Computational Integer Programming
Universidad de los Andes

Lecture 6

Dr. Ted Ralphs

Reading for This Lecture

- Nemhauser and Wolsey Sections II.1.1-II.1.3, II.1.6
- Wolsey Chapter 8
- Valid Inequalities for Mixed Integer Linear Programs, G. Cornuejols (2006)

Describing $\mathbf{conv}(S)$

- As before, we consider a pure integer program

$$\begin{aligned} z_{IP} &= \max\{cx \mid x \in S\}, \\ S &= \{x \in \mathbb{Z}^n_+ \mid Ax \leq b\}. \end{aligned}$$

- Under our assumptions, $\text{conv}(S)$ is a rational polyhedron.
- Thus, in theory, it is possible to generate a complete description of it.
- So why aren't IPs easy to solve?
  - The number of inequalities required is generally HUGE!
  - The number of facets of the TSP polytope for an instance with 120 nodes is more than 10^{100} times the number of atoms in the universe.
  - It is physically impossible to write down a description of this polytope.
  - Not only that, but it is very difficult in general to generate these facets (this problem is not in $\mathcal{P}$ in general).

Improving Bounds

- Our discussions of branch and bound have so far focused on the use of three basic bounding methods.
  - LP relaxation
  - Lagrangian relaxation
  - Dantzig-Wolfe decomposition
- Recall that the bound produced by Lagrangian relaxation and Dantzig-Wolfe decomposition is

$$z_D = \max\{cx \mid A^1x \leq b^1, x \in \mathrm{conv}(S_{LR})\},$$

which is an improvement over that produced by solving the LP relaxation.

- Producing the bound z_D depends on our ability to efficiently optimize over $\mathrm{conv}(S_{LR})$.
- Can we improve the LP relaxation in some way?

Cutting Planes

- Recall that the inequality denoted by (π, π_0) is *valid* for a polyhedron $\mathcal{P}$ if $\pi x \leq \pi_0 \; \forall x \in \mathcal{P}$.
- The term *cutting plane* usually refers to an inequality valid for $\mathrm{conv}(S)$, but which is violated by the solution obtained by solving the (current) LP relaxation.
- Note that this is not a very precise definition and the term is a bit colloquial, but we will use it anyway.
- *Cutting plane methods* attempt to improve the bound produced by the LP relaxation by iteratively adding cutting planes to the initial LP relaxation.
- Adding such inequalities to the LP relaxation *may* improve the bound (this is not a guarantee).

The Separation Problem

- Methods for generating cutting planes dynamically attempt to solve a *separation problem*.
- The separation problem can itself be formulated as an optimization problem in a number of ways.
- Most commonly, we wish to generate the valid inequality that is *most violated*.
- This problem is equivalent (in a complexity sense) to the optimization problem over the same convex set. optimization and separation, we could
- Hence, we could in principle use a cutting plane method as a third alternative to produce the bound z_D.

Methods for Generating Cutting Planes

- In most cases, the separation problems that arise cannot be solved exactly, so we either
  - solve the separation problem heuristically, or
  - solve the separation problem exactly, but for a relaxation.
- The *template paradigm* for separation consists of restricting the class of inequalities considered to just those with a specific form.
- This is equivalent, in some sense, to solving the separation problem for a relaxation.
- Separation algorithms can generally be divided into two classes
  - Algorithms that do not assume any specific structure.
  - Algorithms that only work in the presence of specific structure.

Generating Cutting Planes: Two Viewpoints

- There are a number of different points of view from which one can derive the standard methods used to generate cutting planes for general MILPs.
- As we have seen before, there is an *algebraic* point of view and a *geometric* point of view.
- Algebraic:
  - Take combinations of the known valid inequalities.
  - Use rounding to produce stronger ones.
- Geometric:
  - Use a disjunction (as in branching) to generate several disjoint polyhedra whose union contains S.
  - Generate inequalities valid for the convex hull of this union.
- Although these seem like very different points of view, they turn out to be roughly equivalent.

Generating Valid Inequalities: Algebraic Viewpoint

- Consider the feasible region of the LP relaxation $\mathcal{P} = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$.
- Valid inequalities for $\mathcal{P}$ can be obtained by taking nonnegative linear combinations of the rows of (A, b).
- Except for one pathological case[1], all valid inequalities for $\mathcal{P}$ are either equivalent to or dominated by an inequality of the form

$$uAx \leq ub, u \in \mathbb{R}^m_+.$$

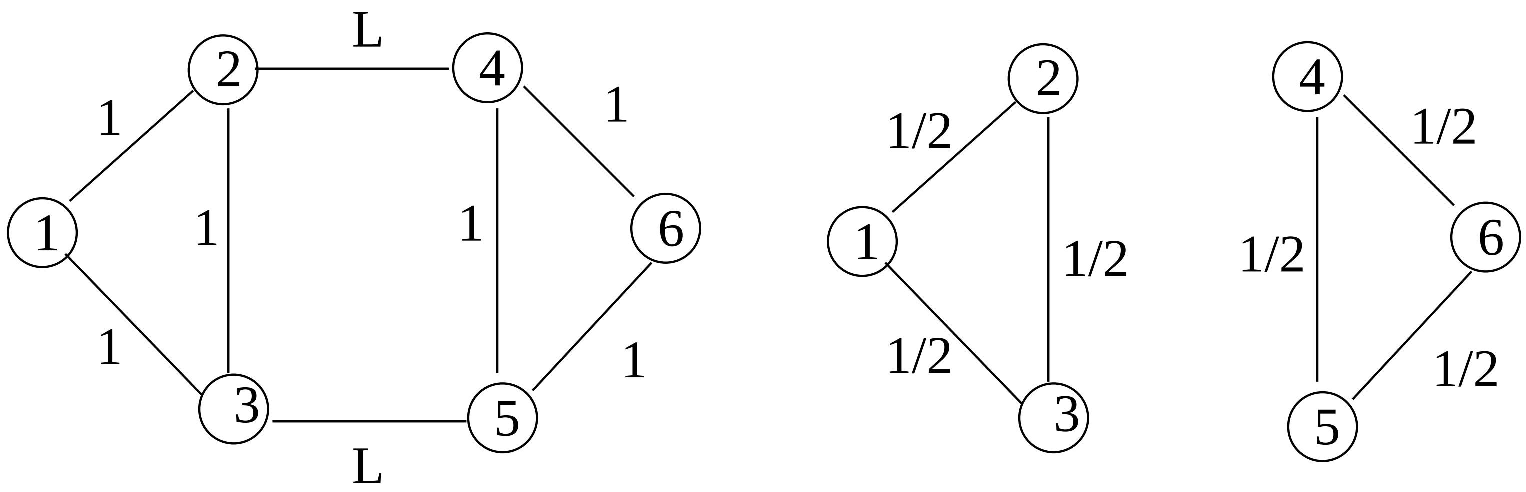
- To avoid the pathological case, we may assume that A contains explicit upper bounds on the variables.

[1] The pathological case occurs when one or more variables have no explicit upper bound *and* both the

Generating Valid Inequalities for $\mathbf{conv}(S)$

- All inequalities valid for $\mathcal{P}$ are also valid for $\text{conv}(S)$, but they are not cutting planes.
- We can do better.
- We need the following simple principle: if $a \leq b$ and a is an integer, then $a \leq \lfloor b \rfloor$.
- Believe it or not, this simple fact is all we need to generate all valid inequalities for $\text{conv}(S)$!

The Perfect Matching Problem



Consider the perfect matching problem.

$$\min \sum_{e=\{i,j\}\in E} c_e x_e$$

$$s.t. \sum_{\{j|\{i,j\}\in E\}} x_{ij} = 1, \ \forall i \in N$$

$$x_e \in \{0,1\}, \qquad \forall e = \{i,j\} \in E.$$

The Odd Cut Inequalities

- Each odd cutset induces a possible valid inequality.

$$\sum_{e \in \delta(S)} x_e \geq 1, S \subset N, |S| \text{ odd.}$$

- Let's derive these another way.
  - Consider an odd set of nodes U.
  - Sum the constraints $\sum_{\{j | \{i,j\} \in E\}} x_{ij} = 1$ for $i \in U$.
  - Relaxing to inequality, we get $2\sum_{e \in E(U)} x_e + \sum_{e \in \delta(u)} x_e \leq |U|$.
  - Dividing through by 2, we obtain $\sum_{e \in E(U)} x_e + \frac{1}{2}\sum_{e \in \delta(u)} x_e \leq \frac{1}{2}|U|$.
  - We can drop the second term of the sum to obtain

$$\sum_{e \in E(U)} x_e \leq \frac{1}{2}|U|.$$

  - What's the last step?

The Chvátal-Gomory Procedure

- Let $A = (a_1, a_2, \ldots, a_n)$ and $N = \{1, \ldots, n\}$.
  1. Choose a weight vector $u \geq 0$.
  2. Obtain the valid inequality $\sum_{j \in N}(ua_j)x \leq ub$.
  3. Round the coefficients down to obtain $\sum_{j \in N}(\lfloor ua_j \rfloor)x \leq ub$. Why can we do this?
  4. Finally, round the right hand side down to obtain the valid inequality

$$\sum_{j \in N}(\lfloor ua_j \rfloor)x \leq \lfloor ub \rfloor$$

- This procedure is called the *Chvátal-Gomory* rounding procedure, or simply the *C-G procedure*.
- Surprisingly, any inequality valid for $\operatorname{conv}(S)$ can be produced by a finite number of iterations of this procedure!

Assessing the Procedure

- Although it is theoretically possible to generate any valid inequality using the C-G procedure, it is far from ideal.
- Depending on the weights chosen, we may not even obtain a supporting hyperplane.
- This is is because we can only push the inequality in until it meets some point in $\mathbb{Z}^n$, which may or may not also be in S.
- In fact, the procedure may not even generate a hyperplane that includes an integer point!
- The coefficients of the generated inequality must be relatively prime to ensure the generated hyperplane includes an integer point.

Proposition 1. *Let* $S = \{x \in \mathbb{Z}^n \mid \sum_{j \in N} a_j x_j \leq b\}$, *where* $a_j \in \mathbb{Z}$ *for* $j \in N$, *and let* $k = gcd\{a_1, \ldots, a_n\}$. *Then* $conv(S) = \{x \in \mathbb{R}^n \mid \sum_{j \in N}(a_j/k)x_j \leq \lfloor b/k \rfloor\}$.

Generating All Valid Inequalities

- Any valid inequality that can be obtained through iterative application of the C-G procedure is a *C-G inequality*.
- For pure integer programs, all valid inequalities are C-G inequalities.

Theorem 1. *Let $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ be a valid inequality for $S = \{x \in \mathbb{Z}^n_+ \mid Ax \leq b\} \neq \emptyset$. Then (π, π_0) is a C-G inequality for S.*

- The number of applications of the C-G procedure necessary to obtain a given valid inequality is called its *C-G rank*, denoted $r(\pi, \pi_0)$.
- The *C-G rank* of a polyhedron is the number of applications of the C-G procedure necessary to obtain $\text{conv}(S)$.
- The rank of a polyhedron, denoted $\rho(\mathcal{P})$, is equal to the maximum of the ranks of its facets.
- For pure integer programs, the rank is always finite.

The Gomory Cut

- Let's consider S, the set of solutions to an IP with one equation:

$$S = \left\{ x \in \mathbb{Z}^n_+ \mid \sum_{j=1}^{n} a_j x_j = a_0 \right\}$$

- For each j, let $f_j = a_j - \lfloor a_j \rfloor$. Then equivalently

$$S = \left\{ x \in \mathbb{Z}^n_+ \mid \sum_{j=1}^{n} f_j x_j = f_0 + k \text{ for some integer k} \right\}$$

- Since $\sum_{j=1}^{n} f_j x_j \geq 0$ and $f_0 < 1$, then $k \geq 0$ and so

$$\sum_{j=1}^{n} f_j x_j \geq f_0$$

is a valid inequality for S called a *Gomory cut*.

The Gomory Cut (cont)

- The importance of Gomory cutting planes is that they can be derived from the tableau while solving an LP relaxation.
- Consider the set $S = \{x \in \mathbb{Z}_+^{n+m} \mid (A, I)x = b\}$ where A has integral coefficients.
- Derive a new valid equation by combining the equations in the representation with weight vector λ to obtain

$$\sum_{j=1}^{n} (\lambda A_j) x_j + \sum_{i=1}^{m} \lambda_i x_{n+i} = \lambda b,$$

where A_j is the j^{th} column of A.

- Applying the previous procedure, we can obtain the valid inequality

$$\sum_{j=1}^{n} (\lambda A_j - \lfloor \lambda A_j \rfloor) x_k + \sum_{i=1}^{m} (\lambda_i - \lfloor \lambda_i \rfloor) x_{n+i} \geq \bar{b} - \lfloor \bar{b} \rfloor.$$

- Note that this is really just a C-G inequality with weights $u_i = \lambda_i - \lfloor \lambda_i \rfloor$.

Deriving Valid Inequalities from the Tableau

- Note that each row of the tableau is a nonnegative linear combination of the original equations.
- Suppose we choose a row in which the value of the basic variable is not an integer.
- Applying the procedure from the last slide, the resulting inequality will only involve nonbasic variables and will be of the form

$$\sum_{j \in NB} f_j x_j \geq f_0$$

where $0 \leq f_j < 1$ and $0 < f_0 < 1$.

- We can conclude that the generated inequality will be violated by the current LP solution.
- Under mild assumptions on the algorithm used to solve the LP, this yields a finite algorithm for solving pure integer programs.
- However, its convergence can be very slow.

Valid Inequalities from Disjunctions

- Valid inequalities for $\mathrm{conv}(S)$ can also be generated based on disjunctions.
- In fact, in some sense, all valid inequalities arise from some sort of logical disjunction.
- In this way, branch and cutting are two different methods of exploiting a given disjunction.
- We will not have time to delve into the details of the tradeoffs between the two, but it is a topic of current research.
- Let $\mathcal{P}_i = \{x \in \mathbb{R}^n_+ \mid A^i x \leq b^i\}$ for $i = 1, \ldots, k$ be such that $S \subseteq \cup_{i=1}^k \mathcal{P}_i$.
- Then inequalities valid for $\cup_{i=1}^k \mathcal{P}_i$ are also valid for $\mathrm{conv}(S)$.

Valid Inequalities for the Union of Polyhedra

Valid inequalities based on disjunctions can be derived from the following straightforward result:

Proposition 2. *If* $\sum_{j=1}^{n} \pi_j^1 \leq \pi_0^1$ *is valid for* $S_1 \subseteq \mathbb{R}_+^n$ *and* $\sum_{j=1}^{n} \pi_j^2 \leq \pi_0^2$ *is valid for* $S_2 \subseteq \mathbb{R}_+^n$*, then*

$$\sum_{j=1}^{n} \min(\pi_j^1, \pi_j^2)x \leq \max(\pi_0^1, \pi_0^1)$$

for $x \in S_1 \cup S_2$.

In fact, all valid inequalities for the union of two polyhedra can be obtained in this way.

Proposition 3. *If* $\mathcal{P}^i = \{x \in \mathbb{R}_+^n \mid A^i x \leq b^i\}$ *for* $i = 1, 2$ *are nonempty polyhedra, then* (π, π_0) *is a valid inequality for* $conv(\mathcal{P}^1 \cup \mathcal{P}^2)$ *if and only if there exist* $u^1, u^2 \in \mathbb{R}^m$ *such* $\pi \leq u^i A^i$ *and* $\pi_0 \geq u^i b^i$ *for* $i = 1, 2$.

Strengthening Gomory Cuts Using Disjunction

- Consider again the set of solutions to an IP with one equation.
- This time, we write S equivalently as

$$S = \left\{ x \in \mathbb{Z}^n_+ \mid \sum_{j:f_j \leq f_0} f_j x_j + \sum_{j:f_j > f_0} (f_j - 1) x_j = f_0 + k \text{ for some integer k} \right\}$$

- Since $k \leq -1$ or $k \geq 0$, we have the disjunction

$$\sum_{j:f_j \leq f_0} \frac{f_j}{f_0} x_j - \sum_{j:f_j > f_0} \frac{(1 - f_j)}{f_0} x_j \geq 1$$

OR

$$- \sum_{j:f_j \leq f_0} \frac{f_j}{(1 - f_0)} x_j + \sum_{j:f_j > f_0} \frac{(1 - f_j)}{(1 - f_0)} x_j \geq 1$$

The Gomory Mixed Integer Cut

- Applying Proposition 2, we get

$$\sum_{j:f_j \le f_0} \frac{f_j}{f_0} x_j + \sum_{j:f_j > f_0} \frac{(1-f_j)}{(1-f_0)} x_j \ge 1$$

- This is called a *Gomory mixed integer* (GMI) cut.
- GMI cuts dominate the associated Gomory cut in general and can also be obtained easily from the tableau.
- In the case of the mixed integer set

$$S = \left\{ x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p} \mid \sum_{j=1}^{p} a_j x_j + \sum_{j=p+1}^{n} g_j x_j = a_0 \right\},$$

the GMI cut is

$$\sum_{j:f_j \le f_0} \frac{f_j}{f_0} x_j + \sum_{j:f_j > f_0} \frac{(1-f_j)}{(1-f_0)} x_j + \sum_{j:g_j > 0} \frac{g_j}{f_0} x_j - \sum_{j:g_j < 0} \frac{g_j}{(1-f_0)} x_j \ge 1$$

Example

Consider the following two variable IP.

$$
\begin{aligned}
\min\ & 20000x_1 + 15000x_2 \\
\text{s.t.}\ & 0.3x_1 + 0.4x_2 \geq 2.0 \\
& 0.4x_1 + 0.2x_2 \geq 1.5 \\
& 0.2x_1 + 0.3x_2 \geq 0.5 \\
& 0 \leq x_1 \leq 9 \\
& 0 \leq x_2 \leq 6 \\
& x_1, x_2 \in \mathbb{Z}
\end{aligned}
$$

The optimal solution to the LP relaxation is $(2, 3.5)$.

Example (cont.)

- The two rows of the optimal tableau corresponding to the solution $(2, 3.5)$ that correspond to binding constraints are

$$x_1 - 4s_1 + 2s_2 = 2.0x_2 + 3s_1 - 4s_2 = 3.5 \tag{1}$$

- Note that these rows are combinations of the rows corresponding to the two binding constraints from the formulation (in standard form).
- The GMI cut resulting from row 2 is

$$6s_1 + 8s_2 \geq 1$$

- In terms of the original variables, this is

$$12x_1 + 11x_2 \geq 65$$

- This is violated by the solution $(2, 3.5)$.

Lift and Project

- Let's now consider $S = \mathcal{P} \cap \mathbb{B}^n$ and assume that the inequalities $x \leq 1$ are included among those in $Ax \leq b$.
- Note that $conv(S) \subseteq conv(\mathcal{P}_j^0 \cup \mathcal{P}_j^1)$ where $\mathcal{P}_j^0 = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j = 0\}$ and $\mathcal{P}_j^1 = \mathcal{P} \cap \{x \in \mathbb{R}^n \mid x_j = 1\}$ for some $j \in \{1, \ldots, n\}$.
- Applying Proposition 3, we see that the inequality (π, π_0) is valid for $\mathcal{P}_j = conv(\mathcal{P}_j^0 \cup \mathcal{P}_j^1)$ if there exists $u^i \in \mathbb{R}_+^m$, and $v^i \in \mathbb{R}_+$ for $i = 0, 1$ such that

$$\begin{aligned} \pi &\leq u^0 A + v^0 e_j, \\ \pi &\leq u^1 A - v^1 e_j, \\ \pi^0 &\geq u^0 b, \\ \pi^0 &\geq u^1 b - v^1, \end{aligned}$$

- Notice that this is a set of linear constraints, i.e., we could write a linear program to generate constraints based on this disjunction.

The Cut Generating LP

- This leads to the cut generating LP (CGLP), which generates the most violated inequality valid for $\mathcal{P}_j$.

$$\begin{array}{rrcl}
\text{min} & \pi\hat{x} - \pi^0 & & \\
\text{subject to} & \pi & \leq & u^0 A + v^0 e_j, \\
& \pi & \leq & u^1 A - v^1 e_j, \\
& \pi^0 & \geq & u^0 b, \\
& \pi^0 & \geq & u^1 b - v^1, \\
& \sum_{i=1}^{m} u_i^0 + v^0 + \sum_{i=1}^{m} u_i^1 + v^1 & = & 1 \\
& u^0, u^1, v^0, v^1 & \geq & 0
\end{array}$$

- The last constraint is just for normalization.
- This shows that the separation problem for $\mathcal{P}_j$ is polynomially solvable.

Gomory Cuts vs. Lift-and-Project Cuts

- Note that all Gomory cuts are lift-and-project cuts.
- In fact, there is a direct correspondence between basic feasible solutions of the CGLP and basic (possibly infeasible) solutions of the usual LP relaxation.
- By pivoting in the LP relaxation, we can implicitly solve the cut generating LP (see Balas and Perregaard).
- Thus, the procedure for generating lift-and-project cuts is almost exactly the same as that for generating Gomory cuts.

Valid Inequalities for the Traveling Salesman Problem

- Consider a complete graph $G = (V, E)$.
- A *tour* in this graph is a cycle containing all nodes, i.e., a set of edges inducing a connected subgraph where the degree of every node is 2.
- Let S be the set of all incidence vectors of tours.
- Let $T \supset S$ be defined by

$$T = \{x \in \mathbb{B}^n \mid x \leq x' \text{ for some } x' \in S\}$$

- We are interested in T because $\text{conv}(T)$ is full-dimensional and therefore easier to analyze.
- The dimension of $\text{conv}(S)$, on the other hand, is $|E| - |V|$ (proving this is nontrivial).
- All inequalities valid for T are also valid for S.

Trivial Inequalities of the TSP Polytope

- It is easy to show that the upper and lower bound constraints are facets of $\text{conv}(T)$.
- In fact, they are also facets of $\text{conv}(S)$ for all graphs with $|V| \geq 5$.
- The degree constraints $\sum_{e \in \delta(\{v\})} x_e = 2$ are valid for $\text{conv}(S)$.
- The inequalities $\sum_{e \in \delta(\{v\})} x_e \leq 2$ are facets of $\text{conv}(T)$.
- How do we separate these inequalities?

The Subtour Elimination Constraints

- The constraints $\sum_{e \in E(W)} x_e \leq |W| - 1$ are called the *subtour elimination constraints*.
- These constraints eliminate integer solutions with cycles that do not include all of the nodes.
- The subtour elimination constraints are facet-defining for $\text{conv}(S)$ if $m \geq 4$ for all W with $2 \leq |W| \leq \lfloor m/2 \rfloor$.
- How can we formulate the problem of generating a most violated subtour elimination constraints with respect to $\hat{x} \in \mathbb{R}^n$?

The 2-matching Inequalities

- Even for small examples, the set of inequalities we have discussed so far do not describe the convex hull of integer solutions.
- Let H be any subset of the nodes with $3 \le |H| \le |V| - 1$.
- Let $\hat{E} \subset (H, V \setminus H)$ be an odd set of disjoint edges crossing the cut defined by H.
- By combining the degree constraints for the nodes in H and the nonnegativity constraints for the edges in $\hat{E}$, we get the *2-matching inequalities*.

$$\sum_{e \in E(H)} x_e + \sum_{e \in \hat{E}} x_e \le |H| + \left\lfloor \frac{|\hat{E}|}{2} \right\rfloor .$$

- These are similar to the odd set inequalities for the perfect matching problem.
- Combining these inequalities with the degree constraints yields a complete description of the matching polytope.

Generalizing the 2-matching Inequalities

- The 2-matching inequalities can be restated as

$$\sum_{e \in E(H)} x_e + \sum_{i=1}^{k} \sum_{e \in E(W_i)} x_e \leq |H| + \sum_{i=1}^{k} (|W_i| - 1) - \frac{k+1}{2}.$$

- To get a 2-matching inequality, we can simply take the sets W_i to be the endpoints of the edges in $\hat{E}$.
- This inequality remains valid even if the sets W_i contain more than two points.
- Each set must contain at least one node in H and one node note in H and the sets must all be disjoint.
- These inequalities are called the *comb inequalities* and are also rank 1 C-G inequalities.
- The sets W_i are called the *teeth* and the set H is called the *handle*.

Higher Rank C-G Inequalities

- We can further generalize the comb inequalities by constructing combs whose teeth are themselves combs.
- These *generalized comb inequalities* are obtained by combining the degree constraints, nonnegativity constraints, subtour elimination constraints, and comb inequalities.
- In fact, the generalized comb inequalities turn out to be facet-defining for $conv(S)$.
- By allowing the vertices of the comb to be cliques, we get the facet-defining *clique-tree inequalities*.
- Additional known classes of facet-defining inequalities.
  - Path Inequalities
  - Wheelbarrows
  - Bicycles
  - Ladders
  - Crowns

More Inequalities

- The inequalities we have discussed so far are still not enough to define the convex hull of solutions.
- There are small graphs for which these inequalities are not enough.
- Because the TSP is $\mathcal{NP}$-hard, it is unlikely that the TSP polytope has bounded rank, so it is likely that many more facets exist.
- Computationally, knowledge of just this set of inequalities has been enough to solve very large examples, however.
- The largest TSP solved to date is 24978 cities.
- This is an integer program with on the order of half a billion variables.
- Of course, it took *85 years* (yes, years!) of CPU time to solve ;).

Separation Procedures

- An *exact separation procedure* for a class of inequalities is an algorithm that is guaranteed to return an inequality of that class violated by a given point if one exists.
- A *heuristic separation procedure* is a procedure that may or may not return a violated inequality of a given class.
- The subtour elimination constraints and the 2-matching inequalities are the only classes for which we have polynomial time exact separation procedures.
- However, powerful heuristics are known for many classes.
- These heuristics can take a long time to run.