ILP Formulation Properties and Strenghtening Techniques

VERSION 2009.2

Andreas M. Chwatal

Algorithms and Data Structures Group
Institute of Computer Graphics and Algorithms
Vienna University of Technology

VU Fortgeschrittene Algorithmen und Datenstrukturen
Mai 2009

What is a strong/good (I)LP formulation?

Linear program:
- low number of variables
- low number of constraints
- complexity grows polynomially in these entities

Integer Linear Program:
- Let $\mathcal{F} = \{x_1, \dots, x_k\}$ be an feasible integer solution to some ILP
- Convex hull

$$\mathrm{conv}(\mathcal{F}) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \,\middle|\, \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0 \right\}$$

- Let P denote the polyhedron associated to the linear relaxation of the ILP:

$$\mathrm{conv}(\mathcal{F}) \subseteq P$$

- Optimal case: $\mathrm{conv}(\mathcal{F}) = P$, but this is often not achievable
- In a strong formulation P closely approximates $\mathrm{conv}(\mathcal{F})$

The pigeonhole

Suppose, we want to place $n+1$ items into n holes, such that each hole contains exactly one item. (This is clearly infeasible)

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, n+1 \tag{1a}$$

$$x_{ij} \in \{0,1\} \quad i = 1, \dots, n+1, \ j = 1, \dots, n \tag{1b}$$

Two alternatives for the additional constraints:

$$x_{ij} + x_{kj} \leq 1, \qquad j = 1, \dots, n, \ i \neq k, \ i,k = 1, \dots, n+1 \tag{2}$$

$$\sum_{i=1}^{n+1} x_{ij} \leq 1, \qquad j = 1, \dots, n \tag{3}$$

Linear relaxation of formulation with (2) is feasible (with $x_{ij} = 1/n$), but infeasible with (3).

Example: Facility Location

Given: n potential facility locations with opening costs c_j, m clients with service costs d_{ij} (for client i from facility j)

Formulation 1:

$$\min \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{4a}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = 1 \qquad \text{for all } i \tag{4b}$$

$$x_{ij} \leq y_j \qquad \text{for all } i,j \tag{4c}$$

$$0 \leq x_{ij} \leq 1, \ y_i \in \{0,1\} \qquad \text{for all } i,j \tag{4d}$$

Facility Location (cont.)

Formulation 2:

$$\min \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{5a}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = 1 \qquad \text{for all } i \tag{5b}$$

$$\sum_{i=1}^{m} x_{ij} \leq m \cdot y_j \qquad \text{for all } j \tag{5c}$$

$$0 \leq x_{ij} \leq 1, \ y_i \in \{0,1\} \qquad \text{for all } i,j \tag{5d}$$

Which formulation is better?

- F1 has $n + nm$ constraints, whereas F2 has $n + m$ constraints
- **But,** $P_{F1} \subset P_{F2}$!! (where P_X denotes the polyhedron corresponding to the LP relaxation of formulation X)

Spanning Trees

In this section we study various ST formulations. Why?

Spanning tree (ST) problems arise in various contexts, often as subproblems to more complex problems.

While the MST is solvable in polynomial time, further constraints usually make the problem NP-hard.

Variants:

- Minimum (Weight) Spanning Tree
- Steiner Tree, Price Collecting ST
- {Delay, Resource, Hop, Diameter, ...} - Constrained S.T.
- Minimum Label Spanning Tree
- ...

Notation

In this section we consider various formulations of spanning trees as Integer Linear Programs (ILPs)

- Graph $G = (V, E), n = |V|, m = |E|$
- $S \subseteq V, T \subseteq V$

$$E(S,T) = \{e = \{i,j\} \in E \mid i \in S \wedge j \in T\}$$

- Complement $\bar{S} = V \backslash S$
- Cutset $\delta(S) = E(S, \bar{S})$

Variables

Constants:

- edge-weights $w_e > 0 \ \forall e \in E$

Variables:

- edge-variables $x_e \in \{0,1\}$, indicating if edge e is part of the solution ($x_e = 1$), or not ($x_e = 0$)
- flow variables $f_e \geq 0$, indicating how much flow goes over edge e

ILP-Formulation

$$\min \sum_{e \in E} x_e w_e \tag{6a}$$

$$\text{s.t.} \sum_{e \in E} x_e = n - 1 \tag{6b}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \ \forall S \subset V, S \neq \emptyset \tag{6c}$$

$$x_e \in \{0, 1\} \tag{6d}$$

Subtour elimination constraints: inequalities 6c ensure that no subgraph induced by S contains a cycle.

Subtour elimination formulation

Let P_{sub} denote the polyhedron associated to the linear programming relaxation of formulation (6), i.e.

$$P_{\text{sub}} = \{\vec{x} \in \mathbb{R}^{|E|} \mid \sum_{e \in E} x_e = n - 1, \sum_{e \in E(S)} x_e = |S| - 1 \quad \text{for all } \emptyset \neq S \subset V, \vec{0} \leq \vec{x} \leq \vec{1}\}$$

Theorem

The extreme points of the polyhedron P_{sub} are the $0 - 1$ incidence vectors of spanning trees.

Cycle-elimination formulation

Special case: Let P_{cec} denote the polyhedron resulting from the subtour-formulation, but restricting E(S) with $\emptyset \neq S \subset V$ to cycles.

Theorem

$P_{\text{sub}} \subseteq P_{\text{cec}}$.

Question: Does $P_{\text{sub}} \subset P_{\text{cec}}$ hold?

Cutset formulation

$$P_{\text{cut}} = \{\vec{x} \in \mathbb{R}^{|E|} \mid \sum_{e \in E} x_e = n - 1, \sum_{e \in \delta(S)} x_e \geq 1 \text{ for all } \emptyset \neq S \subset V, \vec{0} \leq \vec{x} \leq \vec{1}\} \tag{7}$$

Theorem

$P_{\text{cut}} \supseteq P_{\text{sub}}$.
P_{cut} *can have fractional extreme points and is thus larger than* P_{sub}.

Proof.
For any set of nodes S we have

$$E = E(S) \cup \delta(S) \cup E(\bar{S})$$

If $\vec{x} \in P_{\text{sub}}$, then $\sum_{e \in E(S)} x_e \leq |S| - 1$ and $\sum_{e \in E(\bar{S})} x_e \leq |\bar{S}| - 1$.
Since

$$\sum_{e \in E} = n - 1$$

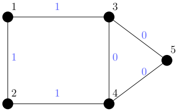
we obtain

$$\sum_{e \in \delta(S)} x_e \geq 1$$

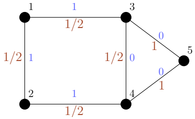
.
Hence, $\vec{x} \in P_{\text{cut}}$ and therefore $P_{\text{cut}} \supseteq P_{\text{sub}}$.

By the example from the next slide we can show that the inclusion may be strict. □

Example of fractional solution of P_{cut}



edge weights w_e



Single Commodity Flow Formulation

$$\min \ \vec{w}^T \vec{x} \tag{8a}$$

$$\text{s.t.} \sum_{e \in \delta^+(1)} f_e - \sum_{e \in \delta^-(1)} f_e = n - 1 \tag{8b}$$

$$\sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 1, \text{ for all } v \neq 1, v \in V \tag{8c}$$

$$f_{ij} \leq (n-1)x_e \text{ for every edge } e = \{i, j\} \tag{8d}$$

$$f_{ji} \leq (n-1)x_e \text{ for every edge } e = \{j, i\} \tag{8e}$$

$$\sum_{e \in E} x_e = n - 1 \tag{8f}$$

$$\vec{f} \geq \vec{0}, 0 \leq x_e \leq 1 \text{ and integral for every edge } e \in E \tag{8g}$$

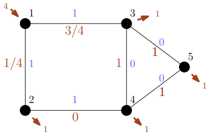
Single Commodity Flow Formulation

$$P_{\text{flow}} = \{\vec{x} \in \mathbb{R}^{|E|} \mid \vec{x}, \vec{f} \text{ satisfy Inequalities 8 without the integrality conditions}\} \tag{9}$$

Remark: The single commodity flow formulation just requires a polynomial number of equations, but the corresponding polyhedron is larger than P_{sub}.

Theorem
$P_{\text{flow}} \supset P_{\text{sub}}$.

Example of solution of P_{flow}



$z^{\text{LP}}_{\text{flow}} = 1 \ \ (z^{ILP}_{\text{opt}} = 2)$

Properties

Theorem

$$P_{\text{sub}} \subseteq P_{\text{cut}} \subseteq P_{\text{flow}} \tag{10}$$

Often directed formulations yield tighter polyhedra.

To derive the directed formulations introduce for each edge $\{i,j\}$ the arcs (i,j) and (j,i) and associate variables y_a to them and set $x_e = y_{ij} + y_{ji}$.
Let P_{dcut} denote the polyhedron of the *directed cut* formulation:

Theorem

$$P_{\text{sub}} = P_{\text{dcut}} \tag{11}$$

Steiner Tree

Steiner Tree (ST) Problem

Given a graph $G = (V,E)$; find a minimum weight subtree of G spanning all terminal nodes $T \subset V$

The subtree might, or might not include some of the other optional "Steiner" nodes $S = V \backslash T$.

Price Collecting Steiner Tree (PCST) Problem

Given a graph $G = (V,E)$ with edge weights w_e for each $e \in E$, root node r. For the other nodes $j \in V\backslash\{r\}$ we have a profit p_j if the tree contains j.

Price Collecting Steiner Tree (cont.)

Let $z_j \in \{0,1\}$ indicate if node j is in the Steiner tree. We can extend the subtour formulation:

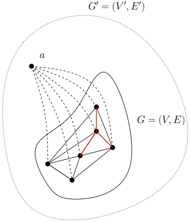
$$\begin{aligned}
\min \ & \sum_{e\in E} w_e x_e - \sum_{i \in V} p_i z_i && && \text{(12a)}\\
\text{s.t.} \ & \sum_{e \in E(U)} x_e \le \sum_{i \in U\backslash\{k\}} z_i && \text{for all } U \subset V \text{ and f.a. } k \in U && \text{(12b)}\\
& \sum_{e\in E} x_e = \sum_{i\in V\backslash\{r\}} z_i && && \text{(12c)}\\
& z_r = 1 && && \text{(12d)}\\
& 0 \le x_e \le 1, 0 \le z_i \le 1 && && \text{(12e)}
\end{aligned}$$

Remark: Inequalities (12b) are called *generalized subtour elimination constraints*
Remark: Of course this formulation does not guarantee $\vec{x}$ and $\vec{z}$ to be integer.

Price Collecting Steiner Tree (cont.)

Determination of the root node:

- artificial root node a
- enhanced graph $G' = (V', E')$
- $V' = V \cup \{a\}$
- $E' = E \cup \{\{a, i\} \mid i \in V\}$
- make sure, that only one of the edges in $E' \backslash E$ is used!



Cycle Elimination Cuts

Cycle-Elimination Cuts:
based on subtour-formulation, but instead of $S \subseteq V$ we only consider cycles $C \subseteq V$.

Cycle elimination cuts can be separated by shortest-path computations

$$\sum_{e \in \delta(C)} x_e \leq |C| - 1 \Leftrightarrow |C| - \sum_{e \in \delta(C)} x_e \geq 1 \Leftrightarrow \sum_{e \in \delta(C)} (1 - x_e) \geq 1$$

Cut Separation by shortest path computations

- consider each edge $e = \{i, j\}$ with $x_e > 0$. (can this be improved?)
- "delete" the edge, or set its weight to some sufficiently large constant
- compute a shortest path from i to j
- if we found a path, and it's value (plus w_{ij}) is less than 1, we can add an inequality enforcing these arcs "not to form a cycle"

Directed Connection Cuts, Cycle Elimination Cuts

- Compute minimum (s, t)-cut by max-flow algorithm
- Given some root node s, this needs to be done for every node $t \in V \backslash \{s\}$ as target node t
- As each node needs to be connected $\sum_{e \in \delta^+(S)} x_e \geq 1, \emptyset \neq S \subset V, s \in S, t \in V \backslash S, s \neq t$
- When sum over such a cut-set $\delta^+(s)$ is lower than one, we have found a cut

Strategies

- Add "most violated" cut vs. add all cuts
- Add cuts "globally", or just for the current node (and it's children) of the branch-and-bound tree
- Various improvements: nested cuts, back-cuts, ...

Remark 1: Considering the edges/nodes in a randomized order might improve the overall process.

Remark 2: For many (tree) problems, the *directed connection cuts* show the best performance in practice, e.g. due to their fast separation.

Dual-Simplex

- **Purpose:** effectively resolve LP after further addition of constraints
- Starting with an dual feasible solution
- Method tries to transform into primal feasible basis while maintaining dual feasibility
- Operations are carried out on primal simplex tableau

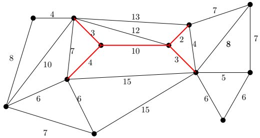
k-node minimum spanning tree (k-MST)

Given:

- (undirected) graph $G = (V, E, w)$
- nonnegative weighting function $w(e) \in \mathbb{R}$

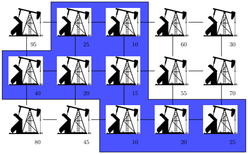
Goal: Find a minimum weight tree spanning exactly k nodes.

Example: $k = 6$, $z = 22$



Applications

Oil-field leasing: Company bought lease for an oil field. After some period, they must return 50% of the oil-field. This part is required to be connected!



Company assigns profit-values to the pumps. Node-weighted $\lceil |V|/2 \rceil$-node MST (blue area) corresponds to the best connected part for the company to return.

Theorem (Complexity of k-MST)

The k-MST problem is NP-hard.

Proof by reduction from Steiner tree.

Programming exercise: develop a (1) flow- and a (2) cycle-elimination or a directed cut formulation for the k-MST problem. Implement the corresponding branch-and-bound and branch-and-cut algorithms!

For more information to the programming exercise see the seperate slides!

- **Valid inequalities**: (linear) inequalities, that are satisfied by all integer points of a formulation
- $\Rightarrow$ strenghtening the formulation
- There exist general techniques to find such inequalities
- Better LP relaxations imply better bounds $\Rightarrow$ speedup of branch-and-bound methods

Chvátal-Gomory Cutting Planes

We consider the following ILP

$$\max \ \mathbf{c}^T\mathbf{x} \tag{13a}$$
$$\text{s.t.} \ \ \mathbf{A}\mathbf{x} \le \mathbf{b} \tag{13b}$$

We can now choose a $\mathbf{u} \in \mathbb{R}^m_+$. All $\mathbf{x}$ satisfying (13) also satisfy

$$\sum_{j=1}^{n} (\mathbf{u}^T\mathbf{A}_j)x_j \le \mathbf{u}^T\mathbf{b} \Rightarrow \sum_{j=1}^{n} \lfloor \mathbf{u}^T\mathbf{A}_j \rfloor x_j \le \mathbf{u}^T\mathbf{b}$$

.. and by using the integrality of $\mathbf{x}$ ..

Chvátal-Gomory Cutting Planes

$$\sum_{j=1}^{n} \lfloor \mathbf{u}^T\mathbf{A}_j \rfloor x_j \le \lfloor \mathbf{u}^T\mathbf{b} \rfloor \tag{14}$$

C-G Example [4]

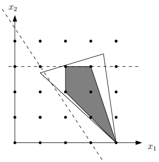
$$\max \ 2x_1 + x_2 \tag{15a}$$
$$\text{s.t.} \ 7x_1 + x_2 \le 28 \tag{15b}$$
$$-x_1 + 3x_2 \le 7 \tag{15c}$$
$$-8x_1 - 9x_2 \le -32 \tag{15d}$$
$$x_1, x_2 \in \mathbb{Z}^+ \cup \{0\} \tag{15e}$$

$u_1 = 0, u_2 = \frac{1}{3}, u_3 = \frac{1}{3}$

$$\Rightarrow -3x_1 - 2x_2 \le -9$$

$u_1 = \frac{1}{21}, u_2 = \frac{7}{22}, u_3 = 0$

$$\Rightarrow x_2 \le 3$$



Gomory Cutting Planes

- Gomory cutting planes are particular manifestations of C-G cutting planes
- Can be directly extracted from simplex tableau

Let $x_0 - \sum_{j=1}^{k} c_j x_j = 0$. Consider the following ILP in its standard form (including slack variables $x_{k+1}, \ldots, x_n$):

$$\max \ \sum_{j=1}^{n} c_j x_j \tag{16a}$$
$$\text{s.t.} \ \sum_{j=0}^{n} a_{ij} x_j = b_i \qquad i = 0, \ldots, m \tag{16b}$$

Gomory Cutting Planes (cont.)

We now introduce a partitioning of the indices, where $\vec{beta} = (\beta_1, \ldots, \beta_n)$ represent the *basic solution* indices, and $\vec{\eta} = (\eta_1, \ldots, \eta_{n-m})$ the *non-basic* indices.
A basic solution is thus given by

$$\vec{x}^* = (x^*_{\beta_1}, \ldots, x^*_{\beta_m}, x^*_{\eta_1} = 0, \ldots, x^*_{\eta_{n-m}} = 0)$$

$x^*_{\beta_1}, \ldots, x^*_{\beta_m}$ is a unique solution to

$$\sum_{j=1}^{m} a_{i\beta_j} x_{\beta_j} = b_i, \quad i = 1, \ldots, m \tag{17}$$

It follows, that $\vec{x}^*_\beta = A_\beta^{-1}\vec{b}$
By this partitioning a solution to 16 is given by

$$\sum_{j=0}^{m} a_{i\beta_j} x_{\beta_j} + \sum_{j=1}^{n-m} a_{i\eta_j} x_{\eta_j} = b_i, \quad i = 1, \ldots, m \tag{18}$$

Gomory Cutting Planes (cont.)

(18) can also be solved in terms of nonbasic variables:

$$x_{\beta_i} + \sum_{j=1}^{n-m} \bar{a}_{\beta_i \eta_j} x_{\eta_j} = x^*_{\beta_i}, \quad i = 0, \ldots, m \tag{19}$$

This is exactly what we have in a row of the Simplex-tableau!!! By rounding down the coefficients we obtain the **Gomory (fractional) cutting planes**:

$$x_{\beta_i} + \sum_{j=1}^{n-m} \lfloor \bar{a}_{\beta_i \eta_j} \rfloor x_{\eta_j} = \lfloor x^*_{\beta_i} \rfloor, \quad i = 0, \ldots, m \tag{20}$$

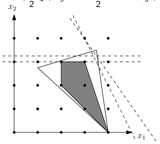
By subtracting (19) from (21) we obtain an equivalent inequality

$$x_{\beta_i} + \sum_{j=1}^{n-m} (\lfloor \bar{a}_{\beta_i \eta_j} \rfloor - \bar{a}_{\beta_i \eta_j}) x_{\eta_j} = \lfloor x^*_{\beta_i} \rfloor - x^*_{\beta_i}, \quad i = 0, \ldots, m \tag{21}$$

Gomory Cutting Planes

Example (continued). We are given the following optimal simplex table:

$$
\begin{array}{llllllll}
x_0 & & +\frac{7}{22}x_3 & +\frac{5}{22}x_4 & =\frac{21}{2} & \Longrightarrow x_0 & \leq 10 & \text{(22a)} \\
& x_1 & +\frac{3}{22}x_3 & -\frac{1}{22}x_4 & =\frac{7}{2} & \Longrightarrow x_1 - x_4 & \leq 3 & \text{(22b)} \\
& & x_2 + \frac{1}{22}x_3 & +\frac{7}{22}x_4 & =\frac{7}{2} & \Longrightarrow x_2 & \leq 3 & \text{(22c)} \\
& & +\frac{3}{2}x_3 & +\frac{5}{2}x_4 + x_5 & =\frac{55}{2} & \Longrightarrow x_3 + 2x_4 + x_5 & \leq 27 & \text{(22d)}
\end{array}
$$



Gomory Cutting Planes

Drawbacks:

- Large LPs $\Rightarrow$ long running times in each iteration
- Many non-zero coefficients $\Rightarrow$ numeric issues

Literature

1. Bertsimas, Weismantel, "Optimization over Integers", Dynamic Ideas, 2005
2. Magnanti, Wolsey, "Optimal Trees", 1994
3. Nemhauser, Wolsey, "Integer and Combinatorial Optimization", Wiley, 1999
4. Lee, "A First Course in Combinatorial Optimization", Cambridge University Press, 2004