Separation of Rank 1 Chvátal-Gomory Inequalities

Brady Hunsaker[1] Craig Tovey[2]

[1]University of Pittsburgh

[2]Georgia Institute of Technology

5 January 2005

Chvátal-Gomory inequalities

- Valid inequalities for pure IPs formed by using known valid inequalities.

Chvátal-Gomory inequalities

- Valid inequalities for pure IPs formed by using known valid inequalities.
- Technique explicitly presented by Chvátal (1973); implicit in earlier work by Gomory (1958).

Chvátal-Gomory inequalities

- Valid inequalities for pure IPs formed by using known valid inequalities.
- Technique explicitly presented by Chvátal (1973); implicit in earlier work by Gomory (1958).

1. Known valid inequalities: $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, $i = 1, \ldots, m$, and $x_j \geq 0, j = 1, \ldots, n$.

Chvátal-Gomory inequalities

- Valid inequalities for pure IPs formed by using known valid inequalities.
- Technique explicitly presented by Chvátal (1973); implicit in earlier work by Gomory (1958).

1. Known valid inequalities: $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$, $i = 1, \ldots, m$, and $x_j \geq 0, j = 1, \ldots, n$.
2. Select nonnegative multipliers $u_1, \ldots, u_m$.

Chvátal-Gomory inequalities

- Valid inequalities for pure IPs formed by using known valid inequalities.
- Technique explicitly presented by Chvátal (1973); implicit in earlier work by Gomory (1958).

1. Known valid inequalities: $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, $i = 1, \ldots, m$, and $x_j \geq 0, j = 1, \ldots, n$.
2. Select nonnegative multipliers $u_1, \ldots, u_m$.
3. The resulting Chvátal-Gomory inequality is

$$\sum_{j=1}^{n} \left\lfloor \sum_{i=1}^{m} u_i a_{ij} \right\rfloor x_j \leq \left\lfloor \sum_{i=1}^{m} u_i b_i \right\rfloor .$$

Chvátal-Gomory rank

Use CG inequalities and original inequalities to generate new CG inequalities.

- The CG rank of the initial inequalities and any inequality dominated by a nonnegative linear combination of the initial inequalities is 0.

Chvátal-Gomory rank

Use CG inequalities and original inequalities to generate new CG inequalities.

- The CG rank of the initial inequalities and any inequality dominated by a nonnegative linear combination of the initial inequalities is 0.
- Any inequality which can be generated as a CG inequality based on rank 0 inequalities—but is not rank 0 itself—has CG rank 1.

Chvátal-Gomory rank

Use CG inequalities and original inequalities to generate new CG inequalities.

- The CG rank of the initial inequalities and any inequality dominated by a nonnegative linear combination of the initial inequalities is 0.
- Any inequality which can be generated as a CG inequality based on rank 0 inequalities—but is not rank 0 itself—has CG rank 1.
- A valid inequality has CG rank k if it can be generated as a CG inequality based on inequalities of rank $0, 1, \ldots, k-1$, but does not have rank less than k.

Motivation: Determining CG rank

Given an IP instance and a valid inequality, what is its Chvátal-Gomory rank?

- This question is often considered when new valid inequalities are proposed for a problem.

Motivation: Determining CG rank

Given an IP instance and a valid inequality, what is its Chvátal-Gomory rank?

- This question is often considered when new valid inequalities are proposed for a problem.
- Determining an upper bound on the rank is often done by trial-and-error.

Motivation: Determining CG rank

Given an IP instance and a valid inequality, what is its Chvátal-Gomory rank?

- This question is often considered when new valid inequalities are proposed for a problem.
- Determining an upper bound on the rank is often done by trial-and-error.
- Lower bound proofs are generally even harder.

Motivation: Determining CG rank

Given an IP instance and a valid inequality, what is its Chvátal-Gomory rank?

- This question is often considered when new valid inequalities are proposed for a problem.
- Determining an upper bound on the rank is often done by trial-and-error.
- Lower bound proofs are generally even harder.
- Not aware of any existing computational method for testing rank 2.

Motivation: Identifying useful CG inequalities

Given a family of IP instances, which low rank CG inequalities are most useful?

- Can we identify structured classes of inequalities for a family of polyhedra?

Motivation: Identifying useful CG inequalities

Given a family of IP instances, which low rank CG inequalities are most useful?

- Can we identify structured classes of inequalities for a family of polyhedra?
- How much tighter is the rank 1 polyhedron than the LP relaxation?

Motivation: Identifying useful CG inequalities

Given a family of IP instances, which low rank CG inequalities are most useful?

- Can we identify structured classes of inequalities for a family of polyhedra?
- How much tighter is the rank 1 polyhedron than the LP relaxation?
- Separating rank 1 CG inequalities is NP-hard (Eisenbrand 1999), so our goals are to learn more about polyhedra, not to develop direct CG techniques for general-purpose solvers.

An LP to identify Rank 1 inequalities

- Candidate inequality: $\sum_{j=1}^{n} f_j x_j \leq g$ with $f_j \in \mathbf{Z}, g \in \mathbf{Z}$.

An LP to identify Rank 1 inequalities

- Candidate inequality: $\sum_{j=1}^{n} f_j x_j \leq g$ with $f_j \in \mathbf{Z}, g \in \mathbf{Z}$.
- If rank 1 CG inequality, then there exist nonnegative u_i such that $\sum_{i=1}^{m} u_i a_{ij} \geq f_j$ for all j and such that $\sum_{i=1}^{m} u_i b_i < g + 1$.

An LP to identify Rank 1 inequalities

- Candidate inequality: $\sum_{j=1}^{n} f_j x_j \leq g$ with $f_j \in \mathbf{Z}, g \in \mathbf{Z}$.
- If rank 1 CG inequality, then there exist nonnegative u_i such that $\sum_{i=1}^{m} u_i a_{ij} \geq f_j$ for all j and such that $\sum_{i=1}^{m} u_i b_i < g + 1$.
- LP for "best" multipliers:

$$\begin{array}{lrcll} \min & \sum_{i=1}^{m} b_i u_i & & & \\ s.t. & \sum_{i=1}^{m} u_i a_{ij} & \geq & f_j & j = 1, \ldots, n \\ & u_i & \geq & 0 & i = 1, \ldots, m \end{array}$$

An LP to identify Rank 1 inequalities

- Candidate inequality: $\sum_{j=1}^{n} f_j x_j \leq g$ with $f_j \in \mathbf{Z}, g \in \mathbf{Z}$.
- If rank 1 CG inequality, then there exist nonnegative u_i such that $\sum_{i=1}^{m} u_i a_{ij} \geq f_j$ for all j and such that $\sum_{i=1}^{m} u_i b_i < g + 1$.
- LP for "best" multipliers:

$$\begin{array}{lrcll} \min & \sum_{i=1}^{m} b_i u_i & & & \\ s.t. & \sum_{i=1}^{m} u_i a_{ij} & \geq & f_j & j = 1, \ldots, n \\ & u_i & \geq & 0 & i = 1, \ldots, m \end{array}$$

- If the optimal objective value satisfies $z \leq g$, then it is rank 0. If $g < z < g + 1$, then the inequality is rank 1. Otherwise it is rank > 1 or invalid.

Column generation for identifying Rank 2 inequalities

- Can form a similar LP for rank 2, but need a multiplier variable for each CG inequality of rank 1.

Column generation for identifying Rank 2 inequalities

- Can form a similar LP for rank 2, but need a multiplier variable for each CG inequality of rank 1.
- Column generation is necessary to identify a rank 1 inequality $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$. π is the vector of dual prices for the base LP.

$$
\begin{array}{rrcll}
\max & \sum_{j=1}^{n} \pi_j \alpha_j - \beta & & & \\
s.t. & \sum_{i=1}^{m} u_i a_{ij} & = & \hat{\alpha}_j & j = 1, \ldots, n \\
 & \sum_{i=1}^{m} u_i b_i & = & \hat{\beta} & \\
 & \hat{\alpha}_j & \geq & \alpha_j & j = 1, \ldots, n \\
 & \lfloor \hat{\beta} \rfloor & \leq & \beta & \\
 & u_i & \geq & 0 & i = 1, \ldots, m \\
 & \alpha_j, \beta & \text{integer} & &
\end{array}
$$

Column generation for identifying Rank 2 inequalities

- Can form a similar LP for rank 2, but need a multiplier variable for each CG inequality of rank 1.
- Column generation is necessary to identify a rank 1 inequality $\sum_{j=1}^{n} \alpha_j x_j \leq \beta$. π is the vector of dual prices for the base LP.

$$
\begin{array}{rrcll}
\max & \sum_{j=1}^{n} \pi_j \alpha_j - \beta & & & \\
s.t. & \sum_{i=1}^{m} u_i a_{ij} & = & \hat{\alpha}_j & j = 1, \ldots, n \\
 & \sum_{i=1}^{m} u_i b_i & = & \hat{\beta} & \\
 & \hat{\alpha}_j & \geq & \alpha_j & j = 1, \ldots, n \\
 & \lfloor \hat{\beta} \rfloor & \leq & \beta & \\
 & u_i & \geq & 0 & i = 1, \ldots, m \\
 & \alpha_j, \beta & \text{integer} & &
\end{array}
$$

- But this is not a linear IP.

Column generation

- $\lfloor \hat{\beta} \rfloor \leq \beta$ is the same as $\hat{\beta} - 1 < \beta$
- Replace that with $\hat{\beta} - 1 + \epsilon \leq \beta$, where ϵ is small

$$\begin{array}{rrcll}
\max & \sum_{j=1}^{n} \pi_j \alpha_j - \beta & & & \\
s.t. & \sum_{i=1}^{m} u_i a_{ij} & = & \hat{\alpha}_j & j = 1, \dots, n \\
 & \sum_{i=1}^{m} u_i b_i & = & \hat{\beta} & \\
 & \hat{\alpha}_j & \geq & \alpha_j & j = 1, \dots, n \\
 & \hat{\beta} - 1 + \epsilon & \leq & \beta & \\
 & u_i & \geq & 0 & i = 1, \dots, m \\
 & \alpha_j, \beta & \text{integer} & &
\end{array}$$

Separation of Rank 1 inequalities

- Reinterpret π as a point in $\mathbf{R}^n$. Then we may view the column generation as separation.
- This allows us to separate over rank 1 inequalities.

$$
\begin{array}{rrcll}
\max & \sum_{j=1}^{n} \pi_j \alpha_j - \beta & & & \\
s.t. & \sum_{i=1}^{m} u_i a_{ij} & = & \hat{\alpha}_j & j = 1, \ldots, n \\
 & \sum_{i=1}^{m} u_i b_i & = & \hat{\beta} & \\
 & \hat{\alpha}_j & \geq & \alpha_j & j = 1, \ldots, n \\
 & \hat{\beta} - 1 + \epsilon & \leq & \beta & \\
 & u_i & \geq & 0 & i = 1, \ldots, m \\
 & \alpha_j, \beta & \text{integer} & &
\end{array}
$$

Succinct certificates of rank

Rank	Certificate
≤ 1	
≥ 2	
≤ 2	

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	
≤ 2	

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	Point x that satisfies rank 0 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g+1$
≤ 2	

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	Point x that satisfies rank 0 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g + 1$
≤ 2	Multipliers for rank 0 and rank 1 inequalities; also certificates of rank ≤ 1 for each rank 1 inequality used

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	Point x that satisfies rank 0 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g + 1$
≤ 2	Multipliers for rank 0 and rank 1 inequalities; also certificates of rank ≤ 1 for each rank 1 inequality used

- Each of the certificates above is succinct.

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	Point x that satisfies rank 0 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g + 1$
≤ 2	Multipliers for rank 0 and rank 1 inequalities; also certificates of rank ≤ 1 for each rank 1 inequality used

- Each of the certificates above is succinct.
- Unfortunately, the "certificate" for rank ≥ 3 has to prove an IP infeasibility.

Succinct certificates of rank

Rank	Certificate
≤ 1	Multipliers for rank 0 inequalities
≥ 2	Point x that satisfies rank 0 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g + 1$
≤ 2	Multipliers for rank 0 and rank 1 inequalities; also certificates of rank ≤ 1 for each rank 1 inequality used

- Each of the certificates above is succinct.
- Unfortunately, the "certificate" for rank ≥ 3 has to prove an IP infeasibility.
- This certificate is a point x that satisfies rank 1 inequalities and $\sum_{j=1}^{n} f_j x_j \geq g + 1$. It is provided for insight, even though it cannot be checked quickly.

Tolerances & bounds are important

Potential numerical trouble due to the interaction of ϵ, the particular instance, and the tolerances of the solver.

- Although the epsilon was written as an absolute epsilon, it is really a relative epsilon.

Tolerances & bounds are important

Potential numerical trouble due to the interaction of ϵ, the particular instance, and the tolerances of the solver.

- Although the epsilon was written as an absolute epsilon, it is really a relative epsilon.
- May need to try several values of ϵ.

Tolerances & bounds are important

Potential numerical trouble due to the interaction of ϵ, the particular instance, and the tolerances of the solver.

- Although the epsilon was written as an absolute epsilon, it is really a relative epsilon.
- May need to try several values of ϵ.
- Need upper bounds on the multipliers u_i: no multiplier needs to be greater than 1.

Tolerances & bounds are important

Potential numerical trouble due to the interaction of ϵ, the particular instance, and the tolerances of the solver.

- Although the epsilon was written as an absolute epsilon, it is really a relative epsilon.
- May need to try several values of ϵ.
- Need upper bounds on the multipliers u_i: no multiplier needs to be greater than 1.
- The separation formulation is still weak, and future work is necessary to improve it. The rest of this talk focuses on potential uses of this approach.

Input may be AMPL or MPS

Two inputs are necessary:

IP instance
: This defines the rank 0 constraints. Two possible formats:
- GNU MathProg (subset of AMPL)
- MPS

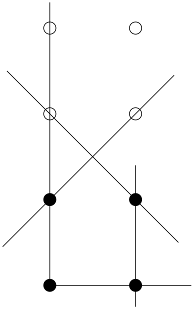
Constraints to test
: Specified in a text file containing a vector of constraint coefficients on each line.

Output is text, including certificates of rank.

Example: High rank

Textbook example:

$$
\begin{aligned}
tx_1 + x_2 &\leq 1 + t \\
-tx_1 + x_2 &\leq 1 \\
x_1 &\leq 1 \\
x_1, x_2 &\geq 0
\end{aligned}
$$



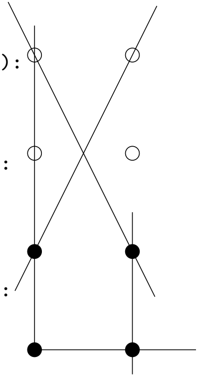
Valid inequality: $x_2 \leq 1$ has rank t.

High rank: $t = 1$ output

```
Coin0001I At line 7 NAME          classic1
Coin0001I At line 8 ROWS
Coin0001I At line 13 COLUMNS
Coin0001I At line 17 RHS
Coin0001I At line 20 ENDATA
Coin0002I Problem classic1 has 3 rows, 2 columns and 5 elements
Coin0008I classic1 read with 0 errors
CG-rank: ==================================================
CG-rank: Testing constraint 2 1  <= 1
Clp0006I 0  Obj 0 Primal inf 1 (1)
Clp0006I 2  Obj 1.5
Clp0000I Optimal - objective value 1.5
CG-rank: Rank of inequality is 1
CG-rank: Multipliers to achieve rank 1
CG-rank: (multipliers followed by constraint in sparse format):
CG-rank: 0.5: 1 1  2 1
CG-rank: 0.5: 1 -1  2 1
```

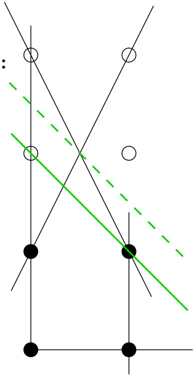
High rank: $t = 2$ output excerpts

```
Testing constraint 2 1  <= 1
Rank of inequality is at least 2
The following point satisfies all rank 0 inequalities
and certifies rank 2 or greater (sparse vector format):
1 0.5 2 2
Adding rank 1 row: 1 1  2 1  <= 2
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.874925007499: 1 2  2 1
0.374925007499: 1 -2  2 1
Rank 2
Multipliers to achieve rank 2:
(multipliers followed by constraint in sparse format):
0.333333333333: 1 -2  2 1
0.666666666667: 1 1  2 1
```



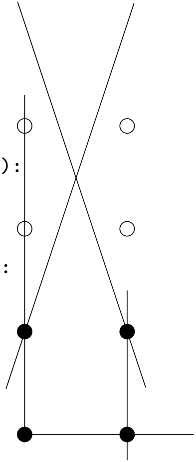
High rank: $t = 2$ output excerpts

```
Testing constraint 2 1  <= 1
Rank of inequality is at least 2
The following point satisfies all rank 0 inequalities
and certifies rank 2 or greater (sparse vector format):
1 0.5 2 2
Adding rank 1 row: 1 1  2 1  <= 2
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.874925007499: 1 2  2 1
0.374925007499: 1 -2  2 1
Rank 2
Multipliers to achieve rank 2:
(multipliers followed by constraint in sparse format):
0.333333333333: 1 -2  2 1
0.666666666667: 1 1  2 1
```



High rank: $t = 3$ output excerpts

```
Testing constraint 2 1  <= 1
Rank of inequality is at least 2
The following point satisfies all rank 0 inequalities
and certifies rank 2 or greater (sparse vector format):
1 0.5 2 2.5
Adding rank 1 row: 2 1  <= 2
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.599940005999: 1 3  2 1
0.599940005999: 1 -3  2 1
Rank 3 or more
```



Example: Knapsack

$$\begin{aligned} \sum_{i=1}^{12} i x_i &\leq 12 \\ x_i &\geq 0 \end{aligned}$$

Facet Coefficients	RHS	CG Rank
0 0 0 0 0 1 2 2 2 2 2 2	2	
0 0 0 0 1 1 1 2 2 2 2 2	2	
0 0 0 2 2 3 4 4 6 6 6 6	6	
0 0 1 1 2 2 2 3 3 4 4 4	4	
0 0 1 2 3 3 3 4 5 6 6 6	6	
0 0 3 4 5 6 7 8 9 12 12 12	12	
0 1 1 2 2 3 4 4 5 5 6 6	6	
0 1 3 4 4 6 8 8 9 11 12 12	12	

Example: Knapsack

$$\begin{aligned} \sum_{i=1}^{12} i x_i &\leq 12 \\ x_i &\geq 0 \end{aligned}$$

Facet Coefficients	RHS	CG Rank
0 0 0 0 0 1 2 2 2 2 2 2	2	2
0 0 0 0 1 1 1 2 2 2 2 2	2	2
0 0 0 2 2 3 4 4 6 6 6 6	6	3+
0 0 1 1 2 2 2 3 3 4 4 4	4	1
0 0 1 2 3 3 3 4 5 6 6 6	6	2
0 0 3 4 5 6 7 8 9 12 12 12	12	3+
0 1 1 2 2 3 4 4 5 5 6 6	6	1
0 1 3 4 4 6 8 8 9 11 12 12	12	3+

Examples of structured inequalities with rank 1

To test the potential of discovering inequalities, we applied the rank 1 optimizer to several well-studied problems:

- Matching:
- Node packing:
- TSP:

Examples of structured inequalities with rank 1

To test the potential of discovering inequalities, we applied the rank 1 optimizer to several well-studied problems:

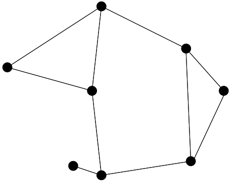
- Matching: odd-set inequalities
- Node packing: odd-hole inequalities
- TSP: comb inequalities

Matching

The maximum matching problem on a graph $G = (V, E)$ is

$$\begin{array}{llll} \max & c^T x & & \\ s.t. & \sum_{e \in \delta(v)} x_e & \leq & 1 & \forall v \in V \\ & x_e & \in & \{0,1\} & \forall e \in E, \end{array}$$

where $\delta(v)$ denotes all edges incident with v. The standard LP relaxation replaces the final constraints above with $x_e \geq 0$.

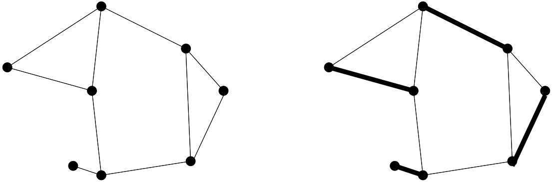


Matching

The maximum matching problem on a graph $G = (V, E)$ is

$$\begin{array}{llll} \max & c^T x & & \\ s.t. & \sum_{e \in \delta(v)} x_e \leq & 1 & \forall v \in V \\ & x_e \in & \{0,1\} & \forall e \in E, \end{array}$$

where $\delta(v)$ denotes all edges incident with v. The standard LP relaxation replaces the final constraints above with $x_e \geq 0$.



Separating rank 1 matching constraints

We tested matching on random unweighted graphs of size from 6 nodes to 12 nodes. For each size, 20 graphs were created with each density $p = 1/n$, $p = 2/n$, and $p = 3/n$.
Sample output of generated inequalities:

```
Adding rank 1 row: 4 1  5 1  6 1  8 1  14 1  <= 2
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.5: 4 1  5 1  <= 1
0.5: 1 1  6 1  7 1  8 1  <= 1
0.5: 4 1  6 1  9 1  <= 1
0.5: 5 1  7 1  13 1  14 1  <= 1
0.5: 8 1  9 1  12 1  14 1  <= 1
0.5: 1 -1  <= 0
1: 7 -1  <= 0
1: 9 -1  <= 0
0.5: 12 -1  <= 0
0.5: 13 -1  <= 0
```

Separating rank 1 matching constraints

We tested matching on random unweighted graphs of size from 6 nodes to 12 nodes. For each size, 20 graphs were created with each density $p = 1/n$, $p = 2/n$, and $p = 3/n$.
Sample output of generated inequalities:

```
Adding rank 1 row: 4 1  5 1  6 1  8 1  14 1  <= 2
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.5: 4 1  5 1  <= 1
0.5: 1 1  6 1  7 1  8 1  <= 1
0.5: 4 1  6 1  9 1  <= 1
0.5: 5 1  7 1  13 1  14 1  <= 1
0.5: 8 1  9 1  12 1  14 1  <= 1
```

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.
- At least 400 were odd-set inequalities or dominated by odd-set inequalities:

$$\sum_{e \in \gamma(S)} x_e \leq \frac{|S| - 1}{2}.$$

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.
- At least 400 were odd-set inequalities or dominated by odd-set inequalities:

$$\sum_{e \in \gamma(S)} x_e \leq \frac{|S| - 1}{2}.$$

- All possible sizes of odd sets were found.

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.
- At least 400 were odd-set inequalities or dominated by odd-set inequalities:
$$\sum_{e \in \gamma(S)} x_e \leq \frac{|S| - 1}{2}.$$
- All possible sizes of odd sets were found.
- Every problem was solved to integer optimality.

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.
- At least 400 were odd-set inequalities or dominated by odd-set inequalities:

$$\sum_{e \in \gamma(S)} x_e \leq \frac{|S| - 1}{2}.$$

- All possible sizes of odd sets were found.
- Every problem was solved to integer optimality.
- Adding odd-set inequalities to the LP relaxation forms the integer hull for matching (Edmonds 1965), and odd-set inequalities have CG rank 1.

Separating rank 1 matching constraints

- 408 rank 1 inequalities were generated.
- At least 400 were odd-set inequalities or dominated by odd-set inequalities:

$$\sum_{e \in \gamma(S)} x_e \leq \frac{|S| - 1}{2}.$$

- All possible sizes of odd sets were found.
- Every problem was solved to integer optimality.
- Adding odd-set inequalities to the LP relaxation forms the integer hull for matching (Edmonds 1965), and odd-set inequalities have CG rank 1.
- It would probably be easy to recognize these patterns, including the proof of CG rank 1, from the output.

Matching exceptional derivations

There were a few odd derivations, such as this odd-set constraint:

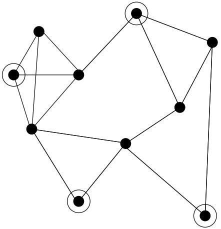
```
Adding rank 1 row: 1 1  3 1  4 1  5 1  6 1  8 1  9 1  13 1  14 1  <= 3
Multipliers to create rank 1 row
(multipliers followed by constraint in sparse format):
0.666533346665: 1 1  2 1  3 1  <= 1
0.666533346665: 4 1  5 1  <= 1
0.333466653335: 1 1  6 1  7 1  8 1  <= 1
0.666533346665: 4 1  6 1  9 1  <= 1
0.333466653335: 5 1  7 1  13 1  14 1  <= 1
0.666533346665: 3 1  11 1  13 1  <= 1
0.666533346665: 8 1  9 1  12 1  14 1  <= 1
```

(Nonnegativity multipliers were also present.)

Node packing

Given a graph $G = (V, E)$, *unweighted node packing* is the problem

$$\begin{array}{lllll} \max & \sum_{v \in V} x_v & & & \\ s.t. & x_u + x_v & \leq & 1 & \forall \{u, v\} \in E \\ & x_v & \in & \{0, 1\} & \forall v \in V. \end{array}$$



Node packing results

- 284 rank 1 inequalities were generated.

Node packing results

- 284 rank 1 inequalities were generated.
- All are odd-hole inequalities:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

Node packing results

- 284 rank 1 inequalities were generated.
- All are odd-hole inequalities:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

- 250 of size 3, 33 of size 5, 1 of size 7.

Node packing results

- 284 rank 1 inequalities were generated.
- All are odd-hole inequalities:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

- 250 of size 3, 33 of size 5, 1 of size 7.
- The rank 1 optimal solution was usually integral but not always:

Node packing results

- 284 rank 1 inequalities were generated.
- All are odd-hole inequalities:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

- 250 of size 3, 33 of size 5, 1 of size 7.
- The rank 1 optimal solution was usually integral but not always:
  - 279 of the 420 initial LPs had integer solutions

Node packing results

- 284 rank 1 inequalities were generated.
- All are odd-hole inequalities:

$$\sum_{v \in H} x_v \leq \frac{|H| - 1}{2}.$$

- 250 of size 3, 33 of size 5, 1 of size 7.
- The rank 1 optimal solution was usually integral but not always:
  - 279 of the 420 initial LPs had integer solutions
  - 411 of the 420 rank 1 polyhedra had integer solutions

TSP

Given a graph $G = (V, E)$:

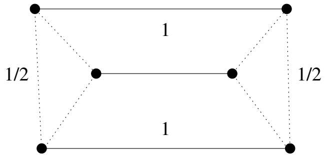
$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
s.t. & \sum_{e \in \delta(v)} x_e & = & 2 & \forall v \in V \\
 & \sum_{e \in \delta(S)} x_e & \geq & 2 & \forall S \subset V, S \neq V, S \neq \emptyset \\
 & x_e & \in & \{0,1\} & \forall e \in E.
\end{array}
$$

TSP

Given a graph $G = (V, E)$:

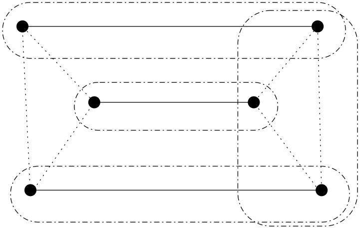
$$
\begin{array}{llll}
\min & \sum_{e \in E} c_e x_e & & \\
s.t. & \sum_{e \in \delta(v)} x_e & = & 2 & \forall v \in V \\
& \sum_{e \in \delta(S)} x_e & \geq & 2 & \forall S \subset V, S \neq V, S \neq \emptyset \\
& x_e & \in & \{0,1\} & \forall e \in E.
\end{array}
$$

The linear relaxation may have fractional solutions such as the following:



TSP Comb Inequalities

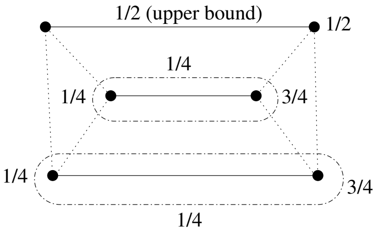
The solution on the previous slide violates a comb inequality:



The sum of the edge variables in the comb can be no more than 4. Comb inequalities are rank 1 CG inequalities.

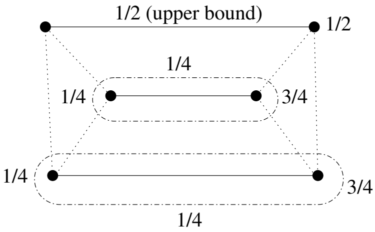
Comb Derivation

The previous comb inequality is identified, but the derivation is not in a typical form:



Comb Derivation

The previous comb inequality is identified, but the derivation is not in a typical form:



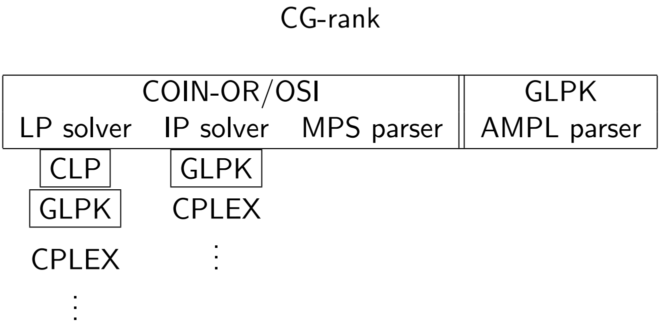
The usual derivation of comb inequalities uses the "inside form" of the subset constraints and may explain part of the difference.

Open-source tools used

CG-rank

LP solver	IP solver	MPS parser	AMPL parser

Open-source tools used



Availability and license

- Source code for rank determination available at http://www.engr.pitt.edu/hunsaker/cg-rank/
- Licensed under an **open-source license**: your choice of CPL or GNU GPL
- Also need to download COIN-OR/OSI and GLPK in order to compile

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known

Algorithm and implementation improvements:

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum

Algorithm and implementation improvements:

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum
- Look at the benefit of adding rank 1 inequalities

Algorithm and implementation improvements:

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum
- Look at the benefit of adding rank 1 inequalities

Algorithm and implementation improvements:

- Improve column generation to allow for larger instances

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum
- Look at the benefit of adding rank 1 inequalities

Algorithm and implementation improvements:

- Improve column generation to allow for larger instances
- Allow input to be specified in a greater variety of formats: MPS, GNU MathProg, CPLEX LP

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum
- Look at the benefit of adding rank 1 inequalities

Algorithm and implementation improvements:

- Improve column generation to allow for larger instances
- Allow input to be specified in a greater variety of formats: MPS, GNU MathProg, CPLEX LP
- Make linking with GLPK optional

Future work

Uses:

- Look for useful rank 1 inequalities on problems of interest where the answer is not known
- Look at the ranks of inequalities that are tight at the optimum
- Look at the benefit of adding rank 1 inequalities

Algorithm and implementation improvements:

- Improve column generation to allow for larger instances
- Allow input to be specified in a greater variety of formats: MPS, GNU MathProg, CPLEX LP
- Make linking with GLPK optional
- Extend to rank 2 separation and rank 3 identification

Future work: Improving the formulation

$$
\begin{array}{lrcll}
\max & \sum_{j=1}^{n} \pi_j \alpha_j - \beta & & & \\
s.t. & \sum_{i=1}^{m} u_i a_{ij} & = & \hat{\alpha}_j & j = 1, \ldots, n \\
 & \sum_{i=1}^{m} u_i b_i & = & \hat{\beta} & \\
 & \hat{\alpha}_j & \geq & \alpha_j & j = 1, \ldots, n \\
 & \hat{\beta} - 1 + \epsilon & \leq & \beta & \\
 & u_i & \geq & 0 & i = 1, \ldots, m \\
 & \alpha_j, \beta & \text{integer} & &
\end{array}
$$