ISyE 6662 (Winter 2017)

Discrete Optimization

Prof. N. Boland
Georgia Institute of Technology

LaTeXer: W. Kong
http://wwkong.github.io
Last Revision: May 1, 2017

Table of Contents

These notes are currently a work in progress, and as such may be incomplete or contain errors.

ACKNOWLEDGMENTS:

Special thanks to *Michael Baker* and his LaTeX formatted notes. They were the inspiration for the structure of these notes.

Abstract

The purpose of these notes is to provide the reader with a secondary reference to the material covered in ISyE 6662.

1 Mixed Integer Programs (MIP)

Definition 1.1. The canonical form of the **mixed integer program** (MIP) in this class is

$$\begin{aligned} \max\ & cx + hy \\ \text{s.t. } & Ax + Gy \leq b \\ & x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \end{aligned}$$

The constraint set will be called the **MIP feasible set** $S = \{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{R}^p_+ : Ax + Gy \leq b\}$ while the **value of the MIP** will be denoted by z_{IP}.

Here, c, h, b are vectors of appropriate dimension and A, G are matrices of appropriate dimension which we will call the **data** or **parameters**.

Remark 1.1. Here are the possible outcomes:

- $S = \emptyset$: MIP is infeasible and $z_{IP} = \infty$
- $S \neq \emptyset$:
  - (1) MIP has an optimal solution (i.e. $\exists (x^*, y^*) \in S$ such that $cx^* + hy^* \geq cy + hy, \forall (x, y) \in S$)
  - (2) MIP is unbounded $z_{IP} = \infty$ (i.e. $\forall w \in \mathbb{R}, \exists (x, y) \in S$ such that $cx + hy \geq w$)

Special Cases

- [IP] (Pure) IP: $p = 0$
- [BIP] (Pure Binary) IP: $x_i \in \{0, 1\}, \forall i = 1, 2, ..., n$ [sometimes we write $B = \{0, 1\}$]
- [LP]: $n = 0$

Definition 1.2. The **LP relaxation** of the standard IP is

$$\begin{aligned} z_{LP} = \max\ & cx + hy \\ \text{s.t. } & Ax + Gy \leq b \\ & x \geq 0, y \geq 0 \end{aligned}$$

Note that variable domain constraints are usually retained. That is, $x \in B^n$ in an MIP becomes $0 \leq x_i \leq 1, \forall i = 1, ..., n$ or $0 \leq x \leq 1$ for short.

Remark 1.2. The value of an MIP's LP relaxation gives an upper bound on the MIP's value so $z_{LP} \geq z_{IP}$. The **gap** (integrality gap or LP gap) of an MIP is the "relative error" in this upper bound:

$$\frac{z_{LP} - z_{IP}}{|z_{IP}|}$$

The denominator is often replaced by $\max\{1, |z_{IP}|\}$ to cope with the $z_{IP} = 0$ case.

Definition 1.3. A **combinatorial optimization problem** is defined as follows. Given $\mathcal{G}$, a "ground set", $\mathcal{F} \subseteq 2^{\mathcal{G}}$, a collection of subsets of $\mathcal{G}$, and given, for each $F \in \mathcal{F}$, a value $c(F)$, often expressed as $c(F) = \sum_{i \in F} c_i$ for $(c_i)_{i \in \mathcal{G}}$, we want to solve: $\max\{c(F) : F \in \mathcal{F}\}$.

1.1 Common Problems

Example 1.1. (The [Binary] Knapsack Problem) Given n possible projects where the j^{th} project has cost a_j and has value c_j. Let

$$x_j = \begin{cases} 1 & j^{th} \text{ project is selected} \\ 0 & \text{o/w} \end{cases}$$

The MIP model is

$$\begin{aligned} \max \ & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \ & \sum_{j=1}^{n} a_j x_j \le b \\ & x \in B^n \end{aligned}$$

Example 1.2. ([Weighted] Matching a.k.a. Edge Packing problem) Given an undirected graph (V, E) with V as vertices (nodes) and $E \subseteq \{\{u, v\} \subseteq V : u \neq v\}$ as edges (arcs), and a value c_e for each edge $e \in E$, find a subset of the edges having maximum total value, so that no edges in the set share a vertex. Let

$$x_e = \begin{cases} 1 & \text{edge } e \text{ is selected} \\ 0 & \text{o/w} \end{cases}$$

The MIP is

$$\begin{aligned} \max \ & \sum_{e \in E} c_e x_e \\ \text{s.t.} \ & [C] \end{aligned}$$

where $[C]$ can have two formulations:

(1) $[C]$ is $\forall e_1, e_2 \in E$ such that $e_1 \cap e_2 = \emptyset$, not$(x_{e_1} = 1$ and $x_{e_2} = 1) \equiv x_{e_1} + x_{e_2} \le 2$.

(2) $[C]$ is $\sum_{e \in E, v \in e} x_e \le 1$ for all $v \in V$.

Note that (2) is stronger than (1) [will be shown later]. In general, *less constraints in MIP does not necessarily mean better*.

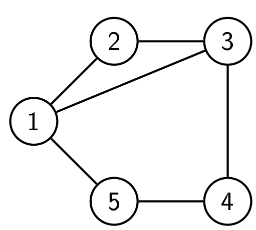
Example 1.3. (Assignment Problem) A special case of weighted matching in where the graph is bipartite (i.e. $\exists W_1, W_2 \subseteq V$, $V = W_1 \cup W_2$ and $W_1 \cap W_2 = \emptyset$ such that $E \subseteq \{\{u, v\} : u \in W_1, v \in W_2\}$). Usually written as

$$\begin{aligned} \max \ & \sum_{e \in E} c_e x_e \\ \text{s.t.} \ & \sum_{e \in \delta(v)} x_e = 1, \forall v \in W_1 \\ & \sum_{e \in \delta(v)} x_e = 1, \forall v \in W_2 \\ & x_e \in \{0, 1\}^{|E|} \end{aligned}$$

{Start Scribe}

Example 1.4. (Node Packing a.k.a. Stable Set) For an undirected graph (V, E), weights w_v for each vertex $v \in V$, we wish to find a subset of the vertices having maximum total weight so that no pair of selected vertices share an edge.

e.g. Consider the graph



The set $\{2, 5\}$ is a maximal (i.e. cannot be enlarged) feasible set (stable set). Let

$$x_v = \begin{cases} 1 & , y \text{ select } v \text{ in stable set} \\ 0 & , \text{otherwise} \end{cases}$$

This problem can be set up as the following MIP:

$$\begin{aligned} \max \ & \sum_{v \in V} w_v x_v \\ \text{s.t. } & x_u + x_v \leq 1 \quad , \forall \{u, v\} \in E \\ & x_v \in \{0, 1\} \qquad , \forall v \in V \end{aligned}$$

In a matrix form, we can define $A = (a_{ev})$ where

$$a_{ev} = \begin{cases} 1 & v \in e \\ 0 & \text{otherwise} \end{cases}$$

for all $v \in V, e \in E$ and equivalently solve the problem $\max\{wx : Ax \leq 1, x \in \{0,1\}^{|V|}\}$.

Example 1.5. (Set Covering / Set Packing / Set Partitioning) We are first given a finite set $\mathcal{M} = \{1, ..., m\}$ and a set of subsets $M_j \subseteq \mathcal{M}$ for $j = 1, 2, ..., n$. We wish to find a particular set $F \subseteq \{1, ..., n\}$ which is a:

- **cover** if $\bigcup_{j \in F} M_j = \mathcal{M}$
- **packing** if $M_i \cap M_j = \emptyset$ for all i, j
- **partition** if it is both a cover and a packing

In addition, given costs c_j (i.e. cost of including j in F) for all $j = 1, ..., n$, and letting

$$x_j = \begin{cases} 1 & \text{, if } j \text{ is selected in } F \\ 0 & \text{, otherwise} \end{cases}$$

the MIP formulation of problems of this type has the form

$$\begin{aligned} & \max cx \\ & \text{s.t.} \sum_{\substack{j=1 \\ k \in M_j}}^{n} x_j \begin{Bmatrix} \geq \\ \leq \\ = \end{Bmatrix} 1 \quad , \forall k \in \mathcal{M} \end{aligned}$$

where we have the following correspondence between the inequalities:

$\geq$ - Covering

$\leq$ - Packing

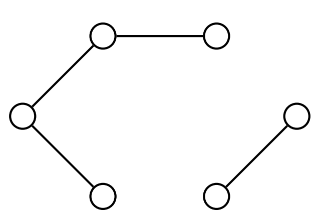
$=$ - Partition

Definition 1.4. Given an undirected graph $G = (V, E)$, the subgraph $G' = (V, E')$ where $E' \subseteq E$ is a **forest** if G' is acyclic.

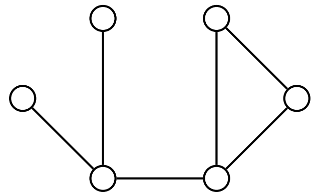
Definition 1.5. A graph is **connected** if there exists a path between each pair of nodes in the graph.

Definition 1.6. A **tree** is a connected forest.

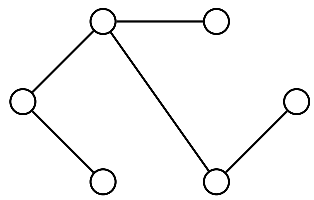
Example 1.6. The following is a forest but is not connected:



The following is an example of a connected graph but is not a forest:



The following is an example of a forest which is connected (i.e. a spanning tree):



Example 1.7. (Max Weighted Forest) Given an undirected graph (V, E) and weights w_e for each $e \in E$, we wish to find a forest of maximum total weight. Let

$$x_e = \begin{cases} 1 & \text{, if } e \in E' \text{ is selected} \\ 0 & \text{, otherwise} \end{cases}$$

The MIP formulation of this problem is

$$\begin{aligned} \max \ & \sum_{e \in E} w_e x_e \\ \text{s.t.} \ & \sum_{e \in E(S)} x_e \leq |S| - 1 \quad && , \forall S \subseteq V \\ & x_e \in \{0, 1\} && , \forall e \in E \end{aligned}$$

where $E(S) = \{e \in E, e \subseteq S\}$. As an exercise, try to find an alternate formulation of the above which does not have an exponential number of constraints (Hint: introduce additional variables).

Example 1.8. (Min Cost Spanning Tree) Given costs c_e for each edge $e \in E$, the MIP formulation is

$$\begin{aligned} \min \ & \sum_{e \in E} c_e x_e \\ \text{s.t.} \ & \sum_{e \in E(S)} x_e \leq |S| - 1 \quad , \forall S \subseteq V \\ & \sum_{e \in E} x_e = |V| - 1 \\ & x \in \{0, 1\}^{|E|} \end{aligned}$$

Another approach to ensuring $(x_e)_{e \in E}$ induces a connected subgraph on all nodes is

$$\sum_{e \in \delta} x_e \geq 1, \forall S \subset V, S \neq \emptyset$$

where $\delta(S) = \{e \in E : |e \cap S| = 1\}$. As an exercise, prove that the first constraint in the MIP formulation and the constraint above can be swapped without changing the feasible region.

1.2 Nonlinear Functions

Example 1.9. (Fixed charge problems) Costs are modeled with a function of the form

$$f(x) = \begin{cases} 0 & \text{, if } x = 0 \\ c + hx & \text{, if } x > 0 \end{cases}$$

where c is our fixed charge. Let

$$y = \begin{cases} 1 & \text{, if } x = 0 \\ 0 & \text{, if } x > 0 \end{cases}$$

Then $f(x) = cy + hx$ provided $0 \le x \le My$ for some known upper limit M on x and $y \in \{0, 1\}$.

Example 1.10. (Bilinear function) Given $f(x, y) = xy, 0 \le x \le M$, and $y \in \{0, 1\}$, let z model xy. This can be done through the constraints $z \le x$, $z \le My$, which model $z \le xy$, and $z \ge x - M(1 - y)$, which models $z \ge xy$.

{End Scribe}

Example 1.11. (*Piecewise Linear Function)* Suppose a piecewise linear function $f(x)$ has **breakpoints** $a_0, a_1, ..., a_p$ with $b_i = f(a_i)$. Observe that for $x \in [a_0, a_p]$, $x = \sum_{k=0}^{p} \alpha_k a_k$ where $\sum_{k=0}^{p} \alpha_k = 1, \alpha_0, \alpha_1, ..., \alpha_p \ge 0$ and $f(x) = \sum_{k=0}^{p} \alpha_k b_k$ <u>provided</u> at most two of the α_k are nonzero and these are consecutive. To model this, let

$$y_k = \begin{cases} 1 & \text{, if } \alpha_i = 0, \forall i \ne k-1, k \\ 0 & \text{, otherwise} \end{cases}$$

for $k = 1, ..., p$. We want to use the constraints

$$\begin{aligned} \alpha_0 &\le y_1 \\ \alpha_1 &\le y_1 + y_2 \\ \alpha_2 &\le y_2 + y_3 \\ &\vdots \\ \alpha_{p-1} &\le y_{p-1} + y_p \\ \alpha_p &\le y_p \\ \sum_{k=1}^{p} y_k &= 1 \\ y &\in \{0, 1\}^p \end{aligned}$$

Example 1.12. (*Fixed Charge Network Flow*) Like min cost network flow, but with a fixed charge. We are first given:

- Digraph (N, A)

- c_a cost/unit flow in arc $a \in A$

- u_a upper bound on flow in arc $a \in A$

- h_a fixed charge for use of arc $a \in A$

- b_i net inflow required at node $i \in N$

Let y_a = units of flow on a. Let

$$x_a = \begin{cases} 1 & \text{, if arc } a \text{ is used (i.e. } y_a > 0) \\ 0 & \text{, otherwise} \end{cases}$$

The IP formulation is

$$\begin{aligned} \min \; & \sum_{a \in A} c_a y_a + \sum_{a \in A} h_a x_a \\ \text{s.t.} \; & \sum_{a \in \delta^-(i)} y_a - \sum_{a \in \delta^+(i)} y_a = b_i && , \forall i \in N \\ & 0 \le y_a \le u_a x_a && , \forall a \in A \end{aligned}$$

Example 1.13. (*Disjunctive Constraints*) In the general case, given m polyhedra

$$P^i := \{x \in \mathbb{R}^n : A^i x \le b^i\}_{i=1,...,m}$$

all bounded, say $P^i \subseteq [0, d^i], \exists d^i \in \mathbb{R}^n$, we want to find a point contained in at least k of them. Let $d_j := \max_{i=1,...,m}$ for all j so $P^i \subseteq [0, d], \forall i$.

Claim. $\exists w$ such that $A^i x \leq b^i + w$ for all $x \in [0, d]$. (Proof left as an exercise)

Let $y_i = 1$ imply $x \in P^i$ with $y \in \{0,1\}^m, \sum_{i=1}^m y_i \geq k$. This can be modeled with the constraints

$$A^i x \leq b^i + w(1 - y_i), \forall i = 1, ..., m$$
$$0 \leq x \leq d$$

1.3 Formulating Models

Example 1.14. Given $n+1$ pigeons and n holes, no pair of pigeons can go in the same hole. Can each pigeon be assigned to a hole? Let

$$x_{ik} = \begin{cases} 1 & \text{, pigeon } i \text{ is assigned hole } k \\ 0 & \text{, otherwise} \end{cases}$$

for all $i = 1, ..., n+1$ and $k = 1, ..., n$. Consider the IP

$$\begin{aligned} z_{IP} = \max\ & 0 \\ \text{s.t.}\ & \sum_{k=1}^{n} x_{ik} = 1 && , i = 1, 2, ..., n+1 \quad (1) \\ & x_{ik} + x_{jk} \leq 1 && , i, j = 1, 2, ..., n+1, i \neq j \quad (2) \\ & x \in \{0,1\}^{n(n+1)} \end{aligned}$$

In the LP relaxation, the last constraint becomes $0 \leq x \leq 1$ and the LP relaxation is feasible! An example of a feasible LP solution is $x_{ik} = 1/n$ for all i, k.

An alternative model: we require the number of pigeons in each hole to be at most 1. In other words,

$$\sum_{i=1}^{n+1} x_{ik} = 1 \text{ for } k = 1, 2..., n \qquad (2')$$

Claim. The LP relaxation of

$$\begin{aligned} z_{IP} = \max\ & 0 \\ \text{s.t.}\ & (1), (2') \\ & x \in \{0,1\}^{n(n+1)} \end{aligned}$$

is infeasible

Proof. Directly, we have

$$\begin{aligned} (2') \implies & \sum_{k=1}^{n} \sum_{i=1}^{n+1} x_{ik} \leq n \\ \implies & \sum_{i=1}^{n+1} \sum_{k=1}^{n} x_{ik} \leq n \\ \overset{(1)}{\implies} & \sum_{i=1}^{n+1} 1 \leq n \\ \implies & n+1 \leq n \end{aligned}$$

□

Thus, this formulation is "stronger"; its LP relaxation is infeasible.

<u>Better Models</u>

Definition 1.7. The same MIP can have many alternative formulations (i.e. different LP relaxations or different constraints). If P_1 and P_2 are polyhedra in $\mathbb{R}^{n+p}$, they are **alternative formulations** of a MIP with feasible set $S \subseteq \mathbb{Z}^n \times \mathbb{R}^p$ if

$$S = P_1 \cap (\mathbb{Z}^n \times \mathbb{R}^p) = P_2 \cap (\mathbb{Z}^n \times \mathbb{R}^p)$$

For MIP $\max\{cx : x \in S\}$, we say P_2 is a better formulation with objective cx if $z_{LP_2} < z_{LP_1}$ where

$$z_{LP_i} = \max\{cx : x \in P_i\}, i = 1, 2$$

If $P_2 \subsetneq P_1$ we say P_2 is a better formulation than . In this case, it is at least as good with respect to every objective, and better for at least one. Note that $x \in P_2 \implies x \in P_1$ implies $P_2 \subseteq P_1$. If also there exists $x \in P_1 \backslash P_2$ then $P_2 \subsetneq P_1$.

Definition 1.8. $P = \text{conv}(S)$ is **ideal formulation** where $\text{conv}(S)$ is the smallest/minimal convex set containing S.

Example 1.15. (*Facility Location Problem*) Given m customers, n possible customers, costs c_j of opening a facility at site j, and costs h_{ij} of serving customer i from a facility at site j, define

$$x_j = \begin{cases} 1, & \text{open facility at site } j \\ 0, & \text{otherwise} \end{cases}, y_{ij} = \text{fraction of } i\text{'s demand met by site } j.$$

We wish to solve the problem

$$\begin{aligned} \min \; & \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} y_{ij} \\ \text{s.t.} \; & \sum_{j=1}^{n} y_{ij} = 1, & i = 1, ..., m \quad (P_1) \\ & \sum_{i=1}^{m} y_{ij} \le m x_j, & j = 1, ..., n \quad (P_1) \\ & x \in \{0,1\}^n, y \in [0,1]^{m\times n}. \end{aligned}$$

Alternatively, this can be reformulated as

$$\begin{aligned} \min \; & \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} y_{ij} \\ \text{s.t.} \; & \sum_{j=1}^{n} y_{ij} = 1, & i = 1, ..., m \quad (P_2) \\ & y_{ij} \le x_j, & i = 1, ..., m, j = 1, ..., n \quad (P_2) \\ & x \in \{0,1\}^n, y \in [0,1]^{m\times n}. \end{aligned}$$

<u>Exercise</u>: Show that $P_2 \subseteq P_1$ and $P_2 \neq P_1$.

<u>Idea</u>: Consider $m = 2, x_1 = \frac{1}{2}$ with

$$P_1 : y_{11} + y_{21} \le 2x_1 = 1$$
$$P_2 : \begin{cases} y_{11} \le x_1 = \frac{1}{2} \\ y_{21} \le x_1 = \frac{1}{2} \end{cases}$$

Graph the feasible regions and compare.

Example 1.16. (*The Traveling Salesman Problem a.k.a. TSP)* Given an undirected graph (V, E), costs c_e of edge $e \in E$, we wish to find a Hamiltonian cycle (a single cycle in the graph containing all vertices in V) of minimum total cost. Let

$$x_e = \begin{cases} 1, & \text{select } e \text{ in cycle (tour)} \\ 0, & \text{otherwise} \end{cases}, \forall e \in E.$$

We can formulate this problem as

$$\begin{aligned} \min \ & \sum_{e \in E} c_e x_e \\ \text{s.t.} \ & \sum_{e \in \delta(v)} x_e = 2 && \forall v \in V \quad (P_1) \\ & \sum_{e \in E(s)} x_e \le |S| - 1, && \forall S \subsetneq V, S \neq V \quad (P_1) \\ & x \in \{0,1\}^{|E|}. \end{aligned}$$

which we call the Dantzig-Fulkerson-Johnson (DFJ) formulation. Alternatively, this can be formulated as

$$\begin{aligned} \min \ & \sum_{e \in E} c_e x_e \\ \textit{s.t.} \ & \sum_{e \in \delta(v)} x_e = 2, && \forall v \in V \quad (P_2) \\ & \sum_{e \in \delta(S)} x_e \ge 2, && \forall S \subsetneq V, S \neq V \quad (P_2) \\ & x \in \{0,1\}^{|E|}. \end{aligned}$$

Exercise: $P_1 = P_2$.

Example 1.17. (*Compact Formulation of the TSP*) Let u_i be the position of city i in the tour. We can use the degree constraints

$$x_{ij} = 1 \implies u_j \ge u_i + 1, \forall i, j \neq 1 \ (P_3)$$

to reformulate the TSP, which is called the Miller, Tucker and Zemlin (MTZ) formulation.

Exercise: Write this implication as a linear constraint.

The u variables and the associated constraints prevent subtours. To see this, suppose that there are subtours in a particular solution. Then pick a set of $\{x_{ij}\}$ where $i, j \neq 1$ and add the inequalities $u_j \ge u_i + 1$ to get

$$\sum_j u_j \ge \sum_i (u_i + 1) \implies 0 \ge \sum_i 1$$

which leads to a contradiction. Note that $u \ge 0$ and $u_i \le n - 1$ for all i. One can show that

$$\text{Proj}_x P_3 = \text{Proj}_{\{1,\dots,|E|\}} \supsetneq P_1 = P_2$$

Definition 1.9. (*Extended Formulations*) Formulations for the same problem often use different variables. Often one formulation uses a subset of the variables of the other; the other is called an **extended formulation**. The set $Q \subseteq \{(x, w) \in \mathbb{R}^n \times \mathbb{R}^p\}$ is an extended formulation for a pure IP with formulation $P \subseteq \mathbb{R}^n$ if

$$(\text{Proj}_{\{1,\dots,n\}} Q) \cap \mathbb{Z}^n = P \cap \mathbb{Z}^n$$

where $\text{Proj}_{\{1,\dots,n\}} Q = \{x : (x, w) \in Q, \exists w \in \mathbb{R}^p\}$. If $\text{Proj}_{\{1,\dots,n\}} Q \subsetneq P$ we say that Q is a better formulation.

Example 1.18. (*Uncapacitated Lot-Sizing*) Given T periods in the planning horizon, initial inventory amount s_0, demand d_t in period t, production cost p_t per unit in period t, fixed charge c_t in period t, and cost of holding a unit h_t in period t, define

$$\begin{aligned} x_t &= \text{number of units made in period } t \\ s_t &= \text{number of units in stoct at the end of period } t \\ y_t &= \begin{cases} 1, & \text{if } x_t > 0 \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

We wish to solve the problem

$$\min \sum_{t=1}^{T}(p_t x_t + c_t y_t + h_t s_t)$$
$$\text{s.t. } (x,y,s) \in P, y \in \{0,1\}^T$$

where

$$\begin{aligned}
P = \{(x,y,s) :& \\
& s_t = s_{t-1} + x_t - d_t, && (1)\\
& 0 \le x_t \le M y_t, && (2)\\
& 0 \le y_t \le 1, && (3)\\
& s_t \ge 0, \text{ [no backlogging]} && (4)\\
& t = 1, ..., T, \\
& y_t \in \mathbb{Z}_t\}
\end{aligned}$$

Logically, we should produce at most the demand of the current period and the remaining periods in a year. This gives us a logical value of $M = \sum_{i=t}^{T} d_i$.

Extended Formulation

Let w_{it} be the quantity made in period i to meet demand in period t for $i \le t$. Then $x_t = \sum_{i=t}^{T} w_{it}$. Now define

$$\begin{aligned}
Q = \{(x,y,s,w) :& \\
& x_t = \sum_{i=t}^{T} w_{it} \\
& s_t = \sum_{i=1}^{t}\sum_{j=t+1}^{T} w_{ij}, \\
& \sum_{i=1}^{t} w_{it} = d_t \\
& 0 \le w_{it} \le d_t y_i, i \le t, \\
& 0 \le y_t \le 1, \\
& t = 1, ..., T, \\
& y_t \in \mathbb{Z}_t\}
\end{aligned}$$

Proposition 1.1. *Q is at least as good as P, i.e., Proj$_{\{1,\ldots,3T\}}Q \subseteq P$.*

Proof. Let $(x,y,s) \in \text{Proj}_{\{1,\ldots,3T\}}Q$. Thus, $\exists w$ such that $(x,y,s,w) \in Q$. Now,

$$\begin{aligned}
s_t - s_{t-1} - x_t &= \sum_{i=1}^{t}\sum_{j=t+1}^{T} w_{ij} - \sum_{i=1}^{t-1}\sum_{j=t+1}^{T} w_{ij} - \sum_{i=t}^{T} w_{ti} \\
&\vdots \\
&= -w_{tt} - \sum_{j=1}^{t-1} w_{jt} \\
&= -\sum_{j=1}^{t} w_{jt} \\
&= -d_t.
\end{aligned}$$

Therefore, (1) in (P) is satisfied. Also

$$x_t = \sum_{i=t}^{T} w_{ti} \le \sum_{i=t}^{T} d_i y_t = \left(\sum_{i=t}^{T} d_i\right) y_t = M_t y_t$$

and therefore (2) in (P) is satisfied. (3) and (4) in (P) are obviously satisfied. □

Proposition 1.2. *Proj*$_{\{1,\dots,3T\}} Q \neq P$ *and hence* Q *is a better formulation.*

Proof. (Sketch) Consider (x, y, s) defined by

$$s_t = 0, x_t = d_t, y_t = \frac{d_t}{M_t}, \forall t = 1, \dots, T$$

This is in P but $\nexists w$ such that $(x, y, w, s) \in Q$. □

Note 1. There exists an integer solution to the LP relaxation using Q. The proof is in N&W II.6.4.

Example 1.19. (*Bin Packing – An Extended Formulation with Exponentially Many Variables*) Given a set of bins, each of size b, n items, of length a_i for each $i = 1, \dots, n$, the problem is to pack all items into bins so as to use the minimum number of bins. The lower bound is

$$\text{Lower Bound} = \left\lceil \frac{\sum_{i=1}^{n} a_i}{b} \right\rceil$$

which we call the "**Liquid Packing**" bound, while the upper bound is

$$\text{Upper Bound} = \left\lceil \frac{\sum_{i=1}^{n} a_i}{b/2} \right\rceil = \left\lceil \frac{2\sum_{i=1}^{n} a_i}{b} \right\rceil.$$

Let K be an upper bound on the number of bins need and

$$y_{ik} = \begin{cases} 1, & \text{if item } i \text{ is put in bin } k \\ 0, & \text{otherwise} \end{cases}, x_k = \begin{cases} 1, & \text{if bin } k \text{ is used} \\ 0, & \text{otherwise} \end{cases}.$$

A compact formulation of this problem is

$$\begin{aligned} \min \ & \sum_{k=1}^{K} x_k \\ \text{s.t.} \ & \sum_{j=1}^{n} a_j y_{jk} \le b x_k, & \forall k \quad (P) \\ & \sum_{k=1}^{K} y_{jk} = 1, & \forall i \quad (P) \\ & 0 \le x \le 1 & (P) \\ & 0 \le y \le 1 & (P) \\ & x \in \mathbb{Z}^k, y \in \mathbb{Z}^k \end{aligned}$$

This formulation is weak. In fact, using only the constraints (P), the optimal objective value z_{LP} is the liquid packing bound. As an exercise, prove this.

Exercise: If we use $y_{jk} \le x_k$ for all j, k to relate y_{jk} to x_k instead, would the formulation be stronger?

Example 1.20. Consider $a_1 = 2, a_2 = 3, a_3 = 4, a_4 = 5, b = 11$. We can use cutting patterns $\{1, 2, 3\}, \{4, 3\}$.

Extended Formulation

Let $\mathcal{F} = \{F \subseteq \{1, \dots, n\} : F \neq \emptyset, \sum_{j \in F} a_j \le b\}$ and $M = |\mathcal{F}|$ so $\mathcal{F} = \{F_1, F_2, \dots, F_M\}$. Let

$$x_i = \begin{cases} 1, & \text{if cutting pattern } i \text{ is used} \\ 0, & \text{otherwise} \end{cases}, \forall i = 1, \dots, M$$

The formulation is

$$\min \sum_{i=1}^{M} x_i$$
$$\text{s.t.} \sum_{i=1 \text{ s.t. } j \in F_i}^{M} x_i = 1, \quad \forall j = 1, ..., n.$$

Note we can define the coefficient matrix

$$G_{ij} = \begin{cases} 1, & \text{if } j \in F_i \\ 0, & \text{otherwise} \end{cases}, i = 1, ..., M, j = 1, ..., n$$

and therefore the constraints are $Gx = 1$. Note that this is thus a set partitioning problem. A more efficient formulation is to use only maximal elements of $\mathcal{F}$. Let's say

$$\hat{\mathcal{F}} = \left\{ F \in \mathcal{F} : b - \sum_{j \in F} a_j < a_{j'}, \forall j' \notin F \right\}$$

and define $(\hat{G}_{ji})$ accordingly (columns are indicator vectors of sets in $\hat{\mathcal{F}}$.

Example 1.21. We could define

$$\hat{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

for our previous example.

The alternative model would then be

$$\min_{x \in \{0,1\}^{\hat{M}}} \sum_{i=1}^{\hat{M}} x_i$$
$$\text{s.t. } \hat{G}x \geq 1$$

where $\hat{M} = |\hat{\mathcal{F}}|$.

Remark 1.3. Note that in the column generation step, to price, we wish to find $F \subseteq \{1, ..., n\}$ such that

$$\sum_{j \in F} a_j \leq b \text{ and } 1 - \sum_{j \in F} \Pi_j < 0.$$

This can be done via the optimization problem

$$\min \left[1 - \sum_{j=1}^{n} \Pi_j y_j \right]$$
$$\text{s.t.} \sum_{j=1}^{n} a_j y_j \leq b$$
$$y \in \{0,1\}^n$$

where

$$y_j = \begin{cases} 1, & \text{if } j \in F \\ 0, & \text{otherwise} \end{cases}.$$

2 Computational Complexity

Goal: Classify problems according to how difficult they are to solve.

- In the worst case, over all possible instances of the problem
- Asymptotically, i.e. as the size of the instance grows

Example 2.1. 1. Hamiltonian cycle problem (HCP):

Given an undirected graph $G = (V, E)$, does there exist a simple cycle in the graph that visits every vertex? This is an example of a **decision problem**; its answer is YES or NO.

2. Knapsack Problem (KP):

Given $a, c \in \mathbb{Z}^n_+$ and $b \in \mathbb{Z}$, find

$$\begin{aligned} \max\ & cx \\ \text{s.t. } & ax \leq b \\ & x \in \mathbb{Z}^n_+. \end{aligned}$$

This is an **optimization problem**.

Definition 2.1. An **instance** is defined by its data. For example,

$$A_{5\times 6} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

is an example node-edge incidence matrix for the first HCP example. This can be encoded in a computer as a triple (B_1, B_2, B_3) in binary where B_1, B_2, B_3 respectively encode the number of nodes, number of edges, and entries in the node-incidence matrix.

Computing

In computing optimization or decision problems, we generally assume that:

- all data is rational
- instance data is encoded as a binary string

For example, an HCP instance (V, E) with $n = |V|, m = |E|$ may be encoded with approximately $\log(n) + \log(m) + nm$ bits.

Exercise 2.1. How many bits are needed to encode a KP instance? (*Note*: different encodings are possible!)

Definition 2.2. Under a giver encoding, the **size** of an instance, a.k.a. as its **length** is the number of bits needed to encode it as a binary string. Formally, we say an instance is a binary string, i.e. an element of $\{0, 1\}^*$.

Definition 2.3. A decision problem $P \subseteq \{0, 1\}^*$ is the set of instances that have YES answers. If there is a computer program (algorithm) that will do this, we say it "recognizes P".

The important issue is: how long will it take to do this as a function of the instance size?

Definition 2.4. The **complement** of a problem P is

$$\bar{P} = \{0, 1\}^* \backslash P$$

and note that sometimes recognizing $\bar{P}$ is a lot harder than recognizing P.

Definition 2.5. Given a problem P, consisting of a (possibly infinite) set of instances, and an algorithm A for solving P, we define

$$r_A^P(\Pi) = \text{the running time required for } A \text{ to solve } P$$

where the running time can be interpreted as the number of basic computational steps e.g. $+, -, \times, \div, :=$, and Π is an instance of P, i.e. $\Pi \in P$. In **worst-case analysis**, we examine

$$t_A^P(n) = \max\{r_A^P(\Pi) : \Pi \in P, l(\Pi) = n\}$$

where $l(\Pi)$ is the length of an instance.

Asymptotic Behaviour

How does $t_A^P(n)$ behave as $n \to \infty$. This is characterized using "big-oh" notation. A function $f : \mathbb{Z}_+ \mapsto \mathbb{R}$ is said to be "big-oh of $g(n)$" or "of the order of $g(n)$" if there exists a constant c and a positive number n such that

$$\forall n \geq N, f(n) \leq c \cdot g(n).$$

This is written as "$f(n)$ is $\mathcal{O}(g(n))$" or $f(n) \sim \mathcal{O}(g(n))$ and means that $g(n)$ grows at least as fast as $f(n)$ as n grows.

Example 2.2. The function $0.1n^2 + 999n - 53$ is $\mathcal{O}(n^2)$. It is also $\mathcal{O}(n^3)$.

Notation 1. We use the following terminology:

Orders of Common Functions	
$\mathcal{O}(1)$	Constant
$\mathcal{O}(\log n)$	Logarithmic
$\mathcal{O}(n)$	Linear
$\mathcal{O}(n^2)$	Quadratic
$\mathcal{O}(n^c), c \geq 1$	Polynomial
$\mathcal{O}(c^n), c \geq 1$	Exponential

Exercise 2.2.

Why is $\log(n) \sim \mathcal{O}(n^c)$ for any $c > 0$ while n^c is not $\mathcal{O}(\log(n))$?

Why is $n^c \sim \mathcal{O}(2^n)$ for any $c > 0$ while 2^n is not $\mathcal{O}(n^c)$?

2.1 Classes of Problems

Definition 2.6. The set of of decision problems P for which there exists an algorithm that recognizes P and takes running time the order of a polynomial function of the size of the input instance is called class

$$\mathcal{P} = \{P : \exists A \text{ that recognizes } \mathcal{P} \text{ with } t_A^P(n) \sim \mathcal{O}(n^c), \exists c \geq 1\}.$$

Examples of Problems in $\mathcal{P}$

- Shortest path in a network with non-negative arc lengths
- Longest path in an acyclic network is $\mathcal{O}(m)$ where m is the number of arcs
- Max flow
- Matroid optimization (e.g. minimum spanning tree)
- Matching
- Linear Programming

Optimization $\equiv$ Decision

For integer-valued optimization problem

$$z = \max\{f(x) : x \in X\}$$

there is a corresponding decision problem $(f, X, K\}$ that is of equivalent difficulty: given $k \in \mathbb{Z}$, does there exist $x \in X$ such that $f(x) \le K$?

Provided we have bounds L and U on the optimal value z of an instance with $L \le z \le U$ and $\log(\max\{|L|, |U|\})$ is polynomial in the size of the instance, the optimization problem can be solved using bisection search with polynomially many calls to an algorithm A that recognizes the decision problem. In particular, it requires $\log(U - L)$ steps.

The converse is obvious (given K, solve the optimization problem to get z: if $z \ge K$ then YES else NO).

Example 2.3. An interesting case is whether or not the Knapsack Problem (KP) is in $\mathcal{P}$. Here is one approach to solving the KP instance $\max\{cx : ax \le b, x \in \mathbb{Z}^n_+\}$. Define the digraph $D = (V, A)$ by

$$\begin{aligned} V =& \{0, 1, 2, ..., b\} \\ A =& \{(i, i + a_j) : i \in V, i + a_j \in V \text{ and } j = 1, 2, ..., n\} \cup \\ & \{(i, i + 1) : i \in V, i \le b - 1\} \end{aligned}$$

and without loss of generality, we may assume that the a_j for $j = 1, 2, ..., n$ are distinct. Arc lengths are

$$\begin{aligned} d_{(i,i+a_j)} &= c_j \text{ for each } (i, i + a_j) \in A \\ d_{(i,i+1)} &= 0 \text{ for each } i = 1, ..., b - 1 \text{ if } a_j \ne 1 \text{ for } j = 1, 2..., n. \end{aligned}$$

Now any path from 0 to b in D corresponds to a KP solution with value equal to the length of the path, and vice versa. Thus finding the longest path in D solves KP. Recall the longest path in an acyclic graph can be solved in time $\mathcal{O}(|A|)$. Note that D is acyclic by construction. This yields an algorithm that solves KP in time $\mathcal{O}(|A|) = \mathcal{O}(nb)$.

The length of the KP input data is polynomial in n and in $l(b) \approx \log(b)$ since

$$l(\Pi) \le l(b) + n(l(\bar{c}) + l(\bar{a}))$$

where Π is a KP instance, $\bar{c} = \max_{i=1,...,n} c_i$, and $\bar{a} = \max_{i=1,...,n} a_i$. However, $nb \sim \mathcal{O}(2^{\log(b)} n)$ and hence this approach does not yield a polynomial time algorithm to solve KP.

Definition 2.7. If an algorithm runs in polynomial time in the input data (not its length) then it is said to be a **pseudopolynomial time algorithm**. This is equivalent to saying the algorithm is polynomial in a unary encoding.

Definition 2.8. The set of **nondeterministic polynomial** $(\mathcal{NP})$ **problems** is the set of decision problems where

$$\begin{aligned} \mathcal{NP} =& \{P : \exists \text{ a certifier for } P \text{ that runs in polynomial time} \\ & \text{in the length of the input data and that, for every} \\ & \text{YES instance, returns YES for some certificate of} \\ & \text{length polynomial in the length of the instance}\} \end{aligned}$$

Example 2.4.

(1) 0-1 IP FEASIBILITY: given $A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m$, does there exist $x \in \{0, 1\}^n$ such that $Ax \le b$?

* Instance: $\Pi = (A, b)$

* Certificate: x

* Certifier: Substitutes in x, checks $Ax \le b$, returns YES if so, else MAYBE

$\implies$ Clearly this can be done with $\mathcal{O}(mn)$ which are basic computational operations, and $x \sim \mathcal{O}(n)$. Therefore, this certifier satisfies the requirements for this problem to be in $\mathcal{NP}$.

(2) COMPOSITES: Given $x \in \mathbb{Z}_+$, is x a composite number, i.e., $\exists(a, b) \in \mathbb{Z}^2_+$ such that $ab = x$, $a, b \ne 1$?

* Instance: x

* Certificate: (a, b)

* Certifier: Multiply ab and check if it is equal to x, with $a, b < x$.

$\implies$ We have $l(a, b) \sim \mathcal{O}(2l(x)) \sim \mathcal{O}(l(x))$ and since the certifier needs a small number of computations, this problem is in $\mathcal{NP}$.

(3) $HCP \in \mathcal{NP}$

$\implies$ The certificate is an ordered pair of vertices, certifier checks every vertex appears exactly once in the list, and between every consecutive pair in the list an edge exists, so it runs in time $\mathcal{O}(|V|)$.

(4) Given a set $N = \{1, ..., n\}$ and integers $c_1, ..., c_n, K$, and L, are there K distinct subsets of N, say $S_1, S_2, ..., S_k \subseteq N$ such that

$$\sum_{j \in S_i} c_j \geq L, \text{ for } i = 1, ..., K?$$

$\implies$ Suppose $K \sim \mathcal{O}(2^n)$ and $\Pi = (c_1, ..., c_n, K, L)$ is a YES instance. Notice

$$\begin{aligned} l(\Pi) &\sim \mathcal{O}(n \lceil \log \bar{c} \rceil + \lceil \log K \rceil + \lceil \log L \rceil) \\ &\sim \mathcal{O}(n \lceil \log \bar{c} \rceil + \lceil \log 2^n \rceil + \lceil \log L \rceil) \\ &\sim \mathcal{O}(n \lceil \log \bar{c} \rceil + n + \lceil \log L \rceil) \end{aligned}$$

but the length of any certificate proving YES, say $S_1, S_2, ..., S_k$, must be at least of length $\mathcal{O}(K)$ since $l(S_i) \geq 1$ for $i = 1, ..., K$, i.e., the length of the certificate is at least $\mathcal{O}(2^n)$, which is **not** polynomial in $l(\Pi)$. Therefore, this problem is **not** in $\mathcal{NP}$.

(5) LP FEASIBILITY: Given $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^m$, does there exist $x \in \mathbb{R}^n$ with $Ax \leq b$?

$\implies$ For a YES instance, a certificate x, can be checked by substitution. How can we be sure that some feasible solutions have small entries (with lengths polynomial in $l(A, b)$)?

Answer: (N&W I.5.3. Proposition 3.1) Combined with some polyhedral theory, tells us:

* $l(\det(A))$ is polynomial in $l(A)$. [Schrijver, 1986, Theorem 3.2]

* If $X = \{x : Ax \leq b\} \neq \emptyset$ then it has a lowest dimensional face that it not empty.

* Any lowest dimensional face can be written as $\{x : \tilde{A}x = b\}$ where $(\tilde{A}, \tilde{b})$ is an $\tilde{m} \times (n+1)$ submatrix of (A, b), $\tilde{m} \leq m$ [Schrijver, 1986, Theorem 8.4 & (22), see also (20)]

* By Gauss-Edmonds elimination (Edmonds 1967) one can find a solution to a system of equations no larger than the determinants of the minors of $(\tilde{A}, \tilde{b})$ [Schrijver 1986, Corollary 3.2b]

Therefore if $X \neq \emptyset$ there exists $x \in X$ with $l(x) \sim \mathcal{O}(\text{poly}(l(A, b)))$ and LP FEASIBILITY is in $\mathcal{NP}$.

Proposition 2.1. $\mathcal{P} \subseteq \mathcal{NP}$

Proof. Consider $P \in \mathcal{P}$. There must exist an algorithm that recognizes P and runs in polynomial time. Use that algorithm as the certifier, with an empty certifier. □

Proposition 2.2. *For all $P \in \mathcal{NP}$, there exists an algorithm A that solves P with $t_A^P(n) \sim \mathcal{O}\left(2^{p(n)} q(n)\right)$ where p and q are polynomials.*

Proof. P must have a certifier algorithm, $C(\Pi, c)$ that returns YES for some certificate c with $l(c)$ polynomial in $l(\Pi)$ if and only if $\Pi \in P$. Let p be that polynomial function.

Let A be the algorithm: given Π, for each certificate c with $l(c) \leq p(l(\Pi))$ run $C(\Pi, c)$. If for any of these c, $C(\Pi, c)$ returns YES, A returns YES; otherwise A returns NO. $C(\Pi, c)$ is run at most $\mathcal{O}\left(2^{p(l(\Pi))}\right)$ and in the $\mathcal{O}(q(l(\Pi))$ for some polynomial q. The result follows. □

Definition 2.9. Recall that the **complement of** $\mathcal{P}$ is

$$\bar{P} = \{0, 1\}^* \backslash P.$$

The complexity class co-$\mathcal{NP}$ is

$$\text{co-}\mathcal{NP} = \{\bar{P} : P \in \mathcal{NP}\}$$

Examples of Problems in co-$\mathcal{NP}$

(1) LP INFEASIBILITY

* Given $(A, b) \in \mathbb{Q}^{m\times(n+1)}$, is $\{x \in \mathbb{R}^n : Ax \leq b\} = \emptyset$?

* Complement of LP FEASIBILITY

(2) PRIMES

* Given $x \in \mathbb{Z}_+$ is x prime?

* Complement of COMPOSITES

* Is in both co-$\mathcal{NP}$ and $\mathcal{NP}$ (see V.R. Pratt, 1975)

(3) 0-1 IP INFEASIBILITY

* Given $(A, b) \in \mathbb{Z}^{m\times(n+1)}$ is $\{x \in \{0, 1\}^n : Ax \leq b\} = \emptyset$

* Complement of 0-1 IP FEASIBILITY

Proposition 2.3. $P \in \mathcal{P} \iff \bar{P} \in \mathcal{P}$

Proof. The same algorithm that recognizes P also recognizes $\bar{P}$, simply by exchanging YES and NO output. □

Corollary 2.1. *This immediately implies that LP INFEASIBILITY is in P and in general co-$\mathcal{P} = \mathcal{P}$.*

Remark 2.1. We have the relationships:

* co-$\mathcal{P} = \mathcal{P}$

*co-$\mathcal{NP} \supseteq \mathcal{P}$

* $\mathcal{NP} \supseteq \mathcal{P}$

* It is unknown if $\mathcal{NP} = \mathcal{P}$, which is an open (one million dollar) problem!

Theorem 2.1. *If $\mathcal{NP} \neq$ co-$\mathcal{NP}$, then $\mathcal{P} \neq \mathcal{NP}$.*

Proof. We know that $\mathcal{P} =$ co-$\mathcal{P}$. Suppose that $\mathcal{P} = \mathcal{NP}$. Then, co-$\mathcal{P} =$ co-$\mathcal{NP} = \mathcal{P} = \mathcal{NP}$ and thus co-$\mathcal{NP} = \mathcal{NP}$.

Polynomially Reducibility □

Definition 2.10. A problem P_1 is said to be **polynomially reducible** to problem P_2 if every instance of P_1 can be solved by solving (one or more) instances of P_2 where the number and size of the instances of P_2 to be solved is polynomial in the length (size) of the P_1 instance.

One instance of P_2 to be solved is a **polynomial transformation**: there exists $f : P_1 \mapsto P_2$ such that $\Pi \in P_1$ if and only if $f(\Pi) \in P_2$ and f runs in polynomial time in $l(\Pi)$.

We denote this as: $P_1 \propto P_2 \equiv P_1$ is polynomially reducible to P_2. In this case, P_2 is at least as hard as P_1.

Definition 2.11. A problem $P \in \mathcal{NP}$ is $\mathcal{NP}$**-Complete** or $\mathcal{NPC}$ if every problem in $\mathcal{NP}$ is polynomially reducible to $\mathcal{P}$.

Example 2.5. The first problem proved to be in $\mathcal{NPC}$ is SATISFIABILITY [SAT] (Steven A. Cook, 1971):

* Given n literals (e.g. logic element, can be either T or F), m clauses $C_1, C_2, ..., C_m$ where a clause is a set of literals or their complements (negations), which are components of the clause, that is true if and only if at least one of its components is true.

* Does there exist an assignment of truth values to literals so that all m clauses are true?

* e.g. $n = 3, m = 3$ with

** Literals $x_1, x_2, x_3 \in \{T, F\}$

** $C_1 = \{x_1, x_2\}, C_2 = \{\neg x_2, \neg x_3\}, C_3 = \{\neg x_1, x_3\}$

*** Also written as $C_1 = x_1 \vee x_2, C_2 = \neg x_2 \vee \neg x_3, C_3 = \neg x_1 \vee x_3$

** Does there exist x_1, x_2, x_3 such that $(x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3) = T$ (written in conjunctive normal form (CNF))?

** This is a YES of SAT since $(x_1, x_2, x_3) = (T, F, T)$ is a solution.

Algorithm 1. *To prove P is $\mathcal{NP}$-Complete:*

(1) Show P is in $\mathcal{NP}$

(2) Find $P' \in \mathcal{NPC}$ that you show can be polynomially reduced to P, i.e. prove that $P' \propto P$.

Proposition 2.4. *0-1 IP FEASIBILITY is $\mathcal{NP}$-Complete.*

Proof. We showed earlier that 0-1 IP is in $\mathcal{NP}$. We will reduce SAT to 0-1 IP as follows. Given an instance of SAT with n literals, m clauses, $C_1, ..., C_m \subseteq \{1, 2, ..., n\} \cup \{\bar{1}, \bar{2}, ..., \bar{n}\}$, construct an instance of 0-1 IP FEASIBILITY with variables

$$x_i \in \{0, 1\} \text{ has } x_i = 1 \iff \text{literal } i \text{ is true}$$

for $i = 1, 2, ..., n$. The IP has a constraint for each clause as

$$\sum_{\substack{i=1 \\ i \in C_j}}^{n} x_i + \sum_{\substack{i=1 \\ \bar{i} \in C_j}}^{n} (1 - x_i) \geq 1 \text{ for } j = 1, 2, ..., m.$$

Clearly the 0-1 IP FEASIBILITY instance is YES if and only if the SAT instance is YES. The size of this 0-1 IP FEASIBILITY instance is obviously polynomial in the size of the SAT instance (# of variables = n, # of constraints = m). $\square$

Proposition 2.5. *0-1 IP FEASIBILITY with all entries in (A, b) being in $\{0, 1\}$ is $\mathcal{NP}$-Complete.*

Proof. Exercise. $\square$

Example 2.6.

SAT Re-statement

Given a set of n literals, $x_1, ..., x_n$ and a set of m clauses $C_1, ..., C_n$ where

$$C_i = \left(\bigvee_{j \in S_i} x_j \right) \vee \left(\bigvee_{j \in S_i'} \bar{x}_j \right) \text{ for } i = 1, 2, ..., m$$

is defined by $S_i, S_i' \subseteq N = \{1, 2, ..., n\}$, is there an assignment of T/F values to the literals so that $C_1 \wedge C_2 \wedge ... \wedge C_m = T$?

Example 2.7.

3SAT

A special case of SAT with each clause containing 3 components.

Proposition 2.6. *SAT $\propto$ 3SAT*

Proof. (Sketch) Suppose we have an instance of SAT.

Exercise: Show that there exists an equivalent SAT instance, "not too large", with every clause of size ≥ 3.

Now consider a clause of size $k > 3$ (k components) say

$$C = \underbrace{(x_1 \vee x_2)}_{y_1} \vee \underbrace{(x_3 \vee x_4)}_{y_2} ... \vee x_k$$

We will replace C by

$$\begin{cases} y_1 \vee ... \vee y_{\frac{k}{2}}, & \text{if } k \text{ is even} \\ y_1 \vee ... \vee y_{\frac{k-1}{2}} \vee x_k, & \text{if } k \text{ is odd} \end{cases}$$

by adding clauses that ensure

$$y_j \equiv x_{2j-1} \vee x_{2j}, \text{ for } j = 1, ..., \left\lfloor \frac{k}{2} \right\rfloor$$

This can be done with 4 extra clauses:

$$\begin{aligned}(\bar{y}_1 \vee x_1 \vee x_2) &\to x_1 = F, x_2 = F \implies y_1 = F \\ \wedge(y_1 \vee x_1 \vee \bar{x}_2) &\to x_1 = F, x_2 = T \implies y_1 = T \\ \wedge(y_1 \vee \bar{x}_1 \vee x_2) &\to x_1 = T, x_2 = F \implies y_1 = T \\ \wedge(y_1 \vee \bar{x}_1 \vee \bar{x}_2) &\to x_1 = T, x_2 = T \implies y_1 = T.\end{aligned}$$

By repeating this process at most $\log k$ times, C will be replaced by clauses with at most 3 components. In total, at most k new literals and at most $4k$ new clauses. Therefore the resulting 3SAT instance has size polynomial in the size of the given SAT instance. □

Corollary 2.2. *3SAT is $\mathcal{NP}$-Complete. (In $\mathcal{NP}$ since it is a special case of SAT)*

Example 2.8.

CLIQUE

Given an undirected graph $G = (V, E)$ and an integer K, does there exist $S \subseteq V$ such that $|S| \geq L$ and $\{i, j\} \in E$ for all $i, j \in S, i \neq j$?

NODE PACKING

Given $G = (V, E)$, integer L, is there $S \subset V, |S| \geq L$ such that $\{v, w\} \notin E$ and $v, w \in S$?

SET PACKING

Given $T = \{t_1, ..., t_m\}$, a family of subsets $\mathcal{F} = \{T_1, ..., T_n\}, T_j \subseteq T$ for $j = 1, 2, ..., n$, integer K, does there exist a subset of $\mathcal{F}$ of cardinality at least K consisting of *disjoint* subsets of T?

SET PARTITIONING

Given $Q = \{q_1, ..., q_s\}, \mathcal{G} = \{Q_1, ..., Q_r\}, Q_j \subseteq Q$ for $j = 1, ..., r$, does there exist $S \subseteq \{1, ..., r\}$ such that $Q_j \cap Q_k = \emptyset$ for all $j, k \in S, j \neq k$ and $\bigcup_{j \in S} Q_j = Q$?

Some Important $\mathcal{NP}$-Complete Problems and Reductions

Corollary 2.3. *We have:*

SAT → 3SAT
SAT → 0-1 FEASIBILITY
SAT → CLIQUE → NODE PACKING
NODE PACKING → VERTEX COVER → HAMILTONIAN CYCLE
NODE PACKING → SET PACKING
SET PACKING → KNAPSACK
SET PACKING → SET PARTITIONING
SET PARTITIONING → SUBSET SUM
SET PARTITIONING → 3 SET PARTITIONING

Proposition 2.7. *SAT $\propto$ CLIQUE*

Proof. (Example) Given SAT instance

$$\underbrace{(x_1 \vee \bar{x}_3)}_{C_1} \wedge \underbrace{(\bar{x}_2 \vee x_3)}_{C_2} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{C_3}$$

We will create nodes (i, j) for each literal where i is the component in the clause and j is the clause index. In the above example, we have nodes $\{(1, 1), (\bar{3}, 1), (\bar{2}, 2), (3, 2), (\bar{1}, 3), (\bar{2}, 3)\}$. Edges are created for pairs of nodes that can be true at the same time.

Formally

$$\begin{aligned}V &= \{(y, j) : y \text{ is a component of clause } j\} \\ E &= \{\{(y, j), (w, j')\} : (y, j), (w, j') \in V, j \neq j', y \neq \bar{w}\}\end{aligned}$$

and the SAT instance is YES if and only if there exists a clique of size $\geq m = \#$ of clauses in this graph. For any clique, each vertex corresponds to a distinct clause, and the existence of all edges between the pairs of vertices in the clique ensures their components can all be true all at once, i.e. $\forall (y, j) \in S$ such that a clique $y = T$ is feasible for SAT. □

Proposition 2.8. *CLIQUE $\propto$ NODE PACKING*

Proof. Given $(G = (V, E), K)$ a CLIQUE instance, construct $G' = (V, \bar{E})$ where

$$\bar{E} = \{\{v, w\} : v, w \in V, v \neq w, \{v, w\} \notin E\}.$$

Then S is a clique in $G \iff S$ is a node packing in G'. Therefore, take $L = K$ to get (G', L) an equivalent instance of NODE PACKING. □

Proposition 2.9. *NODE PACKING $\propto$ SET PACKING*

Proof. Given a node packing instance $(G = (V, E), L)$, take $T = E$, $\mathcal{F} = \{\delta(v) : v \in V\}$, and $K = L$. □

Proposition 2.10. *SET PACKING $\propto$ SET PARTITIONING*

Proof. (Sketch/Illustration) Consider $T = \{1, 2, ..., 5\}, m = 5$, and

$$\mathcal{F} = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{1, 5\}, \{2, 4\}\}, n = 6$$

with $K = 2$. Construct the SET PARTITIONING instance

$$\begin{aligned} Q = \{ &\underbrace{1, 2, 3, 4, 5}_{T}, \quad \underbrace{6, 7}_{K \text{ new elements}} \quad \} \\ \mathcal{G} = \{ &\{1, 2, 6\}, \{1, 2, 7\}, \\ &\{2, 3, 6\}, \{2, 3, 7\}, \\ &\{3, 4, 6\}, \{3, 4, 7\}, \\ &\underbrace{\quad \vdots \quad}_{\text{each set in } \mathcal{F}, T_j \cup \{q_k\}, \forall \text{ new } q_k, \forall k} \\ &\underbrace{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}}_{\{t_i\}, \forall i}\} \end{aligned}$$

{Begin scribe} □

Problem 2.1. SUBSET SUM: Given $b \in \mathbb{Z}^n$, target $B \in \mathbb{Z}_1$, does there exist $S \subseteq N = \{1, ..., n\}$ such that $\sum_{i \in S} b_i = B$?

Proposition 2.11. *SET PARTITIONING $\propto$ SUBSET SUM*

Proof. Recall SET PARTITIONING: given $A \in \{0, 1\}^{m \times n}$, does there exist $x \in \{0, 1\}^n$ such that $Ax = \mathbf{1}$? Suppose we take u_i for $i = 1, ..., m$ and form the weighted row sum $u^T Ax = u^T \mathbf{1}$. Clearly if the SET PARTITIONING instance is a YES instance, then the SUBSET sum instance with $b = u^T A, B = u^T \mathbf{1}$ is also YES. □

Problem 2.2. What values of $(u_i)_{i=1,...,m}$ can we choose to ensure this is reversible, i.e., that if $x \in \{0, 1\}^n$ such that $u^T Ax = u^T \mathbf{1}$ then $Ax = \mathbf{1}$?

Answer. Note that for $x \in \{0, 1\}$ and any $i \in \{1, ..., m\}$, it must be the case that $A_i x \leq n$ where A_i is the i^{th} row of A. Let $u_i = (n + 1)^{i-1}$ for $i = 1, ..., m$. Then we can ensure reversibility by the following lemma.

Lemma 2.1. *Let $\lambda \in \mathbb{Z}_+, \lambda \geq 1$. Then if $\sum_{i=0}^{r} \alpha_i \lambda^i = \sum_{i=0}^{r} \lambda^i$ for $\alpha_i \in \{0, 1, ..., \lambda - 1\}$ it must be that $\alpha_i = 1$ for all $i = 1, ..., r$.*

Proof. Observe that for any $k \in \mathbb{Z}_+$ and any $\beta_1, ..., \beta_{k-1} \in \{0, 1, ..., \lambda - 1\}$ it must be that $(*)$ $\sum_{i=0}^{k-1} \beta_i \lambda^i < \lambda^k$ (Proof: exercise). Now we proceed by (backward) induction on r. Suppose $\alpha_r = 0$. Then,

$$\sum_{i=0}^{r} \alpha_i \lambda^i = \sum_{i=0}^{r-1} \alpha_i \lambda^i < \lambda^r < \sum_{i=0}^{r} \lambda^i$$

which contradicts the required equation. Suppose that $\alpha_r \geq 2$. Then,

$$\sum_{i=0}^{r} \alpha_i \lambda^i \geq \alpha_r \lambda^r \geq 2\lambda^r > \lambda^r + \sum_{i=0}^{r-1} \lambda^i = \sum_{i=0}^{r} \lambda^i$$

by $(*)$ with $\beta_1 = ... = \beta_r = 1$, which again contradicts the required equation. Using induction on r in a descending order then proves the result. □

Proof. (of Proposition) Given $A \in \{0, 1\}^{m \times 1}$, set $u_i = (n+1)^{i-1}$ for $i = 1, ..., m$. Then,

$$\begin{aligned} u^T A x = u^T \mathbf{1} &\implies \sum_{i=1}^{m} u_i A_i x = \sum_{i=1}^{m} u_i \\ &\implies \sum_{i=1}^{m} A_i x (n+1)^{i-1} = \sum_{i=1}^{m} (n+1)^{i-1} \\ &\implies A_i x = 1 \text{ for all } i \in \{1, ..., m\} \end{aligned}$$

by the previous lemma with $\lambda = n + 1$ and $\alpha_i = A_i x$ for $i = 1, ..., m$. So a YES instance of SUBSET SUM is a YES instance of SET PARTITIONING with $b = u^T A$ and $B = u^T \mathbf{1}$. □

Corollary 2.4. *As we have shown SET PARTITIONING $\propto$ SUBSET SUM previously, the two problems are equivalent.*

Remark 2.2. Now $B = \sum_{i=0}^{m} u_i$ and for all j, $b_{ij} = \sum_{i=1}^{m} u_i A_{ij} \leq \sum_{i=1}^{m} u_i$. Therefore the length of (b, B) is bounded above by

$$(n+1)l\left(\sum_{i=1}^{m} u_i\right) = (n+1)l\left(\sum_{i=1}^{m} (n+1)^i\right) \leq (n+1)l\left((n+1)^m\right) \sim (n+1)m\log(n+1)$$

which is polynomial in $l(A) \leq mn$.

Remark 2.3. We know SUBSET SUM is solvable in pseudopolynomial time (as an exercise, show that it is equivalent to the 0-1 Knapsack Problem). Specifically, the instance (b, B) for $b \in \mathbb{Z}^n$ can be solved in time $\mathcal{O}(n\bar{b})$ where $\bar{b} = \max_{j=1,\ldots,n} b_j$. This does not mean that SET PARTITIONING can be solved in pseudopolynomial time as well!

From the above reduction $\bar{b} \sim \mathcal{O}(n^m)$ so the running time of the pseudopolynomial algorithm would be $\mathcal{O}(n^{m+1})$ which is not polynomial in the data for A.

Weak & Strong $\mathcal{NP}$-Completeness

Recall an algorithm is pseudopolynomial if it runs in polynomial time on a unary encoding of the data.

Definition 2.12. An $\mathcal{NP}$-complete problem that can be solved by a pseudopolynomial algorithm is **weakly $\mathcal{NP}$-complete**.

Definition 2.13. An $\mathcal{NP}$-complete problem that is $\mathcal{NP}$-complete under a unary encoding of the data is said to be **strongly $\mathcal{NP}$-complete**. Equivalently if the problem is restricted so that the largest number in any instance Π is bounded above by a polynomial function of $l(\Pi)$, then the problem is strongly $\mathcal{NPC}$.

It immediately follows that unless $\mathcal{P} = \mathcal{NP}$, then if $\mathcal{P} = \mathcal{NPC}$ is not a "numbers problem", then $\mathcal{P}$ is strongly $\mathcal{NPC}$.

Example 2.9.

Number Problems

(1) LP FEASIBILITY is in $\mathcal{P}$

(2) 0-1 IP FEASIBILITY is strongly $\mathcal{NPC}$ (from our SAT reduction)

(3) TSP is strongly $\mathcal{NPC}$ (from our SAT reduction)

(4) KNAPSACK is weakly $\mathcal{NPC}$

(5) SUBSET SUM is strongly $\mathcal{NPC}$

Not Number Problems (all are strongly $\mathcal{NPC}$)

(1) HCP

(2) SAT, 3SAT

(3) CLIQUE ($K \leq n$) where K is the minimum size of clique needed and n is the number nodes in the graph

(4) SET PARTITIONING

$\mathcal{NP}$-Hard

It is possible to polynomially reduce an $\mathcal{NPC}$ problem to a problem that is not a decision problem. Such a problem would still be at least as hard as every problem in $\mathcal{NP}$. We thus extend to a more general class of problems: search problems.

Definition 2.14. Algorithm A solves search problem P if we have the following relationship

$$\begin{array}{ccccc} \Pi & \rightarrow & \boxed{A} & \rightarrow & \text{some } s \in S_P(\Pi) \\ & & \downarrow & & \text{if } S_P(\Pi) \neq \emptyset \\ & & \text{"no" if} & & \\ & & S_P(\Pi) = \emptyset & & \end{array}$$

where $S_P(\Pi)$ is the set of solutions of Π.

Definition 2.15. A **search problem** P consists of a set of instances where each instance consists of a finite set of objects. Also, for each instance $\Pi \in P$, there is a possibly empty set of **solutions** of Π, denoted by $S_P(\Pi)$, where a solution consists of a finite set of objects.

Example 2.10. An example of a search problem is the TSP with an objective (optimization version).

Definition 2.16. A search problem H is $\mathcal{NP}$**-Hard** if it is at least as hard as every problem in $\mathcal{NP}$. i.e. if $P \propto H$ for some $P \in \mathcal{NPC}$. (Exercise: Show that TSP (Decision) $\propto$ TSP (Optimization) via Bisection Search)

Remark 2.4. Note that every $\mathcal{NPC}$ problem is $\mathcal{NP}$-Hard.

Proposition 2.12. *0-1 IP is $\mathcal{NP}$-Hard.*

Proof. We know that 0-1 IP FEASIBILITY is $\mathcal{NP}$-Complete and polynomially reduces to 0-1 IP: take the objective vector with $c = 0$.

{End scribe} □

3 Easily Solved IPs

3.1 Matroids

Recall the Min Cost Spanning Tree (MCST) problem: given undirected connected graph $G = (V, E)$:

- $T \subseteq E$ induces a spanning tree in G given by the subgraph (V, T) if
  - (i) (V, T) is connected
  - (ii) (V, T) is acyclic or equivalently we will say T is acyclic
- Given costs c_e, for all $e \in E$, the MCST problem is to find

$$T^* \in \text{argmin} \left\{ \sum_{e \in T} c_e : T \text{ induces a spanning tree of } G \right\}$$

Kruskal's Algorithm for MCST (1956)

Algorithm 2. *In pseudocode,* ***Kruskal's algorithm*** *is:*

Set $T := \emptyset, S := E$
While ($S \neq \emptyset$ and $|T| < |V| - 1$)
Choose $e \in \text{argmin}\{c_e : e \in S\}$
$S := S \backslash \{e\}$
if $T \cup \{e\}$ is acyclic then $T := T \cup \{e\}$
End While

Remark 3.1. T is acyclic, $|T| = |V| - 1 \iff T$ induces a spanning tree. This is a "greedy" algorithm. It runs in time $\mathcal{O}(|E| \log |E|)$.

Example 3.1. Consider the following run of Kruskal's algorithm:

Cost-Ordered Edges	Kruskal's Algorithm
5, AD	✓
5, CE	✓
6, DF	X
7, AB	✓
7, BE	✓
8, BC	X
8, EF	X
9, BD	X
9, EG	✓, $|T| = 6, |V| = 7$
10, DE	
11, FG	

and we are done at edge EG.

Lemma 3.1. *If (V, U) is a spanning tree of $G = (V, E)$ and $e \in E \backslash U$, there exists a unique cycle in $(V, U \cup \{e\})$, and for any edge e' in this cycle, $e' \neq e$, we have $(V, (U \backslash \{e'\}) \cup \{e\})$ is a spanning tree of G.*

Proof. Exercise. □

Theorem 3.1. *Kruskal's algorithm yields a MCST.*

Proof. Suppose that Kruskal's algorithm yields a tree T. By definition, T is acyclic. Suppose that T is <u>not</u> optimal. For any optimal $\hat{T}$, $T \not\subseteq \hat{T}$ since otherwise the algorithm must have failed to add an edge that kept T acyclic which is impossible. Also $\hat{T} \not\subseteq T$ since $\hat{T} \neq T$, $\hat{T} \subset T$, and $\hat{T}$ acyclic implies that $\hat{T}$ does not span G (not connected).

Now let T^* be an optimal solution that minimizes $|T^* \backslash T| > 0$. Choose $e \in \text{argmin}\{c_e : e \in T \backslash T^*\}$. Clearly if $c_{e'} < c_e$ then $e' \in T \cap T^*$ or $e' \notin T \cup T^*$. Let $C \subseteq T^*$ be the edges in the unique cycle in $T^* \cup \{e\}$ as per the previous lemma. Obviously $C \not\subseteq T$, otherwise T would contain a cycle. So there exists $e' \in C \backslash T$. Thus $c_{e'} \geq c_e$ and by the previous lemma $(V, (T^* \backslash \{e'\}) \cup \{e\})$ is a spanning tree with cost

$$\sum_{e \in T^*} c_e \underbrace{-c_{e'} + c_e}_{\leq 0} \leq \sum_{e \in T^*} c_e \implies c_{e'} = c_e.$$

Thus $T^* \backslash \{e'\} \cup \{e\}$ induces an optimal spanning tree. However,

$$|(T^* \backslash \{e'\} \cup \{e\}) \backslash T| = |T^* \backslash T| - 1 < |T^* \backslash T|$$

contradicting the definition of T^*. □

Problem 3.1. What if we tried to solve Max Weighted Matching by a greedy algorithm? Consider the example

$$A \xrightarrow{1} B \xrightarrow{3} C \xrightarrow{4} D \xrightarrow{3} A.$$

The greedy solution will pick $\{AB, CD\}$ while the optimal solution is $\{BC, DA\}$. So when does a greedy algorithm work? The answer is: if there is matroid structure!

Independence System

- $N = \{1, ..., n\}$ a finite set
- $\mathcal{F}$ a collection of subsets
- $(N, \mathcal{F})$ is an **independence system** if $F_1 \in \mathcal{F}, F_2 \subseteq F_1 \implies F_2 \in \mathcal{F}$
- The dependent sets are $2^N \backslash \mathcal{F}$

Example 3.2. Some examples of independence systems are:

(1) Sets of linearly independent columns of a matrix

(2) Stable sets in a graph

(3) Solutions of 0-1 Knapsack Problems (with non-negative coefficients)

(4) Acyclic subgraphs (forests)

Exercise 3.1. Do matchings in a graph form an independence system?

Maximal Independent Set

- $(N, \mathcal{F})$ an independent system
- $F \in \mathcal{F}$ is a **maximal independent set** (a.k.a. a **basis** or in plural, **bases**) if $F \cup \{i\} \notin \mathcal{F}$ for all $i \in N \backslash F$

Matroid

Definition 3.1. An independence system $(N, \mathcal{F}), \mathcal{F} \neq \emptyset$ is a **matroid** if

$$F_1, F_2 \in \mathcal{F}, |F_1| < |F_2| \implies \exists i \in F_2 \backslash F_1 \text{ s.t. } F_1 \cup \{i\} \in \mathcal{F}$$

or equivalently for any $T \subseteq N$, every independent set that is a subset of T and is maximal in T has the same cardinality

$$m(T) = \max_{S \subseteq T} \{|S| : S \in \mathcal{F}\}$$

where $m(T)$ is called the **rank function**.

Exercise 3.2. Prove the two definitions are equivalent.

Note 2. $T \in \mathcal{F} \iff m(T) = |T|$. So in a matroid, maximal $\equiv$ maximum cardinality. Every basis in a matroid must have the same maximum cardinality; also every basis of the submatroid induced by any subset of N must have the same cardinality.

Example 3.3. Here are some examples of matroids:

(1) Cardinality Matroid: Given K, $N = \{1, ..., n\}$, the pair $(N, \{F \subseteq N : |F| \leq K\})$ is a matroid. (As an exercise, prove this and find the rank function)

(2) Matric Matroid: N is the index set of the columns of a matrix, $F \in \mathcal{F}$ if and only if the columns indexed by F are linearly independent. The rank function is the column rank of the submatrix induced by the given subset.

(3) Graphic Matroid: $G = (V, E)$ is a graph, $\mathcal{F} = \{F \subseteq E : (V, F) \text{ is acyclic}\}$ then $(E, \mathcal{F})$ is a matroid. For $T \subseteq E$, $m(T) = |V|$ subtract the number of connected components of (V, T).

(4) [**not** a matroid] Stable sets of a graph: If $\mathcal{F}$ is the set of subsets of stable sets, it is easy to construct an example where $F_1, F_2 \in \mathcal{F}, |F_1| < |F_2|$ but $F_1 \cup \{i\} \notin \mathcal{F}$ for some $i \in F_2 \backslash F_1$. For example, consider

$$\mathcal{F} = \{\{1\}, ..., \{5\}, \{1, 3\}, \{2, 5\}, \{3, 5\}, \{4, 5\}\}$$

with $F_1 = \{1\}, F_2 = \{2, 5\}$.

Exercise: Find $T \subsetneq N$ in the stable set example above so that not all maximal independent sets have the same cardinality.

Matroid Optimization

Given a matroid $(N, \mathcal{F})$ and $c \in \mathbb{R}^n$, we wish to solve the problem

$$\begin{aligned} \max_T \sum_{j \in T} c_j \\ \text{s.t. } T \in \mathcal{F} \end{aligned}$$

Proposition 3.1. *$T \in \mathcal{F}$ if and only if $|S \cap T| \leq m(S)$ for all $S \subseteq N$.*

Proof. ($\Longleftarrow$) Take

$$\begin{aligned} S = T, |S \cap T| = |T| &\implies m(T) \geq |T| \\ &\implies m(T) = |T| \\ &\implies T \in \mathcal{F} \end{aligned}$$

($\Longrightarrow$) For $S \cap T \subseteq T \implies S \cap T \in \mathcal{F}$ and hence $m(S) \geq |S \cap T|$ since $S \cap T \subseteq S$. $\square$

IP Model

Let

$$x_j = \begin{cases} 1, & \text{if } j \in T \\ 0, & \text{otherwise} \end{cases}.$$

Then for $S \subseteq N$, $|S \cap T| = \sum_{j \in S} x_j$. Therefore, the IP model is

$$\begin{aligned} \max \sum_{j \in N} c_j x_j \\ \text{s.t. } \sum_{j \in S} x_j \leq m(S), \forall S \subseteq N \\ x \in \{0, 1\}^n. \end{aligned}$$

This model has the integrality property its LP relaxation has an integral polytope, i.e., all its extreme points are integers. However, a greedy algorithm solves matroid optimization.

The Greedy Algorithm for Matroid $(N, \mathcal{F})$, $c \in \mathbb{R}^N$

sort $N = \{1, 2, ..., n\}$ so that $c_1 \geq c_2 \geq ... \geq c_n$.
set $S^0 := \emptyset, t := 1$
while ($c_t > 0$ and $t \leq n$) do:
if $S^{t-1} \cup \{t\} \in \mathcal{F}$ then
set $S^t := S^{t-1} \cup \{t\}$
else

set $S^t := S^{t-1}$

set $t := t + 1$

end while

Theorem 3.2. *(N&W, III.3.3., #3.1) At the end of the greedy algorithm, S^{t-1} is optimal.*

Theorem 3.3. *(N&W, III.3.3.,#3.2) If $(N, \mathcal{F})$ is an independence system that is not a matroid, then there exists $c \in \mathbb{R}^n$ such that the greedy algorithm's solution is not optimal.*

Therefore, for an independence system $(N, \mathcal{F})$, the greedy algorithm can be guaranteed to yield an optimal solution if and only if $(N, \mathcal{F})$ is a matroid. So "greediness" characterizes matroids.

Matroid Intersection

The intersection of two matroids $(N, \mathcal{F}_1)$ and $(N, \mathcal{F}_2)$ having the same ground set, N, is the independence system

$$(N, \{F \subseteq N : F \in \mathcal{F}_1 \cap \mathcal{F}_2\}$$

Exercise 3.3. Prove that the above is an independence system.

Intersection of Two Matroids

Example 3.4.

(1) *Intersection of two partition matroids*

* Given a bipartite graph $G = (V \cup W, E)$ with $V \cap W = \emptyset$ and $E \subseteq \{\{v, w\} : v \in V, w \in W\}$.

* Take ground set E and define

$$\begin{aligned}\mathcal{F}_1 &= \{F \subseteq E : |\delta(v) \cap F| \leq 1, \forall v \in V\} \\ \mathcal{F}_2 &= \{F \subseteq E : |\delta(w) \cap F| \leq 1, \forall w \in W\}\end{aligned}$$

where $\{\delta(v) : v \in V\}$ partitions E.

* The set of all matchings in the bipartite graph is $\{F \subseteq E : E \in \mathcal{F}_1 \cap \mathcal{F}_2\}$. Therefore $(E, \{M : M$ is a matching in $G\}$ is the intersection of two partitions matroids.

(2) *Intersection of a partition matroid and a graphic matroid*

* Given a digraph $G = (V, A)$, $\{\delta^-(v) : v \in V\}$ partitions A.

* Therefore, the intersection of the two matroids is the set of all subsets of A with at most one arc entering v for all $v \in V$ and that induces no cycles.

* Will result in something that looks like a branching system (e.g. binary search tree)

* Called "forest of arborescences" or "part of a branching" (N&W).

Theorem 3.4. *Optimization over the intersection of two matroids with ground set N is polynomially solvable in $\mathcal{O}(|N|^3)$ (see N&W III.3.5, Prop. 4.8; see also the discussion after the Weighted Matroid Intersection Algorithm).*

Intersection of Three Matroids

Example 3.5. *(Intersection of two partition matroids and the graphic matroid)*

* Given digraph $G = (V, A)$ we have $\{\delta^-(v) : v \in V\}, \{\delta^+(v) : v \in V\}$ are two different partitions.

* Graphic matroid $\implies$ acyclic subsets of arcs

* Therefore, the following is an independence system:

$$\left(A, \left\{\hat{A} \subseteq A : \begin{smallmatrix} |\hat{A} \cap \delta^+(v)| \leq 1, \forall v \in V, \\ |\hat{A} \cap \delta^-(v)| \leq 1, \forall v \in V, \\ (V, \hat{A}) \text{ is acyclic} \end{smallmatrix}\right\}\right)$$

* The max cardinality of this independence system is equivalent to solving the H.C.P..

* Therefore, the max cardinality intersection of 3 matroid must be $\mathcal{NP}$-Hard.

MATCHING

Recall:

Max Weighted Matching

Given $G = (V, E), c \in \mathbb{R}^{|E|}$, the following MATCH-IP problem is

$$\begin{aligned} \max \sum_{e \in E} c_e x_e & \\ \text{s.t.} \sum_{e \in \delta(v)} x_e &\leq 1, \forall v \in V \\ c &\in \{0, 1\}^{|E|} \end{aligned}$$

Max Cardinality Matching

In the special case where $c = \mathbf{1}$, the matchings form an independence system

$$(E, \{S \subseteq E : |S \cap \delta(v)| \leq 1, \forall v \in V\})$$

that is not a matroid. To solve max cardinality matching, augmenting paths play a key role: for $M \subseteq E$,

* A path in G is alternating w.r.t. M if the edges in the path alternate between edges in M and edges not in M.

* A node v is exposed w.r.t. M if it is not met by an edge in M, i.e., $M \cap \delta(v) = \emptyset$ means v is exposed w.r.t. M.

* An alternating path is augmenting if both of its ends are exposed.

* If $P \subseteq E$ are the edges in an augmenting path w.r.t. a matching M, then

- $|P|$ is odd, say $|P| = 2k - 1$ for some $k \in \mathbb{Z}_+, k \geq 1$
- $|P \cap M| = k - 1$
- $|P \cap (E \backslash M)| = k$
- $M' = (M \backslash P) \cup (P \cap (E \backslash M))$ is a matching and $|M'| = |M| + 1$

Theorem 3.5. *A matching M either has max cardinality or there exists an augmenting path w.r.t. M, i.e., M is not of max cardinality if and only if there exists an augmenting path.*

Proof. ($\Longleftarrow$) By definition of augmenting path and the observations above.

($\Longrightarrow$) Suppose M is not of max cardinality. Then there exists M' a matching with $|M'| > |M|$. Consider $G' = (V, M \cup M')$. The max degree of any vertex G' is 2. Thus, any connected component of G' is either a path or a simple cycle. Any such cycle must have even cardinality, and consist of an equal number of edges from M and M'.

Hence, there must exist a connected component of G' that is a path with an odd number of edges with

$$|P \cap M'| = |P \cap M| + 1$$

where P is the set of edges in the path. Also the edges in P must alternate between M and M'. Therefore P is an augmenting path with respect to M.

Algorithm to Solve Max Cardinality Matching in a Bipartite Graph

Start with M any matching.
Orient all edges in M from right to left.
Orient all edges in M from left to right.
For each exposed node on the left, seek a (directed path) that ends at an exposed node on the right.
If one is found, it is an augmenting path w.r.t. M
Flip edges in the path to get new M with
cardinality one greater
Else
Done; M has max cardinality

This runs in time $\mathcal{O}(\#\text{ of nodes} \times \#\text{ of edges})$. What about non-bipartite graphs & weighted objectives? See N&W II.2.2. □

Proposition 3.2. *For a bipartite graph, every extreme point of the LP relaxation of MATCH-IP is integer. This is not true in general.*

Remark 3.2. The constraints

$$\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2}, \forall S \subseteq V, |S| \text{ is odd}$$

are clearly satisfied by all x that induce a matching. Adding these constraints to MATCH-IP gives an LP with all integer extreme points (NW III.2.4)

3.2 Integer Polyhedra

Definition 3.2. A nonempty polyhedron $P \subseteq \mathbb{R}^n$ is **integral** if each of its non-empty faces contains an integer point. It suffices to consider minimal faces, so if the polyhedron has extreme points, it suffices to require all extreme points are integer.

Example 3.6. The polyhedron $P = \{x \in \mathbb{R}^2 : x_1 + x_2 \leq 1, x_1, x_2 \geq 0\}$ is integral.

Proposition 3.3. *(N&W III.1.1, #1.3) TFAE:*

1. P is integral.

2. LP has an integral optimal solution $\forall c \in \mathbb{R}^n$ for which it has an optimal solution.

3. LP has an integral optimal solution $\forall c \in \mathbb{Z}^n$ for which it has an optimal solution.

4. z_{LP} is integral for all $c \in \mathbb{Z}^n$ for which LP has an optimal solution.

where LP is $z_{LP} = \max\{cx : x \in P\}$.

Proof. $(1 \implies 2)$ LP Theory, N&W I.4.4, #4.5

$(2 \implies 3)$ $\mathbb{Z}^n \subseteq \mathbb{R}^n$

$(3 \implies 4)$ $c \in \mathbb{Z}^n$ gives an optimal solution $\implies$ $\exists$ an optimal solution $x^* \in \mathbb{Z}^n$ such that $cx^* \in \mathbb{Z} \implies z_{LP} \in \mathbb{Z}$.

$(\neg 1 \implies \neg 4)$ Suppose that P is <u>not</u> integral. Say $\hat{x} \in P$ an extreme point of P with $\hat{x}_j \notin \mathbb{Z}$ for some $j \in \{1, ..., n\}$. Now $\exists c \in \mathbb{Z}^n$ such that $\hat{x}$ is the <u>unique</u> solution of LP (N&W I.4.4, Thm 4.6). Either $c\hat{x} \notin \mathbb{Z}$, so z_{LP} is not integral (as required) or $c\hat{x} \in \mathbb{Z}$. In the latter case, we can perturb c by a very small amount and $\hat{x}$ will still be optimal. Thus, there exists $q \in \mathbb{Z}$ sufficiently large that $\hat{x}$ is optimal for

$$\max\left\{\left(c + \frac{1}{q}e_j\right)x : x \in P\right\}$$

and hence for

$$\max\left\{(qc + e_j)\, x : x \in P\right\}$$

giving

$$z_{LP} = (qc + e_j)\hat{x} = \underbrace{qc\hat{x}}_{\in\mathbb{Z}} + \underbrace{x_j}_{\notin\mathbb{Z}} \notin \mathbb{Z}.$$

□

<u>Totally Dual Integral Matrices</u>

Definition 3.3. A system of linear inequalities $Ax \leq b$ is **totally dual integral** (TDI) if $\forall c \in \mathbb{Z}^n$ such that $z_{LP} = \max\{cx : Ax \leq b\}$ exists and the dual $\min\{yb : yA = c, y \geq 0\}$ has an integral optimal solution.

Corollary 3.1. *If $Ax \leq b$ is TDI and b is integral, then $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ is integral.*

Proof. Suppose that $Ax \leq b$ is TDI and let $c \in \mathbb{Z}^n$ be such that $z_{LP} = \max\{cx : Ax \leq b\}$ exists. Then $z_D = \min\{yb : yA = c\}$ has an integral optimal solution, $\hat{y}$ say. So $z_D = \hat{y}b \in \mathbb{Z}$ when b is integral but $z_{LP} = z_D$ and so if b is integral, then z_{LP} is integral, which is case 4 of the previous proposition. Hence P is integral. □

Remark 3.3. Note the converse is <u>not</u> true; there can be an integral P with b integer but the system $Ax \leq b$ is not TDI (N&W III.1.1, Example 1.2).

<u>Example of a TDI System</u>

Consider a complete bipartite graph $G = (V \cup W, E)$ with $E = V \times W$, edge weights b_{ij}, $i \in V, j \in W$, node weights $c_j \in V, d_j \in W$ and the problem

$$\max \sum_{j \in V} c_j x_j + \sum_{j \in W} d_j x_j$$
$$\text{s.t. } x_i + x_j \leq b_{ij}, \forall i \in V, j \in W.$$

Its LP dual is

$$\begin{aligned}
\min \; & \sum_{i\in V}\sum_{j\in W} b_{ij}y_{ij} \\
\text{s.t.} \; & \sum_{j\in W} y_{ij} = c_j, \forall i \in V \\
& \sum_{j\in V} y_{ij} = d_j, \forall i \in V \\
& y_{ij} \geq 0
\end{aligned}$$

which is a Transportation Problem, so if c, d are integer and the problem is feasible, then if it has an optimal solution, it has an integer optimal solution.

By the corollary if the b_{ij}'s are integer, then the set

$$\{x \in \mathbb{R}^n : x_i + x_j \leq b_{ij}, \forall i \in V, \forall j \in W\}$$

is integral.

Remark 3.4. $Ax \leq b$ TDI says nothing about integrality unless b is integer. TDI also depends on the scaling of the constraints.

Proposition 3.4. *(N&W III.1.1, #1.5) If $Ax \leq b$ is any irrational system then there exists $q \in \mathbb{Z}_+$ such that $\left(\frac{1}{q}\right) Ax \leq \left(\frac{1}{q}\right) b$ is TDI.*

Totally Unimodular Matrices

Definition 3.4. A $m \times n$ matrix A is **totally unimodular** (TU) if the determinant of every square submatrix of A is equal to 0, -1, and 1. So A is TU $\implies$ $A \in \{0, -1, 1\}$.

Example 3.7. Consider the matrix

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and consider the submatrix (3×3) from columns $1, 3, 4$ and rows $1, 2, 4$. Its determinant is

$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{vmatrix} = 2 \neq A$$

is not TU.

Remark 3.5. Recognizing a matrix is in TU is in co-$\mathcal{NP}$. Since "not TU" is in $\mathcal{NP}$, a "lucky guess" of some $k \times k$ submatrix with $\det$ not in 0,-1,1 will give a succinct certificate (calculating its determinant is polynomial time).

Recognizing a matrix is TU is less obvious, although it is in P (Seymour, 1980).

Proposition 3.5. *(N&W III.1.2 #2.1) TFAE*

(1) A is TU

(2) A^T is TU

(3) (A, I) is TU

(4) Matrix after deleting a unit row/column of A is TU

(5) Matrix after multiplying a row/column of A by -1 is TU

(6) Matrix after swapping two rows/columns of A is TU

(7) Matrix after duplicating rows/columns of A is TU

Proposition 3.6. *(Cramer's rule) If B is $k \times k$ then if $\det(B) \neq 0$ we have*

$$B^{-1} = \frac{1}{\det(B)} B^*$$

where B^* is the adjoint, which has elements ± 1 times determinants of $(k-1) \times (k-1)$ submatrices of B.

Proposition 3.7. *(N&W III.1.2, #2.2) If A is TU then the polyhedron $\{x \in \mathbb{R}^n : Ax \leq b\}$ is integral for all $b \in \mathbb{Z}^n$, for which it is non-empty.*

Proof. Suppose that A is TU and $\hat{x}$ is an extreme point of $\{x \in \mathbb{R}^n : Ax \leq b\}$ with b integer. Then $\exists (B, \hat{b})$ a submatrix of (A, b) such that $\hat{x} = B^{-1}\hat{b}$. But A is TU so $\det(B) \in \{1, -1\}$. Thus, by Cramer's rule,

$$\hat{x} = \frac{1}{\det(B)} B^* \hat{b} = \pm B^* \hat{b}$$

and B^* entries are $\pm$ determinants of square submatrices of A which implies that its entries are $\{0, 1, -1\}$. Since $\hat{b}$ is also integral, then $\hat{x}$ is integral as well. □

Theorem 3.6. *(N&W III.1.2, #2.5) Let $P(b) = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$. $P(b)$ is integral for all $b \in \mathbb{Z}^m$ for which it is non-empty if and only if A is TU.*

Proof. ($\Longleftarrow$) From last proposition, not that $\begin{pmatrix} A \\ -I \end{pmatrix}$ has full column rank.

($\Longrightarrow$) Read N&W. □

Theorem 3.7. *(N&W III.1.2., #2.7) TFAE*

(1) A is TU (A is $n \times m$)

(2) For all $J \subseteq \{1, ..., n\} := N$, there exists a partition of J into sets J_1, J_2 with $J_1 \cup J_2 = J$, $J_1 \cap J_2 = \emptyset$ such that

$$\left| \sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} \right| \leq 1, \text{ for } i = 1, ..., m.$$

Note: one of J_1 and J_2 may be empty.

Proof. See N&W. □

Definition 3.5. For $A = (a_{ij})$ an $m \times n$ and 0,± 1 matrix with at most two nonzero elements in each column, we define two conditions as follows:

Condition 1: For all columns j with two non-zeros, $\sum_{i=1}^m a_{ij} = 0$.

Condition 2: There exists Q_1, Q_2 a partition of $\{1, ..., m\}$ such that for all columns with two nonzeros, say $a_{ij} \neq 0$ and $a_{i'j} \neq 0$, $i \neq i'$, then

$$\begin{aligned} \text{sgn}(a_{ij}) = \text{sgn}(a_{ij}) &\implies (i \in Q_1 \text{ and } i' \in Q_2) \text{ or } (i \in Q_2 \text{ and } i' \in Q_1) \\ \text{sgn}(a_{ij}) \neq \text{sgn}(a_{ij}) &\implies (i, i' \in Q_2) \text{ or } (i, i' \in Q_1) \end{aligned}$$

Corollary 3.2. *(N&W III.1.2 Prop 2.6 & Corollary 2.8) For A a $0, \pm 1$ matrix with at most two nonzero per column,*

$$\textit{Condition 1} \implies \textit{Condition 2} \implies A \textit{ is TU}$$

Proof. ($1 \implies 2$) Condition 1 implies that every column with two nonzeros must have the nonzeros being of opposite sign. Thus we may take $Q_1 = \{1, ..., m\}, Q_2 = \emptyset$.

(2 $\implies$ TU) For any $I \subseteq \{1, ..., m\}$, $I \cap Q_1$ and $I \cap Q_2$ partitions I. Now if two nonzeros in column j, say $a_{ij}, a_{i'j}$ with $i \neq i'$ have $i, i' \in I$ then

$$\sum_{k \in I \cap Q_1} a_{kj} - \sum_{k \in I \cap Q_2} a_{kj} = \begin{cases} \pm a_{ij} \mp a_{i'j} & \text{if } \operatorname{sgn}(a_{ij}) = \operatorname{sgn}(a_{i'j}) \\ \pm(a_{ij} + a_{i'j}) & \text{if } \operatorname{sgn}(a_{ij}) \neq \operatorname{sgn}(a_{i'j}) \end{cases}$$
$$= 0.$$

If only one nonzero in column j, say a_{ij}, then

$$\sum_{k \in I \cap Q_1} a_{kj} - \sum_{k \in I \cap Q_2} a_{kj} = \pm a_{ij} = \pm 1$$

and by a previous theorem, A^T is TU $\implies$ A is TU.

(TU $\implies$ 2) Exercise. $\square$

Classes of TU matrices

1. *Node-arc incidence matrix of a digraph* with $G = (V, E), |V| = m, |E| = n$ where explicitly, $A = (a_{ij})$ where

$$a_{ij} = \begin{cases} -1 & \text{if the } j^{th} \text{ arc has tail at } i \\ 1 & \text{if the } j^{th} \text{ arc has tail at } i \text{ .} \\ 0 & \text{otherwise} \end{cases}$$

Note, we call the node where the arrow head is at, the head of the arc and the bottom of the arrow, the tail.

2. *Node-edge incidence matrix of a bipartite graph* with $G = (V \cup W, E)$, $|V| + |W| = m$, $|E| = n$ where

$$a_{ij} = \begin{cases} 1 & \text{if node } i \text{ is in arc } j \\ 0 & \text{otherwise} \end{cases}$$

It turns out that this class satisfies Condition 2 with the partition as the natural partition of the bipartite graph.

3. *Interval matrices* which are $0, 1$ matrices with all 1's in any column consecutive. That is, $a_{ij} = a_{kj} = 1$ for $i < k \implies a_{rj} = 1$ for $r \in \{i + 1, ..., k - 1\}$.

For any $Q \subseteq \{1, ..., m\}$ where $A \in \mathbb{R}^{m \times n}$, take B the submatrix of A consisting of the rows indexed by Q. Let Q_1, Q_2 be a partition of Q formed by alternating between them, e.g. if $Q = \{i_1, ..., i_k\}$ where $i_1 < i_2 < ... < i_k$ then let $Q_1 = \{i_1, i_3, ...\}$, $Q_2 = \{i_2, i_4, ...\}$. Then in general

$$\left(\sum_{i \in Q_1} a_{ij} - \sum_{i \in Q_2} a_{ij} \right)_j$$

is a vector with entries in $\{0, -1, 1\}$. To see why this works in general, if there are an even number of 1's, then the sum for that position is 0, otherwise it is ± 1.

4. *Network matrices*: given $G = (V, E)$ a directed graph, $m = |V|, n = |E|$ and given a directed tree – ignoring the arc directions, it is a spanning tree – (V, T) with $|T| = m - 1$.

The matrix for this tree is described by examining the arc-to-arcs relationships. In particular

$$a_{ij} = \begin{cases} 1 & \begin{matrix} i \text{ is forward in the unique path in } (V,T) \\ \text{from the tail of } i \text{ to its head} \end{matrix} \\ -1 & \begin{matrix} i \text{ is backward in the unique path in } (V,T) \\ \text{from the tail of } i \text{ to its head} \end{matrix} \\ 0 & \text{otherwise} \end{cases}$$

The proof that network matrices are TU is in N&W III.1.3, #3.1. It also turns out that interval matrices are network matrices (up to appending identity matrices) when the tree only moves in one direction (N&W III.1.3, #3.3).

Remark 3.6. If A is TU with at most 2 nonzeros per column then A is a network matrix (N&W III.1.3, #3.1).

5. *Two special* 5×5 *matrices:*

$$\begin{bmatrix} 1 & -1 & 0 & 0 & -1 \\ -1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Exercise: Show both are TU.

Theorem 3.8. *Every TU matrix that is not a network matrix or one of the two special* 5×5 *matrices above, can be constructed from them using systematic rules as in* N&W III.1.2, Prop. 2.1 and Prop. 2.11.

Theorem 3.9. *(Seymour, 1980) Also recognizing a TU matrix can be done in time* $\mathcal{O}((n+m)^3)$.

BALANCED & PERFECT MATRIX SET PACKING/COVERING

Definition 3.6. For a 0-1 matrix in $\mathbb{R}^{m \times n}$ and

$$\mathcal{M}_k = \left\{ D \in \{0,1\}^{k \times k} : \begin{matrix} \text{every row sum of } D \text{ is } 2, \\ \text{every column sum of } D \text{ is } 2, \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ is not a submatrix} \end{matrix} \right\}$$

the matrix$A$ is **balanced** if it has no submatrix in $\mathcal{M}_k$ for $k \geq 3$, k odd.

Theorem 3.10. *(N&W III.1.4, #4.3) If a 0-1 matrix is TU then it is balanced.*

Theorem 3.11. *(N&W III.1.4, #4.13) If* A *is a 0-1 matrix with no zero rows or columns,* $P = \{x \in \mathbb{R}^n_+ : Ax \leq 1\}$ *is a fractional packing,* $Q = \{x \in \mathbb{R}^n_+ : Ax \geq 1\}$ *is a fractional covering, then TFAE:*

1. A *is balanced.*
2. P *is integral and so is the polyhedron formed by dropping an rows of* $(A, 1)$.
3. Q *is integral and so is the polyhedron formed by dropping an rows of* $(A, 1)$.

There is a broader class of matrices for which set packing is integral.

Example 3.8. For a node-edge graph with size 3 cliques $C_1 = \{1,2,3\}, C_2 = \{1,2,4\}, C_3 = \{1,3,5\}, C_4 = \{2,3,6\}$, the clique-node incidence matrix is

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

As an exercise, show that A is not balanced and that $P = \{x \in \mathbb{R}^n_+ : Ax \leq 1\}$ is integral. In fact, A is **perfect**.

It turns out that the class of matrices form the relation

$$\text{TU 0-1} \subset \text{Balanced} \subset \text{Perfect}$$

with (integral $Ax \leq b$ for all integral b)$\in$TU 0-1, (set covering, set packing, set partitioning, are all integral)$\in$Balanced, (set packing)$\in$Perfect.

4 Polyhedral Theory

See N&W I.4 for theorems.

- Affine independence of vectors: the set of vectors $x^1, ..., x^k \in \mathbb{R}^n$ are called affinely independent $\iff$ $x^2 - x^1, ..., x^k - x^1 \in \mathbb{R}^n$ are linearly independent.

- The maximum number of affine independent vectors in $\mathbb{R}^n$ is $(n+1)$

- A polyhedron is of dimension k, written as $\dim(P) = k$ if the maximum number of affinely independent points in P is $(k+1)$.

- We use the notation $P = \{x \in \mathbb{R}^n : Ax \le b\}$, $M = \{1, 2, ..., m\}$, $M^= = \{i \in M : a^i x = b_i, \forall x \in P\}$, $M^\le = \{i \in M : a^i x < b_i, \exists x \in P\}$ where a^i is the i^{th} row of A.

Definition 4.1. $x \in P$ is an **interior point** of P is $a^i x < b_i$, for all $i \in M$.

Definition 4.2. $x \in P$ is an **inner point** of P is $a^i x < b_i$, for all $i \in M^\le$.

Proposition 4.1. *(Polyhedral Rank-Nullity) If $P \subseteq \mathbb{R}^n$ and $P \neq \emptyset$ then* $\dim(P) + \underbrace{rank(A^= b^=)}_{=rank(A^=) \text{ if } P \neq \emptyset} = n.$

Example 4.1. Consider the polyhedron

$$P = \left\{ x \in \mathbb{R}^3 : \begin{array}{c} x_1 + x_2 + x_3 \le 1 \\ -x_1 - x_2 - x_3 \le -1 \\ x_1 + x_3 \le 1 \\ -x_1 \le 0 \\ -x_2 \le 0 \\ x_3 \le 2 \\ x_1 + x_2 + 2x_3 \le 2 \end{array} \right\}.$$

We have $M = \{1, 2, ..., 7\}$, $M^= = \{1, 2\}$, $M^\le = \{3, ..., 7\}$. What is $\dim(P)$? From the first two equations, which hold with equality,

$$\text{rank}(A^= b^=) \ge \text{rank}\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \right) \ge 1$$

<u>Exercise</u>: Find 3 affinely independent points in P to imply that $\dim(P) \ge 2$. By the rank-nullity theorem, $\dim(P) = 3 - 1 \le 2$ which implies $\dim(P) = 2$.

Definition 4.3. $\pi x \le \pi_0$ or (π, π_0) is **valid** for P if $\pi x \le \pi_0$ for all $x \in P$.

Definition 4.4. If (π, π_0) is valid for P then $F = \{x \in P : \pi x \le \pi_0\}$ is a **face** of P. F is **proper** if $F \neq \emptyset$ and $F \neq P$. We say F is **represented** by (π, π_0) if $F = \{x \in P : \pi x = \pi_0\}$.

Proposition 4.2. *If $P = \{x \in \mathbb{R}^n : Ax \le b\}$ with equality set $M^= \subseteq M$ and F is a nonempty face of P then:*

** F is a polyhedron*

** $M_F^= \subseteq M$ and*

** $F = \{x \in \mathbb{R}^n : a^i x = b_i, a^j x \le b_j, i \in M_F^=, j \in M_F^\le\}$*

where $M_F^= \supseteq M^=$ and $M_F^\le = M \backslash M_F^=$. Furthermore, the number of distinct faces of P is finite.

Definition 4.5. A face F_1 of P is a **facet** if $\dim(F) = \dim(P) - 1$.

Proposition 4.3. *If F is a facet of P then there exists some (one) inequality $a^k x \le b_k$ for $k \in M^\le$ representing F.*

Example 4.2. (previous example cont.)

Q. Is $-x_1 - x_2 + x_2 \le 1$ valid for P?

A. Since $(3) + 2 \times (4) + (5)$ gives the above, then yes.

Q. Does it induce a proper face of P?

A. Yes, since $(0, 0, 1)^T \in F = \{x \in P : -x_1 - x_2 + x_3 = 1\}$ so $F \neq \emptyset$ and $(1, 0, 0) \in P \backslash F$ so $F \neq P$.

Q. What is $\dim(F)$?

A. Note $x_3 \le 1$ is valid for P. So for $x \in F$, $x_3 = 1 + x_1 + x_2$ with $x_1, x_2 \ge 0 \implies x_1 = x_2 = 0$. So

$$\text{rank}\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \right) = 3$$

and hence $\dim(F) =\leq 3-3=0$. Thus, $F=\{x \in R : (4)\&(5) \text{ at equality}\}$.

Exercise: Answer the same questions for $2x_1 - 7x_2 + 2x_3 \leq 2$.

Theorem 4.1. *(A Minimal Description of P) A full dimensional polyhedron P has a unique (to within scalar multiplication) minimal representation by a finite set of linear inequalities. In particular, for each facet F_i of P, there is an inequality $a^i x \leq b_i$ (unique up to scalar multiplication) representing F_i and*

$$P=\{x \in \mathbb{R}^n : a^i x \leq b_i, i=1,2,...,t\}$$

where t is the number of distinct faces of P.

Furthermore, if $\dim(P)=n-k$ with $k>0$, then

$$P=\{x \in \mathbb{R}^n : a^i x = b_i, i=1,...,k, a^j x \leq b_j, j=k+1,...,k+t\}$$

where (a^i, b_i) for $i=1,2,...,k$ are a maximal set of linearly independent rows of $(A^=, b^=)$ and (a^i, b^i) for $i \in \{k+1,...,k+t\}$ is any inequality from the equivalence class of inequalities representing F_i.

Theorem 4.2. *(N&W Thm 3.6) Let $(A^=, b^=)$ be the equality set for $P \subseteq \mathbb{R}^n$ and let $F=\{x \in P : \pi x = \pi_0\}$ be a proper face of P. TFAE:*

1. F is a facet

2. If $\lambda x = \lambda_0$ for all $x \in F$, then

$$(\lambda, \lambda_0) = (\alpha \pi + uA^=, \alpha \pi_0 + ub^=)$$

for some $\alpha \in \mathbb{R}$ and $u \in \mathbb{R}^{|M^=|}$.

Theorem 4.3. *(Minkowski's Theorem) If $P \neq \emptyset$ and rank$(A)=n$ then $P=Q$ where*

$$Q=\left\{x \in \mathbb{R}^n : x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \mu r^j, \sum_{k \in K} \lambda_k = 1, \lambda \geq 0, \mu \geq 0\right\}$$

where $\{x^k\}_{k \in K}$ is the set of extreme points of P and $\{r^j\}_{k \in J}$ is the set of extreme rays.

Note 3. It is assumed that you have a working knowledge of:

- Projection of polyhedra
- Farkas' Lemma

4.1 IP and LP Ties

(N&W I.4.6) IP can, in some sense, be reduced to LP. In particular conv(S) where

$$S = P \cap \mathbb{Z}^n, P=\{x \in \mathbb{R}^n_+ : Ax \leq b\}$$

and $(A,b) \in \mathbb{Q}^{m \times (n+1)}$ is a rational polyhedron.

Theorem 4.4. *If P, S are as above then:*

(1) There exists a finite set of points $\{q^l\}_{l \in L}$ of S and a finite set of rays $\{r^j\}_{j \in J}$ of P such that

$$S=\left\{x \in \mathbb{R}^n_+ : {x=\sum_{l \in L} \alpha_l q^l + \sum_{j \in J} \beta_j r^j \atop \sum_{l \in L} \alpha_l = 1, \alpha \in \mathbb{Z}^{|L|}_+, \beta \in \mathbb{Z}^{|J|}_+}\right\}$$

(2) If P is a cone, $(b=0)$ then there exists a finite set of rays $\{v^h\}_{h \in H}$ such that

$$S=\left\{x \in \mathbb{R}^n_+ : x = \sum_{h \in H} \gamma_h v^h, \gamma \in \mathbb{Z}^{|H|}_+\right\}.$$

Proof. (1) Let $\{x^k\}_{k\in K}$ be the extreme points of P and $\{r^j\}_{j\in J}$ be its extreme rays. P is rational implies that all extreme points and rays are rational and thus we may assume w.l.o.g. that $r^j \in \mathbb{Z}^n_+$ for $j \in J$. By Minkowski's Theorem,

$$P = \left\{ x \in \mathbb{R}^m_+ : \substack{x=\sum_k \lambda_k x^k + \sum_j \mu_j r^j \\ 1^T\lambda=1, \lambda\geq 0, \mu \geq 0} \right\}.$$

Let

$$Q = \left\{ x \in \mathbb{Z}^n_+ : \substack{x=\sum_k \lambda_k x^k + \sum_j \mu_j r^j \\ 1^T\lambda=1, \lambda\geq 0 \\ 0\leq \mu_j < 1, \forall j \in J} \right\}$$

which is a finite set, say $Q = \{q^l\}_{l\in L}$ for some index set L. Note that $Q \subseteq S$. Now observe $x^i \in S$ if and only if $x^i \in \mathbb{Z}^n$ and $x^i \in P$, i.e.

$$x^i = \sum_k \lambda_k x^k + \sum_j (\mu_j - \lfloor \mu_j \rfloor) + \sum_j \lfloor \mu_j \rfloor r^j$$

where $1^T\lambda = 1, \lambda \geq 0, \mu \geq 0$. Therefore $x^i \in S \iff x^i = q^{l(i)} + \sum_j \lfloor \mu_j \rfloor r^j$ for some $l(i) \in L, \mu \geq 0$. The proof concludes when we let $\beta_j := \lfloor \mu_j \rfloor$.

(2) $q \in S$ implies $\gamma q \in S$ for all $\gamma \in \mathbb{Z}_+$ so we can take

$$\{v^h\}_{h\in H} = \{q^l : l \in L\} \cup \{r^j : j \in J\}$$

from (1). □

Example 4.3. Consider $P = \{x \in \mathbb{R}^2_+ : 2x_1 + 3x_2 \geq 7, 2x_1 - 2x_2 \geq -3\}$ and $S = P \cap \mathbb{Z}^2$. The set P has extreme rays $(1, 0)$ and $(1, 1)$.

Exercise: Determine Q.

Theorem 4.5. *If S, P are as above, then conv(S) is a rational polyhedron.*

Proof. From the proof of the previous theorem, we have

$$x^i = q^{l(i)} + \sum_j \beta^i_j r^j$$

where $q^{l(i)} \in Q$ and $\beta^i_j \in \mathbb{Z}_+$ for all $j \in J$ and $x^i \in S$. So any point $x \in \text{conv}(S)$ can be expressed as

$$\begin{aligned}
x &= \sum_{i\in I} \gamma_i x^i \\
&= \sum_{i\in I} \gamma_i (q^{l(i)} + \sum_j \beta^i_j r^j) \\
&= \sum_{i\in I} \gamma_i q^{l(i)} + \sum_{i\in I}\sum_j \beta^i_j r^j \\
&= \sum_l \underbrace{\left(\sum_{\substack{i\in I \\ l(i)=l}} \gamma_i \right)}_{=:\alpha_l} q^l + \sum_j \underbrace{\left(\sum_{i\in I} \gamma_i \beta^i_j \right)}_{=:\beta_j} r^j \\
&= \sum_l \alpha_l q^l + \sum_j \beta_j r^j
\end{aligned}$$

for some finite set I with $\{x^i\}_{i\in I} \subseteq S$, $1^T\gamma = 1$, $\gamma \geq 0$ and where $\sum_{l\in L} \alpha_l = \sum_{i\in I} \gamma_i = 1$. By Weyl's Theorem, conv(S) is a rational polyhedron. □

This result easily extends to mixed integer sets. The proof of the above theorem also shows that if $S \neq \emptyset$ then the extreme rays of conv(S) coincide with those of P.

We now observe that to solve an integer program

$$(IP)\ \max\{cx : x \in S\}$$

we could instead solve

$$(CIP)\ \max\{cx : x \in \text{conv}(S)\}$$

which is an LP. Thus (IP) inherits useful LP properties: it is unbounded, infeasible, or it has an optimal solution. If it has an optimal solution that is an extreme point of conv(S).

Theorem 4.6. *Given S, P as above, $S \neq \emptyset$ and any $c \in \mathbb{R}^n$*

(a) (IP) is unbounded $\iff (CIP)$ is unbounded.

(b) (CIP) has an optimal solution $\implies \exists x^*$ which is an optimal solution of (CIP) that is optimal for (IP) and

(c) x^* is optimal for $(IP) \implies x^*$ is optimal for (CIP).

Exercise:

(i) $x \in \text{conv}(S), x \in \mathbb{Z}^n_+ \implies x \in S$

(ii) x is an extreme point of conv$(S) \implies x \in S$, where an extreme point x is when $x \neq \frac{1}{2}(x^1 + x^2)$ for any $x^1, x^2 \in$ conv$(S), x^1 \neq x^2$.

Proof. First note conv$(S) \supseteq S$ and so $z_{CIP} \geq z_{IP}$. (1)

(a) Hence if (IP) is unbounded then (CIP) is unbounded.

Now if (CIP) is unbounded, then $\exists r$ a ray of conv(S) such that $cr > 0$. Since conv(S) is a rational polyhedron, we may take $r \in \mathbb{Z}^n_+$. Now take x an extreme point of conv$(S) \implies x \in S \implies x \in \mathbb{Z}^n_+$. Then for all $\gamma \in \mathbb{Z}_+$, $x + \gamma r \in$ conv(S) with $x + \gamma r \in \mathbb{Z}^n$. Therefore, $x \in S$ and (IP) is unbounded.

(b) Take x^* to be an extreme point optimum of (CIP) (since (CIP) is an LP with feasible set in $\mathbb{R}^n$ implies there exists an extreme point). Then $x^* \in S$, so $z_{IP} \geq cx^* = z_{IP}$. By (1) it must be $z_{IP} = z_{CIP} = cx^*$ so x^* is optimal for IP as well.

(c) Exercise. □

Proposition 4.4. *If $\pi x \leq \pi_0$ is valid for S then it is also valid for conv(S).*

Proof. Exercise. □

Lemma 4.1. *Suppose $y^1, ..., y^k \in \mathbb{R}^n$ are affinely independent and*

$$y^1 = \sum_{j \in J} \lambda_j x^j$$

for some $\{\lambda_j\}_{j \in J} \subseteq \mathbb{R}$ and $\{x^j\}_{j \in J} \subseteq \mathbb{R}^n$ with $\mathbf{1}\lambda = 1, \lambda_j > 0, \forall j \in J$. Then $\exists j^* \in J$ such that $x^{j^*}, y^2, ..., y^k$ are affinely independent.

Proof. Note that $y^2, ..., y^k$ must be affinely independent. For the sake of contradiction, assume $\forall j \in J$, $x^j, y^2, ..., y^k$ are not affinely independent. Then $\exists \{\alpha^j_i\}_{i=1}^k$ not all zero such that

$$\alpha^j_1 x^j + \sum_{i=2}^k \alpha^j_i y^i = 0, \sum_{i=1}^k \alpha^j_i = 0.$$

In fact, $\alpha^j_1 \neq 0$ otherwise $y^2, ..., y^k$ are not affinely independent. Now let

$$\beta_1 = 1, \beta_i = \frac{1}{\alpha^j_1} \sum_{i=2}^k \lambda_i \alpha^j_i.$$

Then $\beta \neq 0, \mathbf{1}\beta = 0$ (exercise) but $\sum_{i=1}^k \beta_i y^i = 0$ (exercise) which leads to a contradiction. □

Proposition 4.5. *If $\pi x \leq \pi_0$ defines a face of conv(S) of dimension $k - 1$, then there are k affinely independent points $x^1, ..., x^k \in S$ such that $\pi x^i = \pi_0$ for all $i = 1, 2, ..., k$.*

Proof. By the hypothesis, choose points $\bar{x}^1, ..., \bar{x}^k \in \text{conv}(S)$ such that $\pi\bar{x}^i = \pi_0$ for all $i = 1, ..., k$ so as to maximize the number of them which are in S. If they are all in S then we are done. Otherwise, without loss of generality, suppose that $\bar{x}^1 \notin S$. Then, $\bar{x}^1 = \sum_{j\in J} \lambda_j \hat{x}^j$, for some $\{\hat{x}^j\}_{j\in J} \subseteq S$ and $\{\lambda_j\}_{j\in J} \subseteq \mathbb{R}_+$ with $\mathbf{1}\lambda = 1$.

Now $\pi x \leq \pi_0$ is valid for $\text{conv}(S)$ and hence for S this implies that $\pi\hat{x}^j \leq \pi_0$ for all $j \in J$. Also

$$\begin{aligned} \pi\bar{x}^1 = \pi_0 &\implies \sum_{j\in J} \lambda_j(\pi\hat{x}^j) = \pi_0 \\ &\implies \pi\hat{x}^j = \pi_0, \forall j \in J. \end{aligned}$$

By the lemma, there exists $j^* \in J$ such that $\hat{x}^{j^*}, \bar{x}^2, ..., \bar{x}^k$ is affinely independent and has one more point in S than did $\bar{x}^1, \bar{x}^2, ..., \bar{x}^k$ which is impossible. □

Example 4.4. Consider node packing: $G = (V, E)$ with

$$S = \{x \in \{0,1\}^n : x_i + x_j \leq 1, \forall\{i,j\} \in E\}$$

where $n = |V|$. What is $\dim(\text{conv}(S))$? We claim that it is n.

Proof. The basis vectors $e_i \in S$ for $i = 1, 2, ..., n$ as well as $0 \in S$. So if $e_0 = 0$ then $e_0, e_1, ..., e_n \in S$ are affinely independent. □

Remark 4.1. In the example above, note that $\frac{1}{2} \cdot \mathbf{1} \in P$, the LP relaxation of the node packing problem S. For a formulation where

$$\begin{aligned} x_6 + x_7 + x_8 &\leq 1 \\ x_1 + x_2 + x_3 + x_4 &\leq 1 \end{aligned}$$

are both valid for S, we will see that $\frac{1}{2} \cdot \mathbf{1} \notin \text{conv}(S)$.

Proposition 4.6. *Let $C \subseteq V$ be a clique so $\{i,j\} \in E$ for all $i, j \in C$ and $i \neq j$. The clique constraint is*

$$\sum_{i\in C} x_i \leq 1 \qquad (*)$$

is valid for S.

Proof. Exercise. □

Claim 4.1. Provided C is maximal then $(*)$ defines a facet of $\text{conv}(S)$.

Proof. Let $F = \left\{x \in \text{conv}(S) : \sum_{i\in C} x_i = 1\right\}$. We need to find n affinely independent points in F. Note that $e_i \in F$ for all $i \in C$. For each $j \in V \backslash C$, $\exists v(j) \in C$ with $\{j, v(j)\} \notin E$ since C is maximal. Therefore $e_j + e_{v(j)} \in F$ for all $j \in V \backslash C$.

Without loss of generality, take

$$C = \{1, ..., |C|\}, V \backslash C = \{|C| + 1, ..., |V|\}$$

where we have the full rank matrix

$$M = \begin{bmatrix} I & 0 \\ A & I \end{bmatrix}$$

where A is some permutation of I. *Alternatively*, suppose that

$$\sum_{i\in C} \alpha_i e_i + \sum_{j\in V\backslash C} \beta_j(e_j + e_{v(j)}) = 0$$

for some α, β with $\sum_{i\in C} \alpha_i + \sum_{j\in V\backslash C} \beta_j = 0$. Then

$$LHS = \sum_{i\in C} \left(\alpha_i + \sum_{\substack{j\in V\backslash C \\ v(j)=i}} \beta_j\right) e_i + \sum_{j\in V\backslash C} \beta_j e_j = 0$$

which implies that

$$\alpha_i + \sum_{\substack{j \in V \setminus C \\ v(j)=i}} \beta_j = 0, \forall i, \beta_j = 0, \forall j$$
$$\implies \alpha_i = 0, \forall i.$$

□

Example 4.5. Recall the node-packing polytope conv(S) where

$$S = \{\{0,1\}^n : x_i + x_j \leq 1, \forall \{i,j\} \in E\}$$

for $G = (V, E)$ an undirected graph with $|V| = n$. Recall that $\dim(\text{conv}(S)) = n$ and the clique constraint: $\forall C \subseteq V$ such that C induces a clique in G,

$$\sum_{i \in C} x_i \leq 1 \qquad (*)$$

is valid for conv(S).

Note 4. Node-packing arises whenever a MIP has binary variables, via preprocessing.

Example 4.6. Suppose a MIP includes the constraint

$$13x_1 + 10x_2 + 9x_3 + 7x_4 \underbrace{+ \ldots}_{\geq 0} \leq 15$$

where $x_1, ..., x_4 \in \{0,1\}$. We can derive a **conflict graph** where the nodes are the variables and their complements, and edges are two variables (its ends) where they cannot both be 1. In the above constraint, we have a fully connected conflict graph for the variables.

An **implication constraint** (two variables) for the above is $x_2 + x_3 \leq 1$ while a **clique constraint** is $x_1 + x_2 + x_3 + x_4 \leq 1$.

Proposition 4.7. *(*) defines a facet of conv(S) for S the node-packing polytope for $G = (V, E)$ and C a maximal clique of G.*

Proof. (using N&W Thm. 3.6) Let $F = \{x \in \text{conv}(S) : \sum_{i \in C} x_i = 1\}$ and suppose $\lambda x = \lambda_0$ for all $x \in F$. Then

$$(\lambda, \lambda_0) = \alpha(\pi, \pi_0), \pi = \sum_{i \in C} e_i, \pi_0 = 1.$$

Note that $\alpha = \lambda_0$. Now $e_i \in F$ for all $i \in C$ and thus

$$\lambda e_i = \lambda_0 \implies \lambda_i = \lambda_0, \forall i \in C.$$

Also, for all $j \in V \setminus C$, there exists $i(j) \in C$ such that $\{i(j), j\} \notin E$ since C is maximal and which implies $e_j + e_{i(j)} \in F$ and thus

$$\lambda(e_j + e_{i(j)}) = \lambda_0 \implies \lambda_j + \lambda_{i(j)} = \lambda_0 \implies \lambda_j + \lambda_0 = \lambda_0 \implies \lambda_j = 0.$$

□

5 Cutting Planes and Separation

Consider the LP

$$\begin{aligned} &\max cx \\ &\text{s.t } Ax \leq b \\ &\quad d_r x \leq g_r, \forall r \in \Omega \end{aligned}$$

where Ω is the index set of a possibly exponentially large set of constraints (cutting planes). Can we still solve this LP (efficiently)? How?

A General Cutting Plane Algorithm for LP

Let $T = \emptyset$
While (not done)
Solve $\max\{cx : Ax \le b, d_r x \le g_r, \forall r \in T\}$ to get x^*
Find $r \in \Omega$ such that $d_r x^* > g_r$ or show none exists.
If none exists, set done = TRUE
Else set $T := T \cup \{r\}$
End While

The Separation Problem

Given x^* and a class of constraints $d_r x \le g_r$ for all $r \in \Omega$, find $r \in \Omega$ such that $d_r x^* > g_r$ or show none exists.

The Equivalence of Optimization & Separation

(Grotschel, Lovatz, & Schrijver, 1981) There exists a polynomial time algorithm for solving the separation problem $\iff$ There exists a polynomial time algorithm for solving the LP.

Example 5.1. (TSP undirected LP) Consider the problem

$$\begin{aligned}
\min & \sum_{e \in E} c_e x_e \\
\text{s.t} & \sum_{e \in \delta(v)} x_e = 2, \forall v \in V \\
& 0 \le x_e \le 1, \forall e \in E \\
& \sum_{u \in S} \sum_{v \in V \setminus S} x_{\{u,v\}} \ge 2, \forall S \in \Omega
\end{aligned}$$

where $\Omega := \{S \subseteq V : 2 \le |S| \le |V| - 2\}$ (See N&W II.6 Ex. 3.1. for a graphical example).

Let us solve the separation problem for the N&W example for the Ω constraint, i.e., find $S \in \Omega$ such that

$$\sum_{u \in S} \sum_{v \in V \setminus S} x^*_{\{u,v\}} < 2$$

or show none such S exists. Solution is $S = \{1, 2, 3, 7\}$ since

$$x^*_{15} + x^*_{24} + x^*_{36} = 1 \not\ge 2.$$

Thus, we add the constraint $x_{15} + x_{24} + x_{36} \ge 2$. The optimal solution of this tighter formulation will give the optimal tour.

For TSP, the **Subtour Elimination Constraint** (SEC) separation problem

$$z_{SEP} = \min \left\{ \sum_{u \in S} \sum_{v \in V \setminus S} x^*_{\{u,v\}} : S \subset V, S \neq \emptyset \right\}$$

which can be solved as min cut in a graph, equivalent to $\frac{1}{2}|V|(|V| - 1)$ max flow problems, or by a special combinatorial algorithm (Stoer & Wagner) in $\mathcal{O}(|E||V| + |V|^2 \log |V|)$ and thus TSP LP can be solved in polynomial time.

{Start Scribe}

Remark 5.1. For any subset of nodes H, we have the following valid equation:

$$\sum_{e \in \delta(H)} x_e = 2|H| - 2 \sum_{e \in E(H)} x_e \qquad (*)$$

Proposition 5.1. *Given $H \subseteq V$, $T \subseteq \delta(H)$, the **comb inequality** is*

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

is valid for the TSP.

Proof. Observe that

$$\sum_{e \in T} x_e \leq \begin{cases} |T|, & |T| \text{ is even} \\ |T| - 1 + \sum_{e \in \delta(H) \setminus T} x_e, & |T| \text{ is odd} \end{cases}$$

and hence

$$\begin{aligned}
&\sum_{e \in T} x_e \leq 2 \left\lfloor \frac{|T|}{2} \right\rfloor + \sum_{e \in \delta(H) \setminus T} x_e \\
\implies &\frac{1}{2} \sum_{e \in T} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor + \frac{1}{2} \sum_{e \in \delta(H) \setminus T} x_e \\
\implies &\sum_{e \in T} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor + \frac{1}{2} \sum_{e \in \delta(H)} x_e \\
\implies &\sum_{e \in T} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor + \frac{1}{2} \left(2 \left[|H| - \sum_{e \in E(H)} x_e \right] \right) \\
\implies &\sum_{e \in T} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor + |H| - \sum_{e \in E(H)} x_e
\end{aligned}$$

□

Exercise 5.1. What is the separation problem for the comb inequalities? Is it hard? Easy?

Example 5.2. In the example fractional tour in Figure 1 [put this somewhere nice], we have

$$\begin{aligned}
\sum_{e \in T} x_e + \sum_{e \in E(H)} x_e = \frac{9}{2} \\
|H| = 3 \\
\left\lfloor \frac{|T|}{2} \right\rfloor = 3 + \left\lfloor \frac{|T|}{2} \right\rfloor = 4
\end{aligned}$$

which violates the comb inequality.

Example 5.3. [Lot Sizing and (l, S)-inequalities]

Given variables

$$\begin{cases} x_t &= \text{quantity made in period } t \\ y_t &= \begin{cases} 1, & \text{if } x_t > 0 \\ 0, & \text{if } x_t = 0 \end{cases}, \forall t = 1, ..., T \end{cases}$$

and parameters

$$\begin{aligned}
d_t &= \text{demand in period } t \\
d_{tl} &= \sum_{t'=t}^{l} d_{t'}
\end{aligned}$$

with $L = \{1, ..., l\}$, the (l, S)-inequality of the lot sizing problem is

$$\sum_{t \in L \setminus S} x_t + \sum_{t \in S} d_{tl} y_t \geq d_{1l}$$

which is valid for any $l \in \{1, ..., T\}$ and $S \subseteq L$. Given x^*, y^* (fractional), the lot sizing problem asks if we can find l, S such that

$$\sum_{t \in L \setminus S} x_t^* + \sum_{t \in S} d_{tl} y_t^* < d_{1l}$$

or show none such exists. In other words, can we solve

$$z^{SEP} = \min_{l,S} \left\{ \sum_{t \in L \setminus S} x_t^* + \sum_{t \in S} d_{tl} y_t^* - d_{1l} \right\}$$

where if $z^{SEP} < 0$ then (l, S) is found, else none exists. To solve this, we try solving, for each $l = 1, ..., T$,

$$z_l^{SEP} = \min_{S \subseteq L} \left\{ \sum_{t \in L \setminus S} x_t^* + \sum_{t \in S} d_{tl} y_t^* \right\} - d_{1l}.$$

Note that each $t = 1, ..., L$ is either in S or not in S, so for each, if $x_t^* > d_{tl} y_t^*$ then put $t \in S$, else don't. In other words,

$$S^l = \{t \in \{1, ..., l\} : x_t^* > d_{tl} y_t^*\}$$

is the optimal solution for z_l^{SEP}. Clearly, $z^{SEP} = \min_{l=1,...,T} z_l^{SEP}$ and thus solving the separation problem takes $\mathcal{O}(T^2)$ operations which is polynomial in the size of the instance.

Example 5.4. [Node Packing Separation Problem]

Node packing in $G = (V, E)$ and clique inequalities $\sum_{i \in C} x_i \leq 1$ for $C \subseteq V$ a clique in G, have separation problem for a given x^*,

$$z^{SEP} = \max_{C} \left\{ \sum_{i \in C} x_i^* : C \text{ is a clique} \right\}$$

which is the max weighted clique problem. If $z^{SEP} > 1$, then optimal C gives a violated clique inequality; else there is none. This is an $\mathcal{NP}$-Hard problem, in general.

There are a number of greedy heuristics for solving it, e.g.

[insert diagram here]

Here, we can:

* order the nodes by the x_i^* values

* order the nodes by the degree weighted by x_i^*

Binary KP Cover Inequalities

Define

$$P = \left\{ x \in [0, 1]^n : \sum_{i=1}^{n} a_i x_i \leq b \right\}$$
$$S = P \cap \mathbb{Z}^n$$

and without loss of generality assume that $0 \leq a_i \leq b$ for all $i = 1, ..., n$.

Example 5.5. Given

$$S = \{x \in \{0, 1\}^6 : 13x_1 + 10x_2 + 9x_3 + 6x_4 + 4x_5 + 2x_6 \leq 15\},$$

consider, e.g. $C = \{2, 4, 5\}$, and observe that

$$\sum_{i \in C} a_i = 10 + 6 + 4 > 15$$

where not all variables in C can be 1. So $\sum_{i \in C} x_i \leq |C| - 1$ is valid for S. In general, if $\sum_{i \in C} a_i > b$, where C is called a **cover**, then

$$\sum_{i \in C} x_i \leq |C| - 1$$

is valid for S. The above is called the corresponding **cover inequality**. Note that $\{2, 4, 5\}$ is not a minimal cover since $a_2 + a_4 = 16 > 15$. So the cover inequality

$$x_2 + x_4 \leq 1 \qquad (\dagger)$$

dominates

$$x_2 + x_4 + x_5 \leq 2 \qquad (\ddagger)$$

since $x_5 \leq 1$ and $(\dagger)$ imply $(\ddagger)$. Also, $\hat{x} = (0, 1, 0, \frac{5}{6}, 0, 0) \in P$ violates $(\dagger)$ but does not violate $(\ddagger)$. Hence, adding $(\dagger)$ gives a better formulation than adding $(\ddagger)$.

In general, minimal cover inequalities dominate the others. For given $\hat{x} \in P$, the cover inequality separation problem is

$$\begin{aligned} z^{SEP} &= \max_{C \subseteq \{1,\ldots,n\}} \left(\sum_{i \in C} \hat{x}_i - |C| \right) \text{ s.t } \sum_{i \in C} a_i > b \\ &= \max_{C \subseteq \{1,\ldots,n\}} \left\{ \sum_{i \in C} (\hat{x}_i - 1) y_i : y \in \{0,1\}^n, \sum_{i \in C} a_i y_i \geq b + 1 \right\} \end{aligned}$$

assuming that b is integral. $z^{SEP} > -1$ then the optimal C gives a violated cover inequality; else there is none.

Flow Covers (N&W II.2.4)

Consider

$$P = \left\{ (x,y) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : \begin{array}{c} \sum_{j \in N^+} y_j - \sum_{j \in N^-} y_j \leq b \\ y_j \leq a_j x_j, \forall j \in N \\ x_j \leq 1, \forall j \in N \end{array} \right\}$$

where $N = \{1, \ldots, n\}$, $N^+ \cup N^- = N$, $N^+ \cap N^- = \emptyset$ and $S = P \cap (\mathbb{Z}^n \times \mathbb{R}^n)$. Note: $a_j \geq 0$ for all j is assumed and b may be positive or negative.

Example 5.6. Consider the following example

[insert diagram here]

A feasible point in P is

$$\begin{cases} x = \left(1, \frac{5}{6}, 0, 0, 1, 0\right) \\ y = (8, 5, 0, 0, 4, 0) \end{cases} \in P.$$

Similar to KP covers, we define $C \subseteq N^+$ to be a **flow cover** if $\sum_{j \in C} a_j > b$.

Proposition 5.2. *(N&W 4.3) If $C \subseteq N^+$ is a flow cover, $\lambda = \sum_{j \in C} a_j - b$ and $L \subseteq N^-$ then*

$$\sum_{j \in C} \left(y_j + (a_j - \lambda)^+ (1 - x_j)\right) \leq b + \sum_{j \in L} \lambda x_j + \sum_{j \in N^- \setminus L} y_j, \qquad (FCI)$$

*which we call the **flow cover inequality**, is valid for S.*

Example 5.7. In the last example, we can have

$$\begin{aligned} &C = \{1, 2\} \\ &\lambda = 8 + 6 - 9 = 6 \\ &L = \{6\} \end{aligned}$$

and the inequality

$$y_1 + y_2 + 3(1 - x_1) + (1 - x_2) \leq 9 + 5x_6 + y_5.$$

Our fractional point has $LHS = \frac{79}{6}$ and $RHS = 13$ which cuts off the point.

Theorem 5.1. *(N&W Thm 4.4) [insert statement here]*

{End Scribe}

Comparing Valid Inequalities

Definition 5.1. The inequalities $\pi x \leq \pi_0$ and $\alpha x \leq \alpha_0$ are **equivalent** if $\exists \lambda > 0$ such that $(\alpha, \alpha_0) = \lambda(\pi, \pi_0)$.

Definition 5.2. If $\exists \mu > 0$ such that $\alpha \geq \mu \pi$ and $\alpha_0 \leq \mu \pi_0$ (with at least one inequality strict), then (α, α_0) **dominates** or is stronger that (π, π_0).

Definition 5.3. A **maximal** valid inequality for S is not dominated by any other valid inequality. Maximal valid inequalities induce non-empty faces of conv(S).

5.1 Lifting

Lifting is a systematic process for strengthening valid inequalities and so obtaining higher dimensional faces of a polyhedron from lower dimensional faces.

Example 5.8. Consider conv(S) the node packing polytope on the graph with 6 nodes where nodes 1 to 5 form a pentagon and each node from 1 to 5 is connected to 6. We know that $x_1 + x_2 + ... + x_5 \leq 2$ is valid for S and it defines a facet of conv($S \cap \{\{0,1\}^6 : x_6 = 0\}$) (Proof: Exercise). Note it defines a face of conv(S) of dimension 4. Consider

$$x_1 + x_2 + ... + x_5 + \alpha x_6 \leq 2. \qquad (*)$$

What is the largest value of α such that $(*)$ is valid for S? Clearly $(*)$ is satisfied by any $x \in S$ with $x_6 = 0$. What if $x \in S$ with $x_6 = 1$? By the graph structure and node packing constraints, it must be that

$$x_1 = x_2 = ... = x_5 = 0.$$

So $\max\{x_1 + ... + x_5 + \alpha\} = \alpha \leq 2$. We may set $\alpha = 2$ to get the strongest inequality; we have **lifted** the original inequality to

$$x_1 + ... + x_5 + 2x_6 \leq 2$$

which induces a facet of conv(S) (it has dimension 5).

Example 5.9. Consider

$$S = \{x \in B^7 : 11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19\}$$

and note that $C = \{3, 4, 5, 6\}$ is a minimal cover since

$$6 + 5 + 5 + 4 = 20 > 19 \text{ (}\implies\text{ cover)}$$
$$6 + 5 + 5 = 16 \leq 19 \text{ (}\implies\text{ minimal)}.$$

Consider

$$\hat{S} = \left\{(x_3, x_4, x_5, x_6) \in B^4 : 6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 19\right\}.$$

The constraint

$$x_3 + x_4 + x_5 + x_6 \leq 3 \qquad (**)$$

defines a facet of conv($\hat{S}$) and defines a face of dimension 3 of conv(S) (Proof: Exercise). Consider

$$\beta x_1 + x_3 + x_4 + x_5 + x_6 \leq 3.$$

This is valid for $S \cap \{x : x_1 = 0\}$ for any β. For $x \in \hat{S}$ with $x_1 = 1$,

$$6x_3 + 5x_4 + 5x_5 + 4x_6 \leq 8 \implies x_3 + x_4 + x_5 + x_6 \leq 1$$

so

$$\max\{x_3 + ... + x_6 : x \in S, x_1 = 1\} = 1.$$

Thus, $\beta \leq 3 - 1 = 2$. Taking the largest lifting coefficient, $\beta = 2$, gives the strongest inequality of this form:

$$2x_1 + x_3 + ... + x_6 \leq 3$$

is valid for S and dominates $(**)$. What about

$$2x_1 + \gamma x_2 + x_3 + ... + x_6 \leq 3?$$

Exercise: This is valid for $\gamma \leq 1$ and thus

$$2x_1 + x_2 + x_3 + ... + x_6 \leq 3$$

is valid. Finally, what about $2x_1 + x_2 + x_3 + ... + x_6 + \delta x_7 \leq 3$?

Exercise: Valid for $\delta \leq 0$.

Exercise: Show $2x_1 + x_2 + ... + x_6 \leq 3$ is facet defining for conv(S).

Proposition 5.3. *(N&W II.2, #1.1) Suppose that $S \subseteq B^n$, $S^0 = \{x \in S : x_1 = 0\}$, $S^1 = \{x \in S : x_1 = 1\}$ and*

$$\sum_{j=2}^{n} \pi_j x_j \le \pi_0 \qquad (\dagger)$$

is valid for S. If $S^1 = \emptyset$ then $x_1 \le 0$ (i.e. $x_1 = 0$) is valid for S. Otherwise $S^1 \neq \emptyset$ and

$$\alpha x_1 + \sum_{i=2}^{n} \pi_j x_j \le \pi_0 \qquad (\ddagger)$$

is valid for S for any $\alpha \le \pi_0 - \zeta$ where

$$\zeta = \max\left\{\sum_{j=2}^{n} \pi_j x_j : x \in S\right\}.$$

Furthermore if $(\dagger)$ defines a face of conv(S^0) of dimension k and $\alpha = \pi_0 - \zeta$ then $(\ddagger)$ defines a face of conv(S) of dimension $k+1$. Importantly, if $(\dagger)$ is facet-defining for conv(S^0) then $(\ddagger)$ is facet-defining for conv(S).

Proof. Read N&W. □

Remark 5.2. We can also start with a valid inequality for S^1.

Example 5.10. (KP from above) Consider

$$S^1 = \{x \in S : x_1 = 1\} = \{x \in B^7 : x_1 = 1, 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \le 19 - 11 = 8\}$$

where $x_1 + x_3 \le 1$ is valid for S^1, i.e.

$$x_2 + x_3 \le 1 + \gamma(1 - x_1) \iff \gamma x_1 + x_2 + x_3 \le 1 + \gamma. \qquad (***)$$

This is clearly satisfied by any $x \in S$ with $x_1 = 1$. What about $x_1 = 0$? What is the smallest γ for which $(***)$ is valid for S? We have

$$\max\{x_2 + x_3 : x \in S, x_1 = 0\} = 2$$

and thus

$$1 + \gamma \ge 2 \implies \gamma \ge 2 - 1 = 1.$$

The strongest inequality of this form is found by taking $\gamma = 1$ and see

$$x_1 + x_2 + x_3 \le 1 + 1 = 2$$

is valid for S.

Proposition 5.4. *Suppose $(\dagger)$ is valid for S^1. If $S^0 = \emptyset$ then $x_1 \ge 1$ (i.e. $x_1 = 1$) is valid for S. Otherwise $S^0 \neq \emptyset$ and*

$$\gamma x_1 + \sum_{j=2}^{n} \pi_j x_j \le \pi_0 + \gamma \qquad (\phi)$$

is valid for S for any $\gamma \ge \zeta - \pi_0$ where

$$\zeta = \max\left\{\sum_{j=2}^{n} \pi_j x_j : x \in S^0\right\}.$$

Moreover, if $\gamma = \zeta - \pi_0$ and $(\dagger)$ gives a face of conv(S^1) of dimension k then (ϕ) gives a face of conv(S) of dimension $k+1$. Importantly, if $(\dagger)$ is facet-defining for conv(S^1) then (ϕ) is facet-defining for conv(S).

Remark 5.3. Given a sequential lifting procedure for an initial inequality, if a variable x is lifted first in the sequence it will get a coefficient not smaller than what it would get if lifted later in the sequence, and will get a coefficient larger if lifted last in the sequence that any it would get if lifted earlier.

(Its maximal lifting coefficient is a non-increasing function of its position in the sequence.)

5.2 General Purpose Cuts

From basic LP solutions

$$\begin{aligned}(IP)\ \max\ & cx \\ \text{s.t. } & Ax = b \\ & x \geq 0 \\ & x \in \mathbb{Z}^n.\end{aligned}$$

Consider a basic solution to the LP relaxation

$$x_B = B^{-1}b - B^{-1}Nx_N$$

where B is an $m \times m$ matrix. The basic solution is $\hat{x}$ where $\hat{x}_N = 0, \hat{x}_b = B^{-1}b$. Write $\bar{b} = B^{-1}b$, $\bar{A} = (\bar{a}_{ij}) = B^{-1}N$. Then

$$x_{\mathcal{B}_i} = \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{x}_{ij} x_{N_j}, \forall i = 1, ..., m$$

where $\mathcal{N}$ is the index set for the nonbasic variables and $\mathcal{B}$ is the index set for the basic variables. If $B^{-1}b \geq 0$ then

$$\hat{x} = (B^{-1}b, 0)$$

is feasible for the LP. Suppose that $\bar{b}_i$ is fractional with $\bar{b}_i \notin \mathbb{Z}$. Now $\hat{x}$ is the unique feasible solution with $\hat{x}_N = 0$. So $x_N = 0$ implies that x^{LP}, the solution to the LP relaxation, is not integer. Thus $x_N \neq 0$ is valid for the IP. Since $x \in \mathbb{Z}^n$ for the IP, it must be that

$$\sum_{j \in \mathcal{N}} x_{N_j} \geq 1$$

is valid for the IP.

5.3 Gomory Cuts

Developed in 1958, the procedure is as follows. Given $\bar{b}_i \notin \mathbb{Z}$, let

$$\begin{aligned}f_0 &= \bar{b}_i - \lfloor \bar{b}_i \rfloor \\ f_j &= \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor, \forall j \in \mathcal{N}.\end{aligned}$$

Write

$$x_{\mathcal{B}_i} = \lfloor \bar{b}_i \rfloor - \sum_{j \in \mathcal{N}} \lfloor \bar{a}_{ij} \rfloor x_{\mathcal{N}_j} + f_0 - \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j}$$

where $x_{\mathcal{B}_i}$is integer in any IP feasible solution and $\lfloor \bar{b}_i \rfloor - \sum_{j \in \mathcal{N}} \lfloor \bar{a}_{ij} \rfloor x_{\mathcal{N}_j} \in \mathbb{Z}$ if $x_{\mathcal{N}_j}$ are all integer (they are; $x_{\mathcal{N}_j} = 0$) which implies that $f_0 - \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j} \in \mathbb{Z}$. Thus for $(x_{\mathcal{B}}, x_{\mathcal{N}})$ to be an integer solution, it must be that $f_0 - \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j} \in \mathbb{Z}$. Thus, either

$$f_0 - \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j} \leq 0 \text{ or } f_0 - \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j} \geq 1.$$

However, the latter case implies

$$f_0 \geq 1 + \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j} \in \mathbb{Z} \geq 1$$

which contradicts the fact that $f_0 \in [0, 1)$. Thus,

$$f_0 \leq \sum_{j \in \mathcal{N}} f_j x_{\mathcal{N}_j}$$

is a valid inequality for the IP. It is a **Gomory cut**.

Example 5.11. Consider the problem

$$\begin{aligned} &\max 4x_1 - x_2 = z \\ &\text{s.t. } 7x_1 - 2x_2 + x_3 = 14 && (1) \\ &x_2 + x_4 = 3 && (2) \\ &2x_1 - 2x + x_5 = 3 && (3) \\ &x_1, x_2, x_3, x_4, x_5 \geq 0 \\ &x_1, x_2, x_3, x_4, x_5 \in \mathbb{Z} \end{aligned}$$

where x_3, x_4, x_5 are slack variables. The optimal LP equations are

$$\begin{aligned} z &= \frac{59}{7} - \frac{4}{7}x_3 - \frac{1}{7}x_4 \\ x_1 &= \frac{20}{7} - \frac{1}{7}x_3 - \frac{2}{7}x_4 && (1') \\ x_2 &= 3 - x_4 && (2') \\ x_5 &= \frac{23}{7} + \frac{2}{7}x_3 - \frac{10}{7}x_4. && (3') \end{aligned}$$

From $(1')$ we have $f_0 = \frac{6}{7}, f_1 = \frac{1}{7}, f_2 = \frac{2}{7}$. Thus the Gomory cut is

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7} \iff x_3 + 2x_4 \geq 6.$$

As an exercise, write the above in terms of x_1, x_2.

Alternatively, from $(3')$ we have $f_0 = \frac{2}{7}, f_1 = \frac{5}{7}, f_2 = \frac{3}{7}$ and

$$\frac{5}{7}x_3 + \frac{3}{7}x_4 \geq \frac{2}{7}$$

is valid. We can also add a cut from the objective equation!

Exercise: Try adding these cuts and solve the new LP. Repeat until the IP is solved.

Theorem 5.2. *(N&W II.4, #3.8) If a Gomory cut is added from the row of lowest index with fractional r.h.s., and then use lexicographical dual simplex to solve the new LP and then iterate. After at most*

$$(d+1)^{n+1}\left(d\sum_j (|c_j|+1)\right)$$

cuts the algorithm terminates with the optimal IP solution or proves the IP is infeasible. Here, the LP feasible set is contained in $[0,d]^n$.

Remark 5.4. The Gomory cut can be strengthened to

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

called an **extended Gomory cut**. As an exercise, prove it is valid. It is clearly stronger since

$$\begin{aligned} f_j > f_0 &\iff f_j - f_0 f_j > f_0 - f_0 f_j \\ &\iff f_j > \frac{f_0}{1-f_0}(1-f_j). \end{aligned}$$

Remark 5.5. The Gomory cut can also be extended to the mixed integer case. Consider the basic solution

$$x_i = \bar{b}_i - \sum_{j\in\mathcal{N}_I} \bar{a}_j x_j - \sum_{j \in \mathcal{N}_c} \bar{g}_j y_j$$

where $\mathcal{N}_I, \mathcal{N}_C$ are the indices of the respective integer and continuous nonbasic variables where $x_j \in \mathbb{Z}_+, y_j \in \mathbb{R}_+$. Similar to before, we have

$$x_i = \lfloor \bar{b}_i \rfloor - \sum_{j\in\mathcal{N}_I} \lfloor \bar{a}_j \rfloor x_j + f_0 - \sum_{j\in\mathcal{N}_I} f_j x_j - \sum_{s\in\mathcal{N}_c} \bar{g}_s y_s$$

where by similar arguments (N&W, #8.7 or Exercise) the cut is

$$\sum_{j\in\mathcal{N}_I} \min\left\{\frac{f_j}{f_0}, \frac{1-f_j}{1-f_0}\right\} x_j + \frac{1}{f_0} \sum_{\substack{j\in\mathcal{N}_c\\ \bar{g}_j<0}} \bar{g}_j y_j + \frac{1}{1-f_0} \sum_{\substack{j\in\mathcal{N}_I\\ \bar{g}_j>0}} \bar{g}_j y_j \geq 1$$

5.4 Chvátal-Gomory Rounding

For

$$\begin{aligned} &P = \{x \in \mathbb{R}^m_+ : Ax \leq b\}, A \in \mathbb{R}^{m\times n} \\ &S = P \cap \mathbb{Z}^m \end{aligned}$$

let $u \in \mathbb{R}^m_+$. Now for all $x \in S$,

$$\begin{aligned} Ax \leq b &\implies uAx \leq ub, u \geq 0 \\ &\implies \sum_{j=1}^n ua^j x_j \leq ub \\ &\implies \sum_{j=1}^n \lfloor ua^j \rfloor x_j \leq \lfloor ub \rfloor \end{aligned}$$

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

Example 5.12. (Node packing polytope on an odd cycle) Consider $u = \frac{1}{2}(1,1,...,1)$. We have the resulting (rounded) constraint

$$\sum_{i=1}^{2k+1} x_i \leq \left\lfloor \frac{2k+1}{2} \right\rfloor = k.$$

Example 5.13. (From earlier 2-variable IP) Consider

$$\begin{aligned} 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \end{aligned}$$

and take $u = \left(\frac{1}{7}, \frac{2}{7}, 0\right)$. Then $uAx \leq ub \iff x_1 \leq 2$ is valid.

Remark 5.6. The Chvátal-Gomory (C-G) procedure can be applied recursively. Define $P^0 = P$, $m_0 = m$, and $(A^0, b^0) = (A, b)$. Now define

$$P^{k+1} = \left\{ x \in P^k : \sum_{j=1}^n \lfloor ua^j \rfloor x_j \leq \lfloor ub \rfloor, \forall u \in \mathbb{R}^{m_k}_+ \right\}$$

and one can show that P^{k+1} can be represented by a finite number of constraints (N&W, II.1), say

$$P^{k+1} = \left\{ x \in \mathbb{R}^n_+ : A^{k+1}x \leq b^{k+1} \right\}$$

for some $(A^{k+1}, b^{k+1}) \in \mathbb{R}^{m_{k+1}\times(n+1)}$. We call P^k the **C-G closure** of S of rank k.

Theorem 5.3. *There exists finite k such that $P^k = conv(S)$.*

Proof. N&W II.1.2. □

Remark 5.7. If $\lambda x \leq \lambda_0$ is valid for S and there exists $\alpha \in \mathbb{R}_+$ such that $\alpha(\lambda, \lambda_0)$ is a row of (A^k, b^k) for some k then (λ, λ_0) has C-G rank at most k. If $\alpha(\lambda, \lambda_0)$ is not a row of (A^k, b^k) for any $\alpha \in \mathbb{R}_+$ then (λ, λ_0) has C-G rank at least $k+1$.

If (λ, λ_0) has C-G rank at least k and at most k then it has rank k.

Example 5.14. (Node packing) If $x_1 + x_2 + x_3 \leq 1$ is odd cycle constraint $\implies$ its C-G rank is at most 1. To see this, consider

$$\begin{cases} x_1 + x_2 + x_3 & \leq 1 \\ x_2 + x_3 + x_4 & \leq 1 \\ x_1 + x_3 + x_4 & \leq 1 \\ x_1 + x_2 + x_4 & \leq 1 \end{cases}$$

which are all valid C-G rank 1 inequalities. Taking $u = \frac{1}{3}(1, 1, ..., 1)^T$, gives us a clique inequality of size 4, i.e. this inequality is of C-G rank ≤ 2. Its rank is actually 2 (Proof: Exercise).

Fact 5.1. *A large clique of size of k is C-G rank $\mathcal{O}\left(\lfloor \frac{k+1}{2} \rfloor\right)$.*

6 Branch and Bound

See class notes.

7 Preprocessing

Example 7.1. (Wolsey, Ch. 7.4) Consider the problem

$$\begin{aligned} &\max\ 2x_1 + x_2 - x_3 \\ &\text{s.t. } 5x_1 - 2x_2 + 8x_3 \leq 15 && (1) \\ &\quad -8x_1 - 3x_2 + x_3 \leq -9 && (2) \\ &\quad x_1 + x_2 + x_3 \leq 6 && (3) \\ &\quad 0 \leq x_1 \leq 3 \\ &\quad 0 \leq x_2 \leq 1 \\ &\quad 1 \leq x_3 \end{aligned}$$

Analyze (1), seeking to tighten a bound for variable x_1.

$$\min\{-2x_2 + 8x_3 : 0 \leq x_2 \leq 1, 1 \leq x_3\} = -2 \times 1 + 8 \times 1 = 6$$

Then for x feasible, we must have

$$5x_1 + 6 \leq 15 \implies x_1 \leq \frac{9}{5}$$

and we can tighten $0 \leq x_1 \leq 3$ into $0 \leq x_1 \leq \frac{9}{5}$.

Imposing Bounds

Given constraint $\sum_{j\in B^+} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le b_i$, consider $k \in B^+$. So

$$
\begin{aligned}
& a^i_k x_k + \sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le b_i \\
\implies & a^i_k x_k + \underbrace{\min_{x\in S}\left\{\sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j\right\}}_{=:z} \le b_i \\
\implies & a^i_k x_k + z \le b_i \\
\implies & x_k \le \frac{b_i - z}{a^i_k}
\end{aligned}
$$

where S is a constraint set defined by some of the constraints of the original problem. If $x_k \in \mathbb{Z}$ then we can form the constraint $x_k \le \left\lfloor \frac{b_i - z}{a^i_k} \right\rfloor$.

Let us again analyze (1), using x_3:

$$\min\left\{5x_2 - 2x_2 : 0 \le x_1 \le \frac{9}{5}, 0 \le x_2 \le 1\right\} = 5\times 0 - 2\times 1 = -2$$

and thus $-2 + 8x_3 \le 15 \implies x_3 \le \frac{17}{8}$. Let's analyze (2), using x_1:

$$\min\left\{-3x_2 + x_3 : 0 \le x_2 \le 1, 1 \le x_3 \le \frac{17}{8}\right\} = -3\times 1 + 1 = -2$$

and thus $-8x_1 - 2 \le -9 \implies x_1 \ge \frac{7}{8}$ which reduces one of our constraints to $\frac{7}{8} \le x_1 \le \frac{9}{5}$. Repeat the previous analysis since now $x_1 \ge \frac{7}{8}$. We have

$$5\times\frac{7}{8} - 2\times 1 + 8x_3 \le 15 \implies x_3 \le \frac{101}{64}$$

so we may tighten the last constraint to $1 \le x_3 \le \frac{101}{64}$. Let us test (3) for redundancy:

$$
\begin{aligned}
& \max\left\{x_1 + x_2 + x_3 : \frac{7}{8} \le x_1 \le \frac{9}{5}, 0 \le x_2 \le 1, 1 \le x_3 \le \frac{101}{64}\right\} \\
=& \frac{9}{5} + 1 + \frac{101}{64} < 2 + 1 + 2 = 5 \le 6
\end{aligned}
$$

and so (3) is redundant.

Improving Coefficients

$$
\begin{aligned}
& \sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le z \qquad (x_k = 0) \\
\text{and } & a^i_k + \sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le b_i \qquad (x_k = 1) \\
\iff & a^i_k - (b_i - z) + \sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le b_i
\end{aligned}
$$

and thus $(a^i_k - (b_i - z))x_k + \sum_{j\in B^+\setminus\{k\}} a^i_j x_j - \sum_{j\in B^-} a^i_j x_j \le z$.

Consider the IP with constraint

$$\begin{cases} 2x_1 + 4x_2 \le 5 \\ x_1, x_2 \in \{0,1\} \end{cases}.$$

Note its LP relaxation has extreme points $\left(1, \frac{3}{4}\right)$ and $\left(\frac{1}{2}, 1\right)$. Let $m_2 = \max\{2x_1 : 0 \le x_1 \le 1\} = 2$. Now $m_2 = 2 < 5$ so

$$2x_1 + (4 - (5-2))x_2 \le 5 - (5-2) \iff 2x_1 + x_2 \le 2$$

is valid, which cuts off the first extreme point. Let $m_1 = \max\{x_2 : 0 \leq x_2 \leq 1\} = 1 < 2$ so

$$x_1 + x_2 \leq 1$$

is valid, which cuts off the second extreme point. The new LP

$$\begin{cases} x_1 + x_2 \leq 1 \\ x_1, x_2 \in \{0,1\} \end{cases}$$

has the integrality property and both fractional points have been cut off.

8 Reformulations

8.1 Dantzig-Wolfe Reformulation

Used for an IP in the form

$$z_{IP} = \begin{array}{l} \min cx \\ \text{s.t. } Ax \geq b \\ \quad x \in X = \{x \in \mathbb{Z}^n \times \mathbb{R}^p : \hat{A}x \geq \hat{b}\} \end{array}$$

where optimizing over X is "not too difficult". Recall that conv(X) can be represented in terms of its (finite sets of) extreme points and rays:

$$\text{conv}(X) = \left\{ x = \sum_{k \in K} \lambda_k x^k + \sum_{j \in J} \beta_j r^j, 1\lambda = 1, \lambda \geq 0, \beta \geq 0 \right\}$$

where $\{x^k\}_{x \in X}$ are the extreme points and $\{r^j\}_{j \in J}$ are the extreme rays of conv(X). Use this to substitute out for x in the formulation:

$$z_{IP} = \begin{array}{ll} \min\limits_{\lambda,\beta} & \sum\limits_{k \in K} (cx^k)\lambda_k + \sum\limits_{j \in J} (cr^j)\beta_j \\ \text{s.t.} & \sum\limits_{k \in K} (Ax^k)\lambda_k + \sum\limits_{j \in J} (Ar^j)\beta_j \geq b \\ & \sum\limits_{k \in K} \lambda_k = 1, \lambda \geq 0, \beta \geq 0 \\ & \sum\limits_{k \in K} x^k \lambda_k + \sum\limits_{j \in J} r^j \beta_j \in \mathbb{Z}^n \times \mathbb{R}^p \end{array}$$

which is what we call the **Dantzig-Wolfe** (DW) reformulation.

Column Generation Method

For solving the LP relaxation of the DW reformulation (for simplicity, assume X is bounded). The **master problem** is

$$z_{MP} = \begin{array}{ll} \min & \sum\limits_{k \in K} c^k \lambda_k \\ \text{s.t.} & \sum\limits_{k \in K} a^k \lambda_k \geq 0 \\ & \lambda \geq 0. \end{array}$$

The steps are:

Step 1: Choose an initial set of columns $\hat{K} \subseteq K$

Step 2: Solve the **restricted master problem** (RMP)

$$z_{RMP} = \begin{array}{ll} \min & \sum_{k \in \hat{K}} c^k \lambda_k \\ \text{s.t.} & \sum_{k \in \hat{K}} a^k \lambda_k \geq 0 \\ & \lambda \geq 0. \end{array}$$

to get an optimal LP dual solution u^*.

Step 3: Find a negative reduced cost variable or show that none exists:

* If $\min_{k \in K}(c^k - u^* a^k) \geq 0$ then STOP; the solution to RMP solves the MP
* Else, choose $k \in K$ with $c^k - u^* a^k < 0$ and set $\hat{K} := \hat{K} \cup \{k\}$ and go to Step 2.

Note that Step 3 can be modeled as an optimization problem:

$$\begin{array}{l} \min cx - u^* Ax \\ \text{s.t. } x \in \text{extr}(\text{conv}(X)) \end{array} = \begin{array}{l} \min cx - u^* Ax \\ \text{s.t. } x \in X. \end{array}$$

Example 8.1. (Binary Cutting Stock) The compact formulation is

$$\begin{aligned} z_{BCP} = \min & \sum_j y_j \\ \text{s.t.} & \sum_j x_{ij} = 1, \forall i \\ & \sum_i l_i x_{ij} \leq L y_j, \forall j \\ & x_{ij} \in \{0,1\}, \forall i, j \\ & y_j \in \{0,1\}, \forall j \end{aligned}$$

where y_j is 1 if stock piece j is used, x_{ij} is 1 if length i is cut from stock piece j, l_i is the length of order piece i, and L is the length of stock piece (assume they are all the same length).

Exercise: Deduce that the D-W reformulation of BCP formed by taking

$$X = \left\{ (x,y) \in \{0,1\}^{n \times n \times m} : \sum_i l_i x_{ij} \leq L y_j, \forall j \right\}$$

is equivalent to

$$\begin{aligned} z_{BCP} = \min & \sum_k \lambda_k \\ \text{s.t.} & \sum_k \lambda_k x_i^k = 1, \forall i = 1, ..., n \\ & \lambda \text{ binary} \end{aligned}$$

where

$$\tilde{X} = \{x^k : k = 1, 2, ..., K\} = \left\{ x \in \{0,1\}^n : \sum_i l_i x_i \leq L \right\}.$$

In apply the C.G. method, say u^* is a current dual multiplier for the RMP, step 3 seeks to solve the following pricing problem:

$$z_{PP} = \begin{array}{c} \min \left(1 - \sum_i u_i^* x_i\right) \\ \text{s.t. } \sum_i l_i x_i \leq L \\ x \in \{0,1\}^n \end{array} = 1 - \underbrace{\left[\begin{array}{c} \min \sum_i u_i^* x_i \\ \text{s.t. } \sum_i l_i x_i \leq L \\ x \in \{0,1\}^n \end{array}\right]}_{\text{Binary Knapsack Problem}}$$

If $z_{PP} < 0$ then we add its optimal solution as a new column (cutting pattern) to the restricted master problem; else STOP, we have solved the LP relaxation of the MP.

8.2 Lagrangian Duality

Again used for an IP in the form

$$z_{IP} = \begin{array}{l} \min cx \\ \text{s.t. } Ax \geq b \\ \quad x \in X = \{x \in \mathbb{Z}^n \times \mathbb{R}^p : \hat{A}x \geq \hat{b}\} \end{array}$$

where optimizing over X is "not too difficult". The **Lagrangian relaxation** of (1), $LR(u)$ is defined as (for a given u),

$$z_{LR}(u) = \begin{array}{l} \min cx + u(b - Ax) \\ \text{s.t. } x \in X. \end{array}$$

Proposition 8.1. *(Weak Duality) For any $u \geq 0$, $z_{LR}(u) \leq z_{IP}$.*

Proof. Let $u \geq 0$ and suppose x^* solves the IP. Then

$$\begin{aligned} z_{LP}(u) &\leq cx^* + u(b - Ax^*) \\ &\leq cx^* = z_{IP} \end{aligned}$$

since $x^* \in X$. □

The Lagrangian dual problem is the problem of finding the best LP lower bound:

$$z_{LD} = \begin{array}{l} \max z_{LR}(u) \\ \text{s.t. } u \geq 0. \end{array}$$

Fact 8.1. *$z_{LR}(u)$ is a concave piecewise affine function of u.*

Consider the problem

$$\begin{aligned} \min\ & 8x_1 + 3x_2 + 6x_3 \\ \text{s.t. } & 2x_1 + x_2 + 2x_3 \geq 5 \\ & 4x_1 + 2x_3 \geq 6 \\ & x \in \mathbb{Z}_+^3. \end{aligned}$$

Form a Lagrangian relaxation and have a Lagrangian dual problem by taking

$$X = \{x \in \mathbb{Z}_+^2 : 2x_1 + x_2 + 2x_3 \geq 5\},$$

i.e. dualize the second constraint. Thus,

$$\begin{aligned}
z_{LR}(u) &= \begin{pmatrix} \min 8x_1 + 3x_2 + 6x_3 + \\ \quad u(6 - 4x_1 - 2x_3) \\ \text{s.t. } x \in X \end{pmatrix} \\
&= 6u + \begin{pmatrix} \min\ (8-4u)x_1 + 3x_2 + (6-2u)x_3 \\ \text{s.t. } x \in X \end{pmatrix} \\
&= 6u + \begin{cases} -\infty, & \text{if } \exists \text{ extr. ray } r \text{ of conv}(X) \\ & \text{ s.t. } (8-4u, 3, 6-2u)r < 0 \\ \min_{k\in K} \left\{(8-4u, 3, 6-2u)x^k\right\}, & \text{otherwise} \end{cases}
\end{aligned}$$

where $\{x^k\}_{k\in K}$ is the set of extreme points of conv(X). The extreme points of conv(X) are (can be checked):

$$\{(2,1,0)^T, (0,1,2)^T, (0,5,0)^T, (3,0,0)^T, (0,0,3)^T\}$$

and the extreme rays of conv(X) are

$$\{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\}.$$

We can ensure $z_{LR}(u) \neq -\infty$ (is bounded) by ensuring that for all extreme rays of X, r, we have

$$\begin{aligned}
(8-4u, 3, 6-2u)r \geq 0 &\iff \begin{cases} (8-4u, 3, 6-2u)(1,0,0)^T \geq 0 \\ (8-4u, 3, 6-2u)(0,1,0)^T \geq 0 \\ (8-4u, 3, 6-2u)(0,0,1)^T \geq 0 \end{cases} \\
&\iff u \leq 2
\end{aligned}$$

When $z_{LR}(u)$ is bounded (has an optimum) it must be that an extreme point of conv(X) is optimum if

$$\begin{aligned}
z_{LR}(u) &= \begin{cases} -\infty, & u > 2 \\ 6u + \min_{k\in K}\left\{(8-4u, 3, 6-2u)x^k\right\}, & u \leq 2 \end{cases} \\
&= \begin{cases} -\infty, & u > 2 \\ 6u + \min\left\{19-2u, 15+2u, 15+6u, 24-6u, 18\right\}, & u \leq 2. \end{cases}
\end{aligned}$$

Drawing this out, the optimum value will occur at $u^* = 1$. The corresponding Lagrangian dual (LD) problem is

$$z_{LD} = \begin{matrix} \max z_{LR}(u) \\ \text{s.t. } 0 \leq u \leq 2 \end{matrix} = \begin{matrix} \max\ (\min\ \{19-2u, ..., 18\}) \\ \text{s.t. } 0 \leq u \leq 2 \end{matrix} = 17$$

which is achieved at $u^* = 1$ as expected. We can also model the LD problem as an LP:

$$\begin{aligned}
\max_{\eta, u}\ & \eta \\
\text{s.t. } & \eta \leq 19 - 2u \\
& \eta \leq 15 - 2u \\
& \vdots \\
& \eta \leq 18 \\
& 0 \leq u \leq 2
\end{aligned}$$

In general,

$$z_{LD} = \begin{pmatrix} \max\ \min\limits_{k\in K}\ \left\{cx^k + u(b - Ax^k)\right\} & \\ \quad \text{s.t. } u \geq 0 & \\ \qquad (c-uA)r^j \geq 0, & \forall j \in J \end{pmatrix}$$

$$= \begin{pmatrix} \max\limits_{\eta,u}\ \eta & \\ \quad \text{s.t. } \eta \leq cx^k + u(b - Ax^k), & \forall k \in K \\ \qquad (c-uA)r^j \geq 0, & \forall j \in J \\ \qquad u \geq 0 & \end{pmatrix}$$

$$= \begin{pmatrix} \max\limits_{\eta,u}\ \eta & \\ \quad \text{s.t. } \eta + (Ax^k - b)u \leq cx^k, & \forall k \in K \\ \qquad uAr^j \leq cr^j, & \forall j \in J \\ \qquad u \geq 0 & \end{pmatrix}.$$

This LP can be solved by a cutting plane algorithm.

Kelley's Cutting Plane Algorithm

Step 0: Find initial non-empty $\hat{K} \subseteq K$, $\hat{J} \subseteq J$

Step 1: Solve the master LD LP:

$$\begin{aligned} \max_{\eta,u}\ & \eta \\ \text{s.t. } & \eta + (Ax^k - b)u \leq cx^k, && \forall k \in \hat{K} \\ & uAr^j \leq cr^j, && \forall j \in \hat{J} \\ & u \geq 0 \end{aligned}$$

to get η^* and u^*.

Step 2: Solve the separation problem:

Step 2a: If $\min\{cr - u^*Ar : r$ is a ray of conv$(X)\} < 0$ then add its minimizer r^* as a new element in $\hat{J}$.

Step 2b: If $\min\{cx + u^*(b - Ax) : x$ is an extr. point of conv$(X)\} < \eta^*$ then add its minimizer x^* to $\hat{K}$.

If there is no change in $\hat{J}, \hat{K}$ then STOP; u^* is optimal.

Step 3: Else go to Step 1.

Dual LD

Consider the LP dual to the LD LP model:

$$\begin{pmatrix} \min\limits_{\lambda,\beta} & \sum\limits_{k\in K}(cx^k)\lambda_k + \sum\limits_{j\in J}(cr^j)\beta_j \\ \text{s.t.} & \sum\limits_{k\in K}\lambda_k = 1 \\ & \sum\limits_{k\in K}(Ax^k - b)\lambda_k + \sum\limits_{j\in J}(Ar^j)\beta_j \geq 0 \\ & \lambda \geq 0, \beta \geq 0 \end{pmatrix}$$

$$= \begin{pmatrix} \min\limits_{\lambda,\beta} c\left(\sum\limits_{k\in K} cx^k\lambda_k + \sum\limits_{j\in J} r^j\beta_j\right) \\ \text{s.t. } A\left[\sum\limits_{k\in K} x^k\lambda_k + \sum\limits_{j\in J} r^j\beta_j\right] \geq b \sum\limits_{k\in K}\lambda_k = b \\ \sum\limits_{k\in K}\lambda_k = 1 \\ \lambda \geq 0, \beta \geq 0 \end{pmatrix}.$$

Recall that

$$z_{IP} = \begin{array}{l} \min cx \\ \text{s.t. } Ax \geq b \\ \quad x \in X = \{x \in \mathbb{Z}^n \times \mathbb{R}^p : \hat{A}x \geq \hat{b}\} \end{array}$$

and so

$$z_{LD} = \min\{cx : Ax \geq b, x \in \text{conv}(X)\}.$$

This proves Thm. 6.2. of N&W II.3.

Corollary 8.1. *If X has the integrality property, then $z_{LD} = z_{LP}$, the value of the LP relaxation of the original compact formulation.*

In general, $z_{LD} \geq z_{LP}$.

8.3 Bender's Reformulation

Consider the LP

$$\begin{aligned} z = \max\ & cx + hy \\ \text{s.t. } & Ax + Gy \leq b \\ & x \in X \subseteq \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \end{aligned}$$

and consider x fixed. We get an LP:

$$LP(x) : z_{LP}(x) = \begin{pmatrix} \max hy \\ \text{s.t. } Gy \leq b - Ax \\ y \in \mathbb{R}^p_+ \end{pmatrix}$$

$$= \begin{pmatrix} \min u(b - Ax) \\ \text{s.t. } uG \geq h \\ u \geq 0, u \in \mathbb{R}^m \end{pmatrix}.$$

If the feasible region in the last formulation has extreme points $\{u^k\}_{k\in K}$ and extreme rays $\{v^j\}_{j\in J}$ then

$$
\begin{aligned}
z &= \begin{pmatrix} \max\ cx + z_{LP}(x) \\ \text{s.t. } x \in X \end{pmatrix} \\
&= \begin{pmatrix} \max\ cx + \begin{bmatrix} \min\limits_{k\in K} u^k(b - Ax) \\ \quad \text{s.t. } v^j(b - Ax) \ge 0, \quad \forall j \in J \end{bmatrix} \\ \text{s.t. } x \in X \end{pmatrix} \\
&= \begin{pmatrix} \max\ cx + \eta \\ \quad \text{s.t. } \eta \le u^k(b - Ax), \quad \forall k \in K \\ \qquad v^j(b - Ax) \ge 0, \qquad \forall j \in J \\ \qquad x \in X \end{pmatrix}.
\end{aligned}
$$

This can be solved with a cutting plane algorithm. Given x^*, solve

$$
\begin{aligned}
\min\ & u(b - Ax^*) \\
\text{s.t. } & uG \ge h \\
& u \ge 0.
\end{aligned}
$$

If this is unbounded, it must have a dual ray and we add it to $\hat{J}$. Else, we add an optimal solution u^* to $\hat{K}$.