline

Linearization Model for QUBO

The standard linearization model to compute the minimum value of a quadratic pseudo–Boolean function is

$$\min \quad \left(c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \leqslant i < j \leqslant n} c_{ij} y_{ij} \right)$$

subject to

$$\begin{array}{ll}
y_{ij} \leqslant x_i, & 1 \leqslant i < j \leqslant n,\ c_{ij} < 0, \\
y_{ij} \leqslant x_j, & 1 \leqslant i < j \leqslant n,\ c_{ij} < 0, \\
y_{ij} \geqslant x_i + x_j - 1, & 1 \leqslant i < j \leqslant n,\ c_{ij} > 0, \\
y_{ij} \geqslant 0, & 1 \leqslant i < j \leqslant n, \\
x_j \in \{0, 1\}, & j \in \mathbf{V},
\end{array}$$

whose optimal solutions $\mathbf{x}^\star \in \{0,1\}^n$ are minimizers of f.

Linearization Model for QUBO

The roof–dual bound $C_2(f)$ is obtained by relaxing the integrality in the linearization model [Hammer, Hansen and Simeone '84], i.e.

$$
\begin{aligned}
C_2(f) = \quad & \min \quad \left(c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \leqslant i < j \leqslant n} c_{ij} y_{ij} \right) \\
& \text{subject to} \\
& \qquad y_{ij} \leqslant x_i, \qquad\qquad\quad 1 \leqslant i < j \leqslant n,\ c_{ij} < 0, \\
& \qquad y_{ij} \leqslant x_j, \qquad\qquad\quad 1 \leqslant i < j \leqslant n,\ c_{ij} < 0, \\
& \qquad y_{ij} \geqslant x_i + x_j - 1, \quad 1 \leqslant i < j \leqslant n,\ c_{ij} > 0, \\
& \qquad y_{ij} \geqslant 0, \qquad\qquad\quad 1 \leqslant i < j \leqslant n, \\
& \qquad x_j \in [0, 1], \qquad\quad\ \ j \in \mathbf{V}.
\end{aligned}
$$

Persistencies of the Linearization Model

Half-Integral Solutions Theorem [Balinski' 68]

Every extreme point of the relaxation of the linearization model has components 0, $\frac{1}{2}$ or 1.

Persistency Theorem [Hammer, Hansen and Simeone' 84]

If there exists an optimal solution $\mathbf{x}^+$ of the relaxation of the linearization model having certain variables S with 0–1 values, then there is an optimal solution $\mathbf{x}^\star$ to the linearization model such that $x_j^\star = x_j^+$, $j \in S$.

The identification of these variables (called persistencies) can be very helpful in simplifying the QUBO problem.

Persistencies of the Linearization Model

Half-Integral Solutions Theorem [Balinski' 68]

Every extreme point of the relaxation of the linearization model has components 0, $\frac{1}{2}$ or 1.

Persistency Theorem [Hammer, Hansen and Simeone' 84]

If there exists an optimal solution $\mathbf{x}^+$ of the relaxation of the linearization model having certain variables S with 0–1 values, then there is an optimal solution $\mathbf{x}^\star$ to the linearization model such that $x_j^\star = x_j^+,\ j \in S$.

The identification of these variables (called persistencies) can be very helpful in simplifying the QUBO problem.

Persistencies of the Linearization Model

Half-Integral Solutions Theorem [Balinski' 68]

Every extreme point of the relaxation of the linearization model has components 0, $\frac{1}{2}$ or 1.

Persistency Theorem [Hammer, Hansen and Simeone' 84]

If there exists an optimal solution $\mathbf{x}^+$ of the relaxation of the linearization model having certain variables S with 0–1 values, then there is an optimal solution $\mathbf{x}^\star$ to the linearization model such that $x_j^\star = x_j^+$, $j \in S$.

The identification of these variables (called persistencies) can be very helpful in simplifying the QUBO problem.

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O(\text{max-flow}(2n, 2m) + \text{strong-components}(2n, 2m))$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique.
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O(\text{max-flow}(2n, 2m) + \text{strong components}(2n, 2m))$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O(\text{max-flow}(2n, 2m) + \text{strong-components}(2n, 2m))$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O(\text{max-flow}(2n, 2m) + \text{strong-components}(2n, 2m))$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O\left(\text{max-flow}\left(2n, 2m\right) + \text{strong-components}\left(2n, 2m\right)\right)$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O\left(\text{max-flow}\left(2n, 2m\right) + \text{strong-components}\left(2n, 2m\right)\right)$$

Properties of Persistencies

Questions

- How to find a maximal set of persistencies?
- How to find a maximum set of persistencies?

New Persistency Results

- Any maximal set of persistencies is also maximum possible for the relaxed linearization model
- The maximum set of persistencies of the relaxed linearization model is unique
- The maximum set of persistencies of the relaxed linearization model can be computed in

$$O\left(\text{max-flow}\left(2n, 2m\right) + \text{strong-components}\left(2n, 2m\right)\right)$$

Properties of Persistencies

The above result is proved by

Using the equivalence between posiform maximization and the weigthed vertex packing problem of graphs

Consequently

The linearization models of general Pseudo–Boolean optimization problems also satisfy the previous persistency results

Properties of Persistencies

The above result is proved by

Using the equivalence between posiform maximization and the weigthed vertex packing problem of graphs

Consequently

The linearization models of general Pseudo–Boolean optimization problems also satisfy the previous persistency results

Outline

1. Introduction
2. Linearizations and Persistencies
3. Lower Bounds
4. A Branch-And-Cut Exact Method

If the Roof–Dual is not Sharp, then How to Improve the Bound?

Hierarchy of Bounds

- Boros, Crama and Hammer '90 presented a hierarchy of bounds

$$C_2(f) \leqslant C_3(f) \leqslant C_4(f) \leqslant \cdots \leqslant C_n(f) = \min(f)$$

for QUBO

- $C_2(f)$ corresponds to the roof–dual value of f
- $C_3(f)$ corresponds to the cubic–dual of f [Boros, Crama and Hammer '92]
- $C_4(f)$ corresponds to the square–dual of f
- C_2, C_3 and C_4 are well characterized by LP

If the Roof–Dual is not Sharp, then How to Improve the Bound?

Hierarchy of Bounds

- Boros, Crama and Hammer '90 presented a hierarchy of bounds

$$C_2(f) \leqslant C_3(f) \leqslant C_4(f) \leqslant \cdots \leqslant C_n(f) = \min(f)$$

for QUBO

- $C_2(f)$ corresponds to the roof–dual value of f
- $C_3(f)$ corresponds to the cubic–dual of f [Boros, Crama and Hammer '92]
- $C_4(f)$ corresponds to the square–dual of f
- C_2, C_3 and C_4 are well characterized by LP

If the Roof–Dual is not Sharp, then How to Improve the Bound?

Hierarchy of Bounds

- Boros, Crama and Hammer '90 presented a hierarchy of bounds

$$C_2(f) \leqslant C_3(f) \leqslant C_4(f) \leqslant \cdots \leqslant C_n(f) = \min(f)$$

  for QUBO
- $C_2(f)$ corresponds to the roof–dual value of f
- $C_3(f)$ corresponds to the cubic–dual of f [Boros, Crama and Hammer '92]
- $C_4(f)$ corresponds to the square–dual of f
- C_2, C_3 and C_4 are well characterized by LP

If the Roof–Dual is not Sharp, then How to Improve the Bound?

Hierarchy of Bounds

- Boros, Crama and Hammer '90 presented a hierarchy of bounds

$$C_2(f) \leqslant C_3(f) \leqslant C_4(f) \leqslant \cdots \leqslant C_n(f) = \min(f)$$

  for QUBO
- $C_2(f)$ corresponds to the roof–dual value of f
- $C_3(f)$ corresponds to the cubic–dual of f [Boros, Crama and Hammer '92]
- $C_4(f)$ corresponds to the square–dual of f
- C_2, C_3 and C_4 are well characterized by LP

If the Roof–Dual is not Sharp, then How to Improve the Bound?

Hierarchy of Bounds

- Boros, Crama and Hammer '90 presented a hierarchy of bounds

$$C_2(f) \leqslant C_3(f) \leqslant C_4(f) \leqslant \cdots \leqslant C_n(f) = \min(f)$$

  for QUBO
- $C_2(f)$ corresponds to the roof–dual value of f
- $C_3(f)$ corresponds to the cubic–dual of f [Boros, Crama and Hammer '92]
- $C_4(f)$ corresponds to the square–dual of f
- C_2, C_3 and C_4 are well characterized by LP

Outline

Linearization Model (LM) + Cuts

Let us consider again the relaxation of the LM

$$
\begin{array}{rll}
C_2(f) = & \min & \left(c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \leqslant i < j \leqslant n} c_{ij} y_{ij} \right) \\
& \text{subject to} & \\
& \quad y_{ij} \leqslant x_i, & 1 \leqslant i < j \leqslant n,\ c_{ij} \neq 0, \\
& \quad y_{ij} \leqslant x_j, & 1 \leqslant i < j \leqslant n,\ c_{ij} \neq 0, \\
& \quad y_{ij} \geqslant x_i + x_j - 1, & 1 \leqslant i < j \leqslant n,\ c_{ij} \neq 0, \\
& \quad y_{ij} \geqslant 0, & 1 \leqslant i < j \leqslant n, \\
& \quad x_j \in [0, 1], & j \in \mathbf{V}.
\end{array}
$$

Linearization Model (LM) + Cuts

Consider the C_3 cuts

- Consist of the subset of triangle inequalities

$$\mathbf{W}(\mathcal{S}) = \left\{ (\mathbf{x},\mathbf{y}) \;\middle|\; \begin{array}{rrrl} x_i & +x_j & +x_k & -y_{i,j} - y_{i,k} - y_{j,k} \leqslant 1, \\ -x_i & & & +y_{i,j} + y_{i,k} - y_{j,k} \leqslant 0, \\ & -x_j & & +y_{i,j} - y_{i,k} + y_{j,k} \leqslant 0, \\ & & -x_k & -y_{i,j} + y_{i,k} + y_{j,k} \leqslant 0, \end{array} \left(\begin{array}{c} 1 \leqslant i < j < k \leqslant n \\ (i,j,k) \in \mathcal{S} \end{array} \right) \right\}.$$

- $\mathcal{S}$ represents the set of triplets (i,j,k) corresponding to the triangle inequalities involving variables x_i, x_j and x_k. Four basic cases are considered:
  - $\mathcal{S}_0 = \left\{(i,j,k) \in V^3 \,\middle|\, c_{ij} c_{ik} c_{jk} \neq 0 \right\}$
  - $\mathcal{S}_1 = \left\{(i,j,k) \in V^3 \,\middle|\, c_{ij} \neq 0 \text{ and } (c_{ik} \neq 0 \text{ or } c_{jk} \neq 0) \right\}$
  - $\mathcal{S}_2 = \left\{(i,j,k) \in V^3 \,\middle|\, c_{ij} \neq 0 \right\}$
  - $\mathcal{S}_3 = \left\{(i,j,k) \in V^3 \,\middle|\, c_{ij} \neq 0 \text{ or } c_{ik} \neq 0 \text{ or } c_{jk} \neq 0 \right\}$
- Theorem: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_3)$
- Conjecture: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_2)$

Linearization Model (LM) + Cuts

Consider the C_3 cuts

- Consist of the subset of triangle inequalities

$$\mathbf{W}(\mathcal{S}) = \left\{ (\mathbf{x},\mathbf{y}) \;\middle|\; \begin{array}{rrrl} x_i & +x_j & +x_k & -y_{i,j} - y_{i,k} - y_{j,k} \leqslant 1, \\ -x_i & & & +y_{i,j} + y_{i,k} - y_{j,k} \leqslant 0, \\ & -x_j & & +y_{i,j} - y_{i,k} + y_{j,k} \leqslant 0, \\ & & -x_k & -y_{i,j} + y_{i,k} + y_{j,k} \leqslant 0, \end{array} \quad \left(\begin{array}{c} 1 \leqslant i < j < k \leqslant n \\ (i,j,k) \in \mathcal{S} \end{array} \right) \right\}.$$

- $\mathcal{S}$ represents the set of triplets (i,j,k) corresponding to the triangle inequalities involving variables x_i, x_j and x_k. Four basic cases are considered:
  - $\mathcal{S}_0 = \{(i,j,k) \in V^3 \mid c_{ij}c_{ik}c_{jk} \neq 0\}$
  - $\mathcal{S}_1 = \{(i,j,k) \in V^3 \mid c_{ij} \neq 0 \text{ and } (c_{ik} \neq 0 \text{ or } c_{jk} \neq 0)\}$
  - $\mathcal{S}_2 = \{(i,j,k) \in V^3 \mid c_{ij} \neq 0\}$
  - $\mathcal{S}_3 = \{(i,j,k) \in V^3 \mid c_{ij} \neq 0 \text{ or } c_{ik} \neq 0 \text{ or } c_{jk} \neq 0\}$
- Theorem: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_3)$
- Conjecture: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_2)$

Linearization Model (LM) + Cuts

Consider the C_3 cuts

- Consist of the subset of triangle inequalities

$$\mathbf{W}(\mathcal{S}) = \left\{ (\mathbf{x},\mathbf{y}) \;\middle|\; \begin{array}{lllll} x_i & +x_j & +x_k & -y_{i,j} - y_{i,k} - y_{j,k} \leqslant 1, \\ -x_i & & & +y_{i,j} + y_{i,k} - y_{j,k} \leqslant 0, \\ & -x_j & & +y_{i,j} - y_{i,k} + y_{j,k} \leqslant 0, \\ & & -x_k & -y_{i,j} + y_{i,k} + y_{j,k} \leqslant 0, \end{array} \left(\begin{array}{c} 1 \leqslant i < j < k \leqslant n \\ (i,j,k) \in \mathcal{S} \end{array} \right) \right\}.$$

- $\mathcal{S}$ represents the set of triplets (i,j,k) corresponding to the triangle inequalities involving variables x_i, x_j and x_k. Four basic cases are considered:
  - $\mathcal{S}_0 = \{(i,j,k) \in V^3 \,|\, c_{ij}c_{ik}c_{jk} \neq 0\}$
  - $\mathcal{S}_1 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ and } (c_{ik} \neq 0 \text{ or } c_{jk} \neq 0)\}$
  - $\mathcal{S}_2 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0\}$
  - $\mathcal{S}_3 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ or } c_{ik} \neq 0 \text{ or } c_{jk} \neq 0\}$
- Theorem: $C_3 = \mathrm{LM} + \mathbf{W}(\mathcal{S}_3)$
- Conjecture: $C_3 = \mathrm{LM} + \mathbf{W}(\mathcal{S}_2)$

Linearization Model (LM) + Cuts

Consider the C_3 cuts

- Consist of the subset of triangle inequalities

$$\mathbf{W}(\mathcal{S}) = \left\{ (\mathbf{x}, \mathbf{y}) \,\middle|\, \begin{array}{rrrl} x_i & +x_j & +x_k & -y_{i,j} - y_{i,k} - y_{j,k} \leqslant 1, \\ -x_i & & & +y_{i,j} + y_{i,k} - y_{j,k} \leqslant 0, \\ & -x_j & & +y_{i,j} - y_{i,k} + y_{j,k} \leqslant 0, \\ & & -x_k & -y_{i,j} + y_{i,k} + y_{j,k} \leqslant 0, \end{array} \quad \left(\begin{array}{c} 1 \leqslant i < j < k \leqslant n \\ (i,j,k) \in \mathcal{S} \end{array} \right) \right\}.$$

- $\mathcal{S}$ represents the set of triplets (i, j, k) corresponding to the triangle inequalities involving variables x_i, x_j and x_k. Four basic cases are considered:
  - $\mathcal{S}_0 = \{(i,j,k) \in V^3 \,|\, c_{ij} c_{ik} c_{jk} \neq 0\}$
  - $\mathcal{S}_1 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ and } (c_{ik} \neq 0 \text{ or } c_{jk} \neq 0)\}$
  - $\mathcal{S}_2 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0\}$
  - $\mathcal{S}_3 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ or } c_{ik} \neq 0 \text{ or } c_{jk} \neq 0\}$
- Theorem: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_3)$
- Conjecture: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_2)$

Linearization Model (LM) + Cuts

Consider the C_3 cuts

- Consist of the subset of triangle inequalities

$$\mathbf{W}(\mathcal{S}) = \left\{ (\mathbf{x}, \mathbf{y}) \;\middle|\; \begin{array}{rrrrl} x_i & +x_j & +x_k & -y_{i,j} - y_{i,k} - y_{j,k} \leqslant 1, \\ -x_i & & & +y_{i,j} + y_{i,k} - y_{j,k} \leqslant 0, \\ & -x_j & & +y_{i,j} - y_{i,k} + y_{j,k} \leqslant 0, \\ & & -x_k & -y_{i,j} + y_{i,k} + y_{j,k} \leqslant 0, \end{array} \left(\begin{array}{c} 1 \leqslant i < j < k \leqslant n \\ (i,j,k) \in \mathcal{S} \end{array} \right) \right\}.$$

- $\mathcal{S}$ represents the set of triplets (i, j, k) corresponding to the triangle inequalities involving variables x_i, x_j and x_k. Four basic cases are considered:
  - $\mathcal{S}_0 = \{(i,j,k) \in V^3 \,|\, c_{ij} c_{ik} c_{jk} \neq 0\}$
  - $\mathcal{S}_1 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ and } (c_{ik} \neq 0 \text{ or } c_{jk} \neq 0)\}$
  - $\mathcal{S}_2 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0\}$
  - $\mathcal{S}_3 = \{(i,j,k) \in V^3 \,|\, c_{ij} \neq 0 \text{ or } c_{ik} \neq 0 \text{ or } c_{jk} \neq 0\}$
- Theorem: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_3)$
- Conjecture: $C_3 = \text{LM} + \mathbf{W}(\mathcal{S}_2)$

A LP Branch–and–Cut (B&C) model for QUBO

LP-B&C-QUBO($f, \mathcal{S}, \mathcal{P}$)

Input: Let f be a quadratic pseudo–Boolean function f. $\mathcal{S}$ is the set of triplets considered to define the triangle inequalities. $\mathcal{P}$ is the set of 4-tuples considered to define the square inequalities.

Step 1: Find an incumbent $\mathbf{x}^{+}$ for f using the tabu search implementation of Palubeckis '04.

Step 2: Solve the LP

$$z(f, \mathcal{S}, \mathcal{P}) = \min \left\{ L_f(\mathbf{x}, \mathbf{y}) \,\middle|\, (\mathbf{x}, \mathbf{y}) \in \mathbf{W}^{[3]}(\mathcal{S}) \cup \mathbf{W}^{[4]}(\mathcal{P}), \mathbf{x} \in \mathbb{U}^n \right\}.$$

Save the optimal basic feasible solution B.

Step 3: Remove all triangle and square cuts that have zero dual values, i.e. remove those cuts that are non-binding. The resulting problem is a 0–1 MIP.

Step 4: Solve the LP relaxation of the MIP by warm starting it with the basis B. Load the incumbent $\mathbf{x}^{+}$ as a solution of the MIP and then solve it.

Output: The minimum value of f is equal to the optimum of the MIP, and every minimizer $\mathbf{x}^{\star}$ of the MIP is also a minimizer of f.

Application Covered Next

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

The MAX–CUT of the g3-8 torus graph was found

- There are four torus graphs considered in the DIMACS library of mixed semidefinite-quadratic-linear programs
- The torus graphs are 3D-toroidal graphs, originated from the Ising model
- LP-B&C-QUBO($\mathcal{S}_1, \emptyset$) was able to prove optimality for the first time to graph g3-8, which has ± 1 interactions and 512 vertices
- It required 302 156 nodes and 1 871 155 sec to find this proof on a standard computer

2D Ising Models

Found better solutions for 2D Ising models than top meta–heuristics for QUBO

Instance	*Vertices*	LP-B&C-QUBO *with* $\mathcal{S} = \mathcal{S}_1$ *and* $\mathcal{P} = \emptyset$				
				*Computing Time**		
		MAX–CUT	*Nodes*	*Incumbent*	*Relaxation*	*MIP*†
G11	100×8	564	30	8.5 s	1.6 s	12.2 s
G12	50×16	556	39	8.4 s	1.8 s	17.7 s
G13	25×32	582	36	8.5 s	1.8 s	22.7 s
G32	100×20	[1 410,1 412]	83 837	35.2 s	5.3 s	10 000.0 s
G33	80×25	[1 382,1 383]	134 133	35.6 s	6.0 s	10 000.0 s
G34	50×40	[1 384,1 388]	66149	35.2 s	5.9 s	10 000.0 s
G57	100×50	[3 492,3 505]	20 598	111.4 s	21.7 s	10 000.0 s
G62	100×70	[4 862,4 886]	10 109	178.7 s	36.9 s	10 000.0 s
G65	100×80	[5 550,5 581]	4 199	217.4 s	47.1 s	10 000.0 s
G66	90×100	[6 352,6 387]	5 065	258.8 s	159.7 s	10 000.0 s
G67	100×100	[6 932,6 981]	7 683	303.7 s	323.8 s	10 000.0 s

* Computed on an AMD Athlon 64 X2 Dual Core 4800+, 2.41 GHz, 4GB RAM and runs XP.

† The MIP solver stage was set to run at most 10 000 sec.

Application Covered Next

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

Minimum–3–Partition (M3P) of Graphs

M3P

- Given a weighted graph $G = (V, E, \mathbf{w})$, the MkP problem is the problem of partitioning the set of vertices V into k disjoint subsets such that the total weight of the edges joining vertices of the same partition is minimum.
- To solve M3P we use the solver LP-B&C-QUBO($f, \mathcal{S}, \mathcal{Z}$), where $\mathcal{S}$ is $\mathcal{S}_1$ or $\mathcal{S}_2$ and $\mathcal{Z}$ defines the set of pure square cuts

Main reference about the M3P problem

Anjos, M., B. Ghaddar and F. Liers.
A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem.
Research report, Combinatorial Optimization in Physics (COPhy) (July 2007).

Minimum–3–Partition (M3P) of Graphs

M3P

- Given a weighted graph $G = (V, E, \mathbf{w})$, the MkP problem is the problem of partitioning the set of vertices V into k disjoint subsets such that the total weight of the edges joining vertices of the same partition is minimum.
- To solve M3P we use the solver LP-B&C-QUBO($f, \mathcal{S}, \mathcal{Z}$), where $\mathcal{S}$ is $\mathcal{S}_1$ or $\mathcal{S}_2$ and $\mathcal{Z}$ defines the set of pure square cuts

Main reference about the M3P problem

Anjos, M., B. Ghaddar and F. Liers.
A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem.
Research report, Combinatorial Optimization in Physics (COPhy) (July 2007).

Minimum–3–Partition (M3P) of Graphs

M3P

- Given a weighted graph $G = (V, E, \mathbf{w})$, the MkP problem is the problem of partitioning the set of vertices V into k disjoint subsets such that the total weight of the edges joining vertices of the same partition is minimum.
- To solve M3P we use the solver LP-B&C-QUBO$(f, \mathcal{S}, \mathcal{Z})$, where $\mathcal{S}$ is $\mathcal{S}_1$ or $\mathcal{S}_2$ and $\mathcal{Z}$ defines the set of pure square cuts

Main reference about the M3P problem

Anjos, M., B. Ghaddar and F. Liers.
A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem.
Research report, Combinatorial Optimization in Physics (COPhy) (July 2007).

Minimum–3–Partition (M3P) of Graphs

Optimal Minimum–3–Partitions of 2D and 3D Ising models

			SBC		LP with $(\mathcal{S}_1, \mathcal{Z})$		LP with $(\mathcal{S}_2, \mathcal{Z})$	
Instance	*Weights*	*M3P*	*Nodes*	*Time**	*Nodes*	*Time***	*Nodes*	*Time***
4×4	Gaussian	-954 077	1	16 s	1	1.7 s	1	2.1 s
5×5		-1 484 348	2	23 s	5	2.7 s	13	5.3 s
6×6		-2 865 560	1	312 s	1	4.4 s	9	10.4 s
7×7		-3 282 435	1	3 128 s	9	8.2 s	13	20.9 s
8×8		-5 935 339	1	8 503 s	27	12.7 s	45	43.9 s
4×4	±1	-13	1	$<$ 0.005 s	1	1.8 s	1	2.4 s
5×5		-20	1	4 s	28	4.4 s	14	5.6 s
6×6		-29	1	22 s	107	7.5 s	68	10.8 s
7×7		-40	1	112 s	277	13.8 s	170	25.8 s
8×8		-55	1	1 598 s	243	22.6 s	330	50.1 s
9×9		-64	1	27 349 s	50 175	1 116.5 s	25 794	1 256.4 s
2 × 3 × 4	±1	-20	1	3 s	8	5.6 s	8	6.9 s
2 × 4 × 4		-28	4	234 s	522	19.1 s	592	25.4 s
3 × 3 × 3		-26	1	11 s	20	8.0 s	53	11.9 s
3 × 3 × 4		-36	1	50 s	453	30.0 s	1 222	60.5 s
3 × 4 × 4		-48	1	719 s	17 499	862.9 s	15 629	639.7 s
3 × 4 × 5		-63	16	32 133 s	13 123	1 126.5 s	32 709	2 657.1 s
4 × 4 × 4		-65	19	30 975 s	171 846	15 247.2 s	136 671	11 157.3 s

* Sun Sparc 1200 MHz.

** Computed on an AMD Athlon 64 X2 Dual Core 4800+, 2.41 GHz, 4GB RAM and runs XP.

Application Covered Next

Graph Models

- MAX–CUT
- MAX–Clique
- MIN–VC
- Graph Coloring
- Graph Partitioning
- Graph Balancing
- MIN–3–Partition

Engineering and Social Sciences

- MAX–SAT
- Via Minimization
- VLSI design
- 2D and 3D Ising Model
- 1D Ising Chain
- Fault Diagnosis
- Hierarchical Clustering
- Vision
- Preventing DDoS attacks
- Finding Highly Connected Proteins
- Combinatorics of Real World Graphs

QUBOs derived from Vision problems

QUBO's derived from Vision problems

- Preprocessing could fix about 15% of the variables within 1 sec
- Branch-and-Cut can solve the entire problem in about 10 sec

THANK YOU